



Universitat d'Alacant
Universidad de Alicante

Esta tesis doctoral contiene un índice que enlaza a cada uno de los capítulos de la misma.

Existen asimismo botones de retorno al índice al principio y final de cada uno de los capítulos.

[Ir directamente al índice](#)

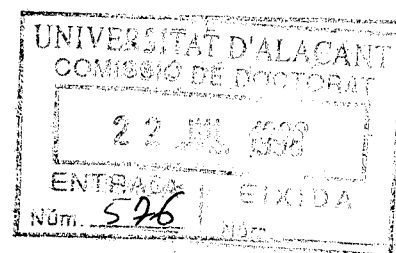
Para una correcta visualización del texto es necesaria la versión de [Adobe Acrobat Reader 7.0](#) o posteriores

Aquesta tesi doctoral conté un índex que enllaça a cadascun dels capítols. Existeixen així mateix botons de retorn a l'índex al principi i final de cadascun dels capítols .

[Anar directament a l'índex](#)

Per a una correcta visualització del text és necessària la versió d' [Adobe Acrobat Reader 7.0](#) o posteriors.

Aproximación computacional al tratamiento de la anáfora pronominal y de tipo adjetivo mediante gramáticas de unificación de huecos



Tesis Doctoral

*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
Alicante, julio de 1998*



*Tesis Doctoral presentada por Antonio Ferrández Rodríguez
Dirigida por Lidia Moreno Boronat y Manuel Palomar Sanz*



Universitat d'Alacant
Universidad de Alicante

La Dra. Lidia Moreno Boronat, titular de universidad del Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia y el Dr. Manuel Palomar Sanz, titular de universidad del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante,

CERTIFICAN:

que D. Antonio Ferrández Rodríguez, ha realizado bajo nuestra dirección la Tesis Doctoral titulada: "Aproximación computacional al tratamiento de la anáfora pronominal y de tipo adjetivo mediante gramáticas de unificación de huecos".

Hallándose concluida y reuniendo las condiciones de originalidad y rigor científico requeridos, autorizamos su presentación con el fin que pueda ser defendida y leída ante la Comisión Evaluadora correspondiente.

Para que así conste, expiden y firman el presente certificado en Valencia y Alicante respectivamente, a uno de julio de mil novecientos noventa y ocho.

Dra. Lidia Moreno Boronat

Dr. Manuel Palomar Sanz



Universitat d'Alacant
Universidad de Alicante

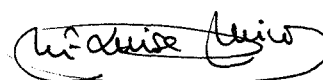
El Dr. Rafael Carrasco Jiménez, titular de universidad del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante, en calidad de Director de dicho Departamento,

RATIFICA:

la presentación de la Tesis Doctoral titulada: “Aproximación computacional al tratamiento de la anáfora pronominal y de tipo adjetivo mediante gramáticas de unificación de huecos” que D. Antonio Ferrández Rodríguez ha realizado bajo la dirección de los doctores Lidia Moreno Boronat y Manuel Palomar Sanz.

Para que así conste, expide y firma la presente en Alicante, a uno de julio de mil novecientos noventa y ocho.

Por delegación



Subdirectora del departamento

Dr. Rafael Carrasco Jiménez



Universitat d'Alacant
Universidad de Alicante

Agradecimientos

Mi más profundo y sincero agradecimiento a mis excepcionales directores, Lidia y Manolo, les debo tanto que es muy difícil expresar con palabras.

A mi esposa Marga, gracias a la cual se han hecho más llevaderas la infinidad de horas dedicadas a este trabajo, por su comprensión y apoyo. Y a mis padres, que me han transmitido su fuerza y tesón.

Y por último (y no menos importante), a mis compañeros Jesús, Patricio, Muñoz, Romero, Armando, Jaime y Borja, y a todo el grupo de investigación de programación lógica por su apoyo, sugerencias y buenos momentos pasados. A Natividad Prieto por sus valiosas sugerencias y revisiones. Al grupo de investigación de procesamiento del lenguaje natural del Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia. Y a todos los componentes del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante.

Índice

1. INTRODUCCIÓN	11
1.1 EL PROBLEMA DE LA RESOLUCIÓN DE LA ANÁFORA	11
1.2 RELACIÓN DE LA ANÁFORA CON OTROS TEMAS DE ESTUDIO	14
1.2.1 <i>Elipsis</i>	14
1.2.2 <i>Extracción de información</i>	19
1.2.3 <i>Traducción automática</i>	21
1.2.4 <i>Resolución del ligamiento de sintagmas preposicionales</i>	22
2. DEFINICIÓN DE CONCEPTOS	25
2.1 DEFINICIÓN DE ANÁFORA	25
2.2 TIPOS DE ANÁFORA	30
2.2.1 <i>En función del marco en que sucede</i>	30
2.2.1.1 Anáfora intraoracional	31
2.2.1.2 Anáfora discursiva	31
2.2.2 <i>En función de la accesibilidad del antecedente</i>	31
2.2.2.1 Anáfora morfosintáctica: nominal, verbal y oracional	32
2.2.2.2 Anáfora semántica: sinonimia, hiperonimia, contextual y forzada	33
2.2.2.3 Anáfora pragmática	34
2.2.3 <i>En función del tipo de referencia</i>	34
2.2.3.1 Anáfora profunda	34
2.2.3.2 Anáfora superficial	35
2.2.4 <i>Según el tipo de expresión anafórica</i>	36
2.2.4.1 Anáfora pronominal	37
2.2.4.2 Descripciones definidas	37
2.2.4.3 Anáfora de tipo "one", anáfora tipo adjetivo y anáfora superficial	39
numérica	39
2.2.4.4 Anáfora verbal	40

2.2.4.5 Adverbios y complementos circunstanciales	41
2.3 OBJETIVOS DE ESTE TRABAJO	41
<u>3. ESTRATEGIAS PARA LA RESOLUCIÓN DE LA ANÁFORA</u>	43
3.1 FUENTES DE INFORMACIÓN	44
3.1.1 <i>Información morfológica</i>	44
3.1.2 <i>Información léxica</i>	45
3.1.3 <i>Información sintáctica</i>	46
3.1.4 <i>Información semántica</i>	53
3.1.5 <i>Información pragmática</i>	55
3.1.5.1 <i>Información del mundo exterior e inferencia</i>	55
3.1.5.2 <i>Información referente a la construcción del discurso</i>	56
3.1.5.2.1 <i>Distancia</i>	56
3.1.5.2.2 <i>Competencia</i>	57
3.1.5.2.3 <i>Prominencia</i>	58
3.1.5.2.4 <i>Unidad</i>	58
3.1.6 <i>Información sobre la expresión anafórica</i>	58
3.1.7 <i>Información obtenida a partir del estudio del corpus</i>	59
3.2 ALGORITMOS	61
3.2.1 <i>Primeras aproximaciones al tratamiento de la anáfora</i>	62
3.2.2 <i>Sistemas integrados basados en el conocimiento</i>	67
3.2.2.1 <i>Sistemas democráticos basados en restricciones y preferencias</i>	68
3.2.2.2 <i>Sistemas democráticos basados únicamente en preferencias</i>	70
3.2.2.3 <i>Sistemas pobres en conocimiento</i>	76
3.2.2.4 <i>Sistemas consultivos: teoría del foco del discurso</i>	83
3.2.3 <i>Sistemas alternativos</i>	91
<u>4. FORMALISMO GRAMATICAL SUG</u>	95
4.1 FUNDAMENTOS TEÓRICOS	95
4.1.1 <i>Concepto de "gramática"</i>	95
4.1.2 <i>Análisis sintáctico</i>	98
4.1.3 <i>Análisis semántico</i>	99
4.2 GRAMÁTICAS BASADAS EN LA UNIFICACIÓN	103
4.2.1 <i>Fundamentos teóricos</i>	103
4.2.2 <i>Clasificación de las gramáticas de unificación</i>	106
4.2.2.1 <i>Gramáticas de cláusulas definidas</i>	107
4.2.2.2 <i>Gramáticas de huecos</i>	110
4.3 GRAMÁTICA DE UNIFICACIÓN DE HUECOS (SUG)	111
4.3.1 <i>SUG como una extensión de las DCG</i>	111
4.3.2 <i>Definición formal</i>	113
4.3.3 <i>Reglas de producción</i>	114

<i>Índice</i>	<i>iii</i>
4.3.4 <i>Hechos SUG</i>	117
4.3.4.1 Hecho “coordinated”	117
4.3.4.2 Hecho “juxtaposition”	120
4.3.4.3 Hecho “fusion”	121
4.3.4.4 Hecho “basicWord”	123
4.3.4.5 Hecho “esPalabra”	124
4.3.5 <i>Ejemplo de una gramática en notación SUG</i>	125
4.4 SISTEMA SUP	127
4.4.1 <i>Esquema de trabajo del sistema SUP</i>	128
4.4.2 <i>Traductor de SUG a Prolog</i>	129
4.4.3 <i>Análisis sintáctico</i>	130
4.4.3.1 Estructura de huecos devuelta por el analizador	130
4.4.3.2 Reduciendo el número de accesos al diccionario	133
4.4.4 <i>Análisis semántico</i>	136
4.4.4.1 Proceso de análisis semántico en SUP	136
4.4.4.2 Especificación del lenguaje de la fórmula lógica	139
4.4.4.2.1 Oraciones declarativas	140
4.4.4.2.2 Conectivas lógicas	140
4.4.4.2.3 Fórmula lógica de los cuantificadores	141
4.4.4.2.4 Fórmula lógica de la coordinación	142
4.4.4.2.5 Fórmula lógica de los verbos y sintagmas verbales	143
4.4.4.2.6 Fórmula lógica de los sintagmas nominales	144
4.4.4.2.7 Fórmula lógica de los sintagmas preposicionales	145
5. PROPUESTA DE RESOLUCIÓN DE LA ANÁFORA	147
5.1 ALGORITMO DE RESOLUCIÓN DE LA ANÁFORA.....	147
5.1.1 <i>Determinación de la correferencia</i>	152
5.1.2 <i>Información utilizada en el algoritmo</i>	154
5.1.2.1 Información léxica y morfológica	154
5.1.2.2 Información sintáctica	155
5.1.2.3 Información semántica	161
5.1.3 <i>Sistema de restricciones y preferencias</i>	162
5.1.3.1 Restricciones	162
5.1.3.2 Preferencias	164
5.2 APLICACIÓN DEL ALGORITMO SOBRE TEXTOS NO RESTRINGIDOS.....	165
5.2.1 <i>Entrada del sistema</i>	166
5.2.2 <i>Análisis sintáctico parcial</i>	168
5.2.3 <i>Transformación del algoritmo de tratamiento de la anáfora</i>	172
6. EXPERIMENTOS Y RESULTADOS	177
6.1 EVALUACIÓN DEL COMPORTAMIENTO DEL SISTEMA SUP.....	177
6.1.1 <i>Gramática</i>	177

6.1.2 Corpus de prueba de frases declarativas	180
6.1.3 Resultados obtenidos	181
6.2 EXPERIMENTO 1: TRATAMIENTO DE LA ANÁFORA EN TEXTOS RESTRINGIDOS	185
6.3 EXPERIMENTO 2: TRATAMIENTO DE LA ANÁFORA EN TEXTOS NO RESTRINGIDOS	186
6.4 ANÁLISIS COMPARATIVO DE RESULTADOS	187
7. CONCLUSIONES Y TRABAJOS FUTUROS	191
7.1 CONTRIBUCIONES	191
7.2 TRABAJOS FUTUROS	194
7.3 RESUMEN FINAL	194
8. BIBLIOGRAFÍA	197
APÉNDICE A: GRAMÁTICA SUG RESTRINGIDA	211
APÉNDICE B: CORPUS DE FRASES DECLARATIVAS	217
APÉNDICE C: CORPUS DE FRASES CON ANÁFORA	241
APÉNDICE D: RESULTADOS OBTENIDOS DEL TRATAMIENTO DE LA ANÁFORA PRONOMINAL EN EL CORPUS DEL APÉNDICE C	243
APÉNDICE E: RESULTADOS OBTENIDOS DEL TRATAMIENTO DE LA ANÁFORA DE TIPO ADJETIVO EN EL CORPUS DEL APÉNDICE C	255
APÉNDICE F: RESULTADOS OBTENIDOS DEL TRATAMIENTO DE LA ANÁFORA EN TEXTOS NO RESTRINGIDOS	267

APÉNDICE G: TRADUCTOR DE SUG A PROLOG

279

APÉNDICE H: ALGORITMO DE RESOLUCIÓN DE LA ANÁFORA

303

Universitat d'Alacant
Universidad de Alicante

Índice de figuras

FIGURA 1. REGLAS GRAMATICALES SUG PARA LA RESOLUCIÓN DE LA OMISIÓN DEL NÚCLEO DEL SINTAGMA NOMINAL.....	19
FIGURA 2. CLASIFICACIÓN DE TIPOS DE REFERENCIAS.	27
FIGURA 3. TIPOS DE ANÁFORA EN FUNCIÓN DE LA ACCESIBILIDAD DEL ANTECEDENTE.	32
FIGURA 4. EJEMPLO DE APLICACIÓN DE LA RESTRICCIÓN C-DOMINIO NÚMERO 2 EN: “ANTE ÉL, PEDRO VIO UNA SERPIENTE”.....	48
FIGURA 5. EJEMPLO DE APLICACIÓN DE LA RESTRICCIÓN C-DOMINIO NÚMERO 2 EN: “ANTE PEDRO, ÉL VIO UNA SERPIENTE”.....	48
FIGURA 6. EJEMPLO DE APLICACIÓN DE LA RESTRICCIÓN C-DOMINIO NÚMERO 1 EN: “ANTE PEDRO, PEDRO VIO UNA SERPIENTE”.....	48
FIGURA 7. APLICACIÓN DE LA RESTRICCIÓN C-DOMINIO NÚMERO 3 EN: “LUISA SE ABURRE”.	49
FIGURA 8. APLICACIÓN DE LA RESTRICCIÓN C-DOMINIO NÚMERO 4 EN: “LUISA LE ABURRE”.	49
FIGURA 9. EJEMPLO DE CORREFERENCIA EN: “JUAN ENTRÓ Y ÉL COMPRÓ”.	49
FIGURA 10. VARIACIÓN DE LA DISTANCIA EN EL TIPO DE EXPRESIÓN ANAFÓRICA.	57
FIGURA 11. RELACIÓN ENTRE COMPETENCIA Y EXPRESIÓN ANAFÓRICA.	57
FIGURA 12. RELACIÓN ENTRE UNIDAD Y EXPRESIÓN ANAFÓRICA.	58
FIGURA 13. INFORMACIÓN NECESARIA PARA RESOLVER CADA TIPO DE EXPRESIÓN ANAFÓRICA.....	59
FIGURA 14. CÁLCULO DEL ÁNGULO DE SEPARACIÓN ENTRE DOS VECTORES BASADO EN EL PRODUCTO ESCALAR ENTRE AMBOS.	71
FIGURA 15. CONSTANTE A APLICAR PARA LA NORMALIZACIÓN DE VECTORES.....	73
FIGURA 16. ESQUEMA DE CÁLCULO DEL FACTOR DE CERTEZA.	75
FIGURA 17. VALORES NUMÉRICOS ASIGNADOS A LAS PREFERENCIAS.	78
FIGURA 18. CAMBIOS DE FOCO DEL DISCURSO.	86
FIGURA 19. PLANTILLAS OBTENIDAS DEL ANÁLISIS DEL CORPUS SEGÚN DAGAN E ITAL.....	92
FIGURA 20. FRECUENCIAS DE EXPRESIONES ANAFÓRICAS ENCONTRADAS EN EL CORPUS.	93
FIGURA 21. REGLA PARA LA RESOLUCIÓN DEL PRONOMBRE “IT” CUANDO FUNCIONA COMO OBJETO... 94	94
FIGURA 22. EJEMPLO DE UNA GRAMÁTICA INCONTEXTUAL.	96
FIGURA 23. EJEMPLOS DE PALABRAS EN EL LENGUAJE DE CPPO.....	100
FIGURA 24. EJEMPLOS DE CUANTIFICADORES.	100
FIGURA 25. LENGUAJE DE FÓRMULA LÓGICA PROPUESTO POR COVINGTON.....	101
FIGURA 26. REPRESENTACIÓN DE SINTAGMAS NOMINALES PLURALES SEGÚN ALLEN.	102
FIGURA 27. EJEMPLO DE UNA “ESTRUCTURA DE RASGOS”.	103
FIGURA 28. EJEMPLOS DE RELACIONES JERÁRQUICAS ENTRE ATRIBUTOS.....	104
FIGURA 29. EJEMPLO DE “SUBSUNCIÓN”, EN LA QUE D_{SN} SUBSUME A D_{SN3SG}	104
FIGURA 30. EJEMPLO DE DOS ESTRUCTURAS CON INFORMACIÓN DIFERENTE PERO COMPATIBLE.....	105
FIGURA 31. EJEMPLO DE DOS ESTRUCTURAS CON INFORMACIÓN DIFERENTE Y CONFLICTIVA.....	105
FIGURA 32. EJEMPLO DE UNA REGLA GRAMATICAL DE PATR-II.	105

FIGURA 33. ESTRUCTURA FUG EQUIVALENTE A LA REGLA MOSTRADA EN LA FIGURA 32.	106
FIGURA 34. EJEMPLOS DE GRAMÁTICAS DCG.	108
FIGURA 35. EQUIVALENCIA ENTRE REGLAS DE FORMALISMOS PATR-II, FUG Y DCG.	109
FIGURA 36. GRAMÁTICA PROPUESTA POR CARVALHO PARA DEFINIR RELACIONES DE CORREFERENCIA.	113
FIGURA 37. EJEMPLOS DE SINTAGMAS NOMINALES.	114
FIGURA 38. COMPARACIÓN ENTRE DCG Y SUG: CONSTITUYENTES OPCIONALES.	115
FIGURA 39. REGLAS GRAMATICALES SUG PARA LAS ORACIONES PERSONALES E IMPERSONALES.	116
FIGURA 40. IMPLEMENTACIÓN EN PROLOG DE LOS PREDICADOS "IFTHEN" E "IFTHENELSE".	116
FIGURA 41. CONSTITUYENTES OPCIONALES CON RESTRICCIONES.	117
FIGURA 42. EJEMPLO DE COORDINACIÓN CON RESTRICCIÓN: ADJETIVOS CALIFICATIVOS.	118
FIGURA 43. COORDINACIÓN DE ORACIONES.	119
FIGURA 44. COORDINACIÓN DE SINTAGMAS NOMINALES DE DISTINTO TIPO.	119
FIGURA 45. ANÁLISIS DE LA COORDINACIÓN MEDIANTE NOTACIÓN DCG.	120
FIGURA 46. YUXTAPOSICIÓN Y COORDINACIÓN DE ADJETIVOS CALIFICATIVOS.	121
FIGURA 47. FUSIÓN DE SINTAGMAS NOMINALES DE TIPO VERBAL.	122
FIGURA 48. FUSIÓN EN LAS REGLAS GRAMATICALES DE LA APOSICIÓN.	123
FIGURA 49. EJEMPLOS DE HECHOS SUG "BASICWORD".	124
FIGURA 50. DICCIONARIO SUG IMPLEMENTADO MEDIANTE HECHOS "ESPALABRA".	125
FIGURA 51. EJEMPLO DE GRAMÁTICA SUG CON "ORACION" COMO SÍMBOLO INICIAL.	126
FIGURA 52. EJEMPLO DE FRASE ACEPTADA POR LA GRAMÁTICA DE LA FIGURA 51.	127
FIGURA 53. ESQUEMA DE TRABAJO DEL SISTEMA SUP.	128
FIGURA 54. ESTRUCTURA DE HUECOS DEL SINTAGMA NOMINAL "UN PERRO".	132
FIGURA 55. EJEMPLO DE LA ESTRUCTURA DE HUECOS OBTENIDA EN LA COORDINACIÓN.	133
FIGURA 56. COMPARACIÓN ENTRE LA LISTA DE PALABRAS EMPLEADA EN SUP Y LA UTILIZADA EN DCG.	134
FIGURA 57. PROCESO DE PODA CON LA ORACIÓN IMPERSONAL: "LLUEVE MUCHO".	135
FIGURA 58. PODA REALIZADA EN LA ORACIÓN: "ELLOS LO ROMPIERON".	135
FIGURA 59. ESPECIFICACIÓN DE LA FÓRMULA LÓGICA EN UNA DCG.	137
FIGURA 60. ALGORITMO PARA EL CÁLCULO DE LA FÓRMULA LÓGICA.	139
FIGURA 61. FASE DE ACTIVACIÓN DEL MÓDULO DE TRATAMIENTO DE LA ANÁFORA DENTRO DEL SISTEMA SUP.	148
FIGURA 62. SALIDA PARA LA FRASE: "PEDRO Y ANA ENTRARON Y ÉL COMPRÓ".	149
FIGURA 63. ALGORITMO PARA LA RESOLUCIÓN DE LA ANÁFORA.	150
FIGURA 64. EJEMPLO DE LA LISTA DE ANTECEDENTES.	152
FIGURA 65. EJEMPLO DE CORREFERENCIA EN: "LUISA Y ANA COMPRARON UNA PERA VERDE Y UNA MANZANA ROJA AYER Y PEDRO ELIGIÓ LA VERDE".	153
FIGURA 66. EJEMPLO DE <u>NO</u> CORREFERENCIA EN: "LUISA COMPRÓ UNA MANZANA VERDE Y OTRA ROJA".	154
FIGURA 67. EJEMPLO DE COMPROBACIÓN DE RESTRICCIONES C-DOMINIO POR MEDIO DE LA ESTRUCTURA DE HUECOS DE SUP EN: "JUAN ENTRÓ Y ÉL COMPRÓ".	157
FIGURA 68. RESTRICCIONES C-DOMINIO EN "MARGA COMPRÓ PARA ELLA".	157
FIGURA 69. RESTRICCIONES C-DOMINIO EN "LA HERMANA DE ELLA".	158
FIGURA 70. RESTRICCIONES C-DOMINIO EN "PARA MARGA Y PARA ELLA".	158
FIGURA 71. RESTRICCIONES C-DOMINIO EN "COMPRÉ UN LIBRO PARA MARGA Y PARA LA HERMANA DE ELLA".	159
FIGURA 72. RESTRICCIONES C-DOMINIO EN "MARÍA DIJO QUE ELLA VIÓ A PEDRO".	159
FIGURA 73. RESTRICCIONES C-DOMINIO EN "ÉSTE ES EL HOMBRE QUE ÉL VIO".	160
FIGURA 74. APLICACIÓN DEL PARALELISMO SINTÁCTICO EN: "ANA MIRÓ A LA PERRA Y LA GATA, Y ELLA PREFIRIÓ LA SEGUNDA".	160
FIGURA 75. APLICACIÓN DE IRSAS PARA LA COMPROBACIÓN DE CONSISTENCIA SEMÁNTICA EN EL TEXTO: "EL RATÓN ₁ SE PARÓ CERCA DEL COCHE. ÉSTE ₂ SE COMIÓ UN TROZO DE QUESO".	161
FIGURA 76. CRITERIOS DE PREFERENCIA.	164
FIGURA 77. CÓDIGO PROLOG QUE REALIZA LA TRANSFORMACIÓN DE ETIQUETAS XEROX A SUG.	166
FIGURA 78. ESQUEMA DE TRABAJO DEL TRADUCTOR DE ETIQUETAS XEROX A SUG.	167

FIGURA 79. EJEMPLO DE TRADUCCIÓN DE ETIQUETAS XEROX A SUG.	167
FIGURA 80. GRAMÁTICA SUG RELAJADA PARA TEXTOS NO RESTRINGIDOS.	170
FIGURA 81. RESULTADO DEL ANÁLISIS PARCIAL DE UNA ORACIÓN DEL CORPUS.	172
FIGURA 82. EJEMPLO DE APLICACIÓN DEL ALGORITMO SOBRE TEXTOS NO RESTRINGIDOS.	176
FIGURA 83. FUSIÓN DE SINTAGMAS NOMINALES TIPO SUSTANTIVO Y TIPO ADJETIVO.	179
FIGURA 84. ALGORITMO DE PODA.	180
FIGURA 85. INFORME DE TIEMPOS EMPLEADOS PARA EL ANÁLISIS DEL CORPUS DE FRASES DECLARATIVAS.	181
FIGURA 86. COLECCIÓN DE ORACIONES EMPLEADA PARA LA EVALUACIÓN DE SUP.	182
FIGURA 87. EFICIENCIA DE LAS ORACIONES DE LA FIGURA 86.	183
FIGURA 88. ANÁLISIS DE DOS FRASES DEL CORPUS DE FRASES CON ANÁFORA.	185

1. Introducción

1.1 El problema de la resolución de la anáfora

La resolución de la anáfora se podría decir que es uno de los problemas más difíciles pendientes de solución en el campo de estudio del procesamiento del lenguaje natural. El *procesamiento del lenguaje natural* es una de las ramas principales de la *Inteligencia Artificial*, y estudia una propiedad importante de la inteligencia humana: su capacidad de comunicarse por medio del lenguaje. Si queremos construir ordenadores que simulen inteligencia, éstos deben ser capaces de comunicarse con otros agentes en lenguaje natural, ya sean ordenadores o humanos. Aunque hay muchos desarrollos en este campo, estamos todavía lejos de construir un sistema de propósito general, y la mayoría de ellos sólo pueden trabajar en entornos restringidos, como por ejemplo la extracción de cierta información de una base de datos.

Podemos encontrar una definición general de anáfora en el trabajo de Rico [143], que la define como la relación de referencia que se establece entre una forma lingüística (*expresión anafórica*) y un objeto, una persona o una situación que ya han sido mencionados de manera implícita o explícita con anterioridad durante el proceso comunicativo y que llamaremos *antecedente*. Siguiendo este mismo trabajo, la anáfora se puede considerar como un fenómeno básicamente discursivo, ya que es el propio discurso el que proporciona y distribuye en diferentes enunciados la información necesaria para interpretar las relaciones anafóricas, y además el mantener la coherencia discursiva es uno de los principales objetivos de la anáfora. Por tanto, el paradigma de estudio en este caso será la *pragmática*, ya que es dentro de esta perspectiva donde se considera el lenguaje como discurso:

La pragmática define el discurso como una entidad global dotada de cohesión y coherencia. Ésta estudia los diferentes aspectos de la estructura del discurso y el contexto que intervienen en la interpretación del enunciado.

Según Dahlbäck [47], la comprensión del discurso se ha convertido en uno de los campos de estudio más importantes dentro del procesamiento del lenguaje natural.

Dahlbäck considera que para comprender un discurso se necesitan al menos tres tipos de información:

- ◆ El primero es el correspondiente a la *información lingüística*. Éste incluye el significado de las palabras y el modo de combinar gramaticalmente estas palabras para obtener oraciones con sentido.
- ◆ El segundo viene a ser las *acciones desarrolladas por el hablante y los objetivos e intenciones del mismo*. Éste incluye la composición de oraciones, órdenes, preguntas, etc. Por ejemplo, una pregunta puede tener un significado real de una petición, como en la oración: *¿Puede abrir la puerta?*. Por consiguiente, necesitamos algún tipo de conocimiento para comprender la relación entre lo que el hablante dice y lo que realmente quiere decir. El éxito de sistemas que tienen la capacidad de utilizar el lenguaje natural como humanos depende del reconocimiento de las intenciones de los participantes del discurso.
- ◆ El tercer tipo se encuentra *fuera del discurso*. Desde el principio del estudio del lenguaje natural, se sabe que la información contenida únicamente en el discurso no es suficiente para comprenderlo, es necesario tener un conocimiento general previo del objeto de ese discurso. La carencia de este entendimiento hace que el discurso sea incomprensible o abierto a multitud de interpretaciones. Los participantes en un discurso normalmente asumen que los otros participantes también tienen esa información general. Esta suposición ofrece la posibilidad de hacer rápidas referencias a objetos que no han sido explícitamente mencionados en el discurso previo, pero que existen en los antecedentes del propio objeto del discurso.

La resolución de la anáfora es un importante proceso dentro del lenguaje natural. Las expresiones anafóricas se utilizan por los humanos de una manera tan intuitiva y sencilla que en muchas ocasiones se pasa por alto su importancia. Podemos encontrar varios ejemplos en las siguientes frases que exigen la resolución correcta de la anáfora¹: *If the baby, does not thrive on raw milk_j, boil it_j. Si estos zapatos_i, no te caben en tus pies_j, puedes cambiarlos_j. Si cae [una bomba incendiaria]_i, cerca de ti, no pierdas la cabeza_j. Ponla_i en una cubeta y cúbrela_i con arena.*

Las estrategias que se están aplicando actualmente se basan en las aplicadas en la teoría lingüística tradicional. La desventaja que existe a la hora de implementar estas teorías es la necesidad de representar y manejar diferentes tipos de conocimiento lingüístico y del mundo exterior.

Actualmente se acepta (como ejemplo podemos citar los trabajos de Rico [143], Stuckardt [157] y Carbonell y Brown [24]) que el proceso de resolución de la anáfora en el lenguaje natural se basa en una serie de estrategias que emplean diferentes clases de conocimiento. Es decir, que el proceso de determinar el conjunto de posibles antecedentes de una determinada anáfora esté gobernado por información *morfológica, sintáctica, semántica y pragmática*. Esto quiere decir que teóricamente no podemos basarnos en una

¹ En este trabajo denotaremos con subíndices alfanuméricos la relación existente entre una expresión anafórica y su antecedente. También delimitaremos entre corchetes los antecedentes o expresiones anafóricas formados por varias palabras, como ocurre en: *Si cae [una bomba incendiaria]_i, cerca de ti, no pierdas la cabeza_j.*

única fuente de conocimiento, aunque en la práctica se han implementado determinados algoritmos que funcionan relativamente bien con poca información, como por ejemplo el de Lappin y Leass [95] donde se utiliza tan sólo información sintáctica para la resolución de la anáfora pronominal, obteniéndose un porcentaje de éxito del 85%.

A continuación pasamos a revisar brevemente el conocimiento limitado en estas fuentes de información necesarias para la resolución de la anáfora:

- ◆ La información *morfológica* hace referencia a la necesaria concordancia en *número, género y persona* entre la expresión anafórica y su antecedente. Por ejemplo, en la frase *Pedro_i y Ana_j son buenos amigos. Sin embargo ella_i comenzó a evitarlo_i*.
- ◆ La información *sintáctica* corresponde a la relación que ha de existir entre las funciones gramaticales de la expresión anafórica y su antecedente. Esta información viene englobada en las llamadas restricciones *c-dominio*, en la propia estructura sintáctica del antecedente y el *paralelismo sintáctico*, y como ejemplo de su utilidad podemos mostrar el siguiente ejemplo: *Ana no dió a ningún niño [la camisa verde de manga corta]_i, pero a Marga le dio la roja_j*, en el cual se utiliza la información correspondiente a la estructura sintáctica del sintagma nominal antecedente de la expresión anafórica *la roja*, para determinar que se refiere a una nueva entidad: *la camisa roja de manga corta*.
- ◆ La información *semántica* obliga a que haya una cierta compatibilidad entre los rasgos semánticos de la expresión anafórica y su antecedente. Esta información es fundamental en ciertas ocasiones, como por ejemplo en la siguiente frase: *El ratón_i se paró cerca del coche_j. Éste_i encontró un trozo de queso_k y se lo_k comió*.
- ◆ Y finalmente la información *pragmática* permite introducir cierta información del mundo exterior e inferencia de conocimiento, información importante por ejemplo en la siguiente frase: *El FBI_i protege al país_j de todo aquel que amenaza su libertad*, o en: *Los soldados_i dispararon a los niños_j y ellos_j cayeron*.

Dentro de esta información pragmática, también podemos incluir la información referente a la construcción del discurso en el que se desarrolla la anáfora. Esta información se engloba dentro de la denominada *teoría del foco del discurso*, y en determinadas situaciones es útil conocerla, como en la siguiente: *Juan_i era un buen amigo tanto de Luisa_j como de Ana_k; Luisa_j era especialmente encantadora. Él_i estaba secretamente enamorado de ella_j*.

Sin duda alguna, actualmente la información más difícil de manejar es la correspondiente a la del mundo exterior e inferencia de conocimiento, ante la posible representación a utilizar y el volumen que necesitaría, por lo que es un punto todavía en desarrollo. Por ello, tal y como ya hemos comentado, los algoritmos se están centrando en restricciones y preferencias heurísticas que emplean información morfológica, sintáctica y semántica. Sin embargo, estas aproximaciones funcionan relativamente bien. Un estudio reciente reveló que un algoritmo sencillo para la resolución de pronombres no reflexivos basado en criterios morfosintácticos y sintácticos obtenía unos porcentajes de éxito superiores al 80%. Estos resultados se han confirmado en el ya citado trabajo de Lappin y Leass [95].

En la literatura actual aparecen diversas versiones de manejo de estas fuentes de información según el posible valor que se les aplique. Hay un punto común entre la

mayoría de ellas y es la distinción entre *restricciones* y *preferencias*. Las *restricciones* tienden a ser absolutas, y como tales eliminan antecedentes factibles de una determinada anáfora. Por otro lado, las *preferencias* tienden a ser relativas, por lo que requieren el uso de información adicional y tienen carácter consultivo.

También conviene distinguir entre dos conceptos: *resolución* y *generación de la anáfora*. La primera viene a buscar la entidad a la cual se hace referencia, mientras que la segunda lo que hace es crear una referencia sobre una entidad del discurso. El objeto de este trabajo se centrará en el primero de estos conceptos, la resolución de la anáfora, y más concretamente en su resolución sobre textos escritos, es decir, no trataremos la anáfora que se produce en el lenguaje hablado.

Esta Tesis se organizará en ocho capítulos. Aparte del capítulo actual introductorio, en el siguiente capítulo se desglosarán los conceptos y definiciones necesarios para alcanzar una visión general del alcance del problema que aquí tratamos: la resolución de la anáfora. En el capítulo tres se revisan las estrategias de tratamiento de la anáfora que se están aplicando en la actualidad. A partir del capítulo cuatro se presentan nuestras aportaciones a la resolución de este problema. En primer lugar, se introduce un sistema implementado en Prolog basado en el formalismo gramática de unificación de huecos (SUG, Slot Unification Grammar), el cual supone una extensión de las gramáticas de cláusulas definidas (DCG, Definite Clause Grammar). En el capítulo número cinco se expone el método de resolución que se integra en el anterior sistema, mostrando los resultados obtenidos en el capítulo seis. Finalizaremos con un capítulo dedicado a conclusiones y las referencias bibliográficas. Además se adjuntan una serie de apéndices, en los que se muestran los resultados obtenidos en los experimentos realizados, así como los corpus y gramática utilizados en estos experimentos, y el programa Prolog que los lleva a cabo.

En este capítulo introductorio continuaremos en la siguiente sección presentando de manera breve diversos puntos de conexión entre la resolución de la anáfora con otros temas de estudio del procesamiento del lenguaje natural: la elipsis, la extracción de información, la traducción automática y la resolución del ligamiento de sintagmas preposicionales. Mediante estas relaciones con otros campos pretendemos reseñar la importancia que tiene en la actualidad un correcto tratamiento de la anáfora.

1.2 Relación de la anáfora con otros temas de estudio

1.2.1 Elipsis

La *elipsis* es un fenómeno que también ha sido tratado ampliamente. Existen diferentes definiciones de la elipsis, aunque todas comparten un significado común en el que se la considera como un mecanismo de economía lingüística, o sea, una forma de suprimir elementos repetidos de un texto que quedan sobrentendidos para el receptor del mismo. Por

ejemplo, en [64] Franchini define la elipsis como el fenómeno o procedimiento de la omisión de elementos sintagmáticos (nucleares) en el discurso. Según Allen [3] la elipsis implica el uso de oraciones que son incompletas sintácticamente. Covington en [42] habla de elipsis como la omisión de palabras repetidas dentro de la frase.

La *anáfora* tiene también una visión similar a la elipsis ya que también se trata de otro mecanismo de economía lingüística, salvo que en este caso aparecen elementos que nos indican la supresión de los elementos repetidos. Por ejemplo, en [80] Hirst define la anáfora como el mecanismo para hacer en el discurso una referencia abreviada de una o varias entidades, con la esperanza de que el receptor del discurso sea capaz de determinar la identidad de la referencia. Y en [143] Rico define la anáfora de manera general como la relación de referencia que se establece entre una forma lingüística y un objeto, una persona o una situación que ya han sido mencionados de manera implícita o explícita con anterioridad durante el proceso comunicativo y que llamaremos antecedente.

De ambas definiciones de elipsis y anáfora, se desprende que ambas son formas de economía lingüística, diferenciándose en que la primera supone una *omisión* de elementos repetidos, mientras que la segunda representa una *sustitución*. Así tendríamos como ejemplo de elipsis²: *Mi amigo me saludó cuando Ø entró*, donde Ø indica la posición donde se recuperaría el sintagma nominal con función de sujeto que ha sido elidido, *mi amigo*. Y como ejemplo equivalente de anáfora³: *Mi amigo_i me saludó cuando él_i entró*, en el que se produciría la sustitución del pronombre personal *él* por el sintagma nominal *mi amigo*. La diferencia entre ambas frases es que en la primera no aparece ninguna entidad lingüística que deba ser vinculada con un antecedente mediante una relación de correferencia, sino que simplemente se deja un vacío en la estructura sintáctica de la frase.

En [64] Franchini comenta que en casos en que por razones gramaticales la supresión total (elipsis) de elementos repetidos es imposible, el sistema de la lengua ofrece otra forma de economizar menos radical que la elipsis a través del procedimiento de pronominalización (anáfora), que no es la sustitución por cero sino la sustitución por un elemento deíctico o pro-forma. Así, en: *Juan enjabona al bebé*, y *María lo, seca*, equivale a decir: *Juan enjabona al bebé y María seca al bebé*, lo que supondría una repetición molesta. Además aquí la elipsis no sirve a este propósito⁴: **Juan enjabona al bebé y María seca Ø*.

Otra diferencia en cuanto a la forma de economía lingüística que suponen la elipsis y la anáfora es que todos los casos de elipsis de contexto lingüístico, sean de la clase que sean, se limitan a las categorías gramaticales del nombre sustantivo y el verbo finito (luego no se incluyen los adjetivos y adverbios). Los modificadores y complementos por sí solos nunca pueden constituir el elemento elíptico de una construcción, y sólo podrán sustituirse por una

² A partir de ahora denotaremos mediante el símbolo Ø la posición que debería ocupar el elemento elidido.

³ Recordamos que denotaremos mediante subíndices alfanuméricos la relación de referencia entre la expresión anafórica y su antecedente, y entre corchetes los antecedentes o expresiones anafóricas formados por varias palabras.

⁴ En este trabajo las frases incorrectas tanto desde el punto de vista gramatical, como semántico, como desde el punto de vista de la resolución de la anáfora irán precedidas por un *.

proforma: *Juan se comió el bocadillo_i*, y *María se lo_i llevó*, ya que sería errónea la versión elíptica: **Juan se comió el bocadillo* y *María se llevó Ø*. Otra cosa es que también aparezcan elididos junto al núcleo algún adyacente del mismo: *Juan canta bien* y *Pedro Ø también*. *Juan bebió vino* y *Pedro Ø también*. *Juan come el arroz con cuchara* y *Pedro Ø con tenedor*. Sin embargo, la omisión de un adyacente sin su núcleo correspondiente no sería correcta: **Juan canta bien* y *Pedro baila Ø*. ($\emptyset = \text{bien}$). **Juan bebe el vino* y *Pedro se lleva Ø*. ($\emptyset = \text{vino}$). Este último ejemplo entraría dentro de la pronominalización: *Juan bebe el vino_i*, y *Pedro se lo_i lleva*.

Un nuevo rasgo que comparten elipsis y anáfora, es que tal y como describe Franchini en [64] ambas suceden en tres contextos: situacional, convencional y lingüístico, que pasamos a describir brevemente a continuación:

- ◆ En el contexto situacional los elementos elípticos o antecedentes anafóricos se deducen de la situación concreta. Por ejemplo, en la situación en la que se ofrece a alguien una manzana, acercándole la fruta en la mano, tendríamos un caso de elipsis en: *Toma*, y un caso de anáfora en: *Tómala*.
- ◆ Contexto convencional: en *Una tortilla a la francesa*, por una convención tácita se sabe que *a la francesa* es una construcción elíptica de *a la manera francesa*. En *La mayúscula*, también se hace referencia convencionalmente a *La letra mayúscula*.

En los casos de anáfora se trata de fórmulas fijas, lexicalizadas, con evidente pronominalización en las que resulta prácticamente imposible restituir exactamente el elemento a que el pronombre hace referencia: *darla con queso*, *arreglárselas*, *pasarlo bien*.

- ◆ La elipsis que se produce en un contexto lingüístico, tal y como ocurre en la frase *Juan canta bien* y *Pedro Ø mal*, realiza la supresión del elemento nuclear del segundo miembro con la finalidad de evitar una repetición, ya que dicho elemento nuclear sería idéntico al del primer miembro. La anáfora también se puede producir en este contexto, como en el caso de la frase *Juan enjabona al bebé_i*, y *María lo_i seca*.

Otra característica común es que tanto en la elipsis como en la anáfora existe la posibilidad de aparición del componente omitido después de la posición que ha de ocupar, en cuyo caso estaremos hablando de *elipsis catafórica* o *catáfora*:

- ◆ Elipsis catafórica: *Si Ø gana en la lotería, Juan se compra un piano*.
- ◆ Catáfora, que tal y como define Covington en [42] agrupa los casos en que la expresión anafórica aparece antes que el antecedente al cual se refiere: *Cerca de él_i, Juan_i vio una serpiente*.

Y también ambas se desarrollan tanto en el marco de una sola oración como entre diferentes oraciones. Para el caso de la elipsis, tal y como se expone en el trabajo de Palomar [127]:

- ◆ Caso intraoracional. Se refiere a las elipsis producidas en el interior de una frase simple o coordinada *Juan aprobó Física* y *Pepe Química*. En estos, los elementos vacíos se controlan por elementos paralelos del primer miembro: *Pedro vendrá hoy* y *Marta mañana*. Si los elementos del primer miembro fueran vacíos, entonces se controlarían por elementos externos a la estructura coordinada o por

elementos desplazados al extremo derecho de la estructura: *Pedro compra y Luis vende coches viejos*.

Para el caso de la anáfora, ocurrirá cuando tanto el antecedente como la forma lingüística se encuentran en la misma oración. Un ejemplo podría ser el ya comentado anteriormente: *Juan enjabona al bebé_i y María lo_i seca*.

- ◆ **Caso interoracional.** En los casos en que se implican varias frases: *Algunos piensan que Pedro ganará la carrera la semana que viene. Pero Ø no lo conseguirá*.

Para la anáfora, ocurrirá cuando se supere el marco de una sólo oración, tal y como ocurre en el siguiente texto: *Me saludó una chica_i. La_i conocía, pero no recordaba su nombre*.

Otro detalle que resalta la necesidad de un tratamiento conjunto de la elipsis y la anáfora es que se pueden dar casos en que aparezcan más de una elipsis y/o anáfora en el mismo texto: *Cuando las conocí, Marta tenía 15 años y Rosario tenía 14 Ø. Cinco años después la primera tenía 20 Ø y la segunda Ø 19 Ø y ya Ø se había casado con el hijo de un rico propietario de por allí*.

El tratamiento de la *elipsis verbal* también puede tener una parte común con la *anáfora verbal*, la cual ocurre cuando la omisión viene determinada por una proforma como ocurre en el caso de la frase *Juan cantó bien y Pedro lo hizo mal*. Esta construcción sería equivalente a la siguiente elipsis verbal: *Juan cantó bien y Pedro Ø mal*.

Un problema cuya resolución podría encuadrarse tanto dentro de la elipsis como de la anáfora es el de la omisión del sujeto de las oraciones cuyo verbo adopta forma pronominal. En [23] Brucart destaca que una de las diferencias más notables en cualquier intento de comparación del español con lenguas tan próximas como el francés o el inglés radica en la posibilidad que presenta el español de elidir el sujeto de estas oraciones. Por ejemplo, en la frase *Como peras*, el sujeto es un pronombre personal en primera persona que está elidido, situación que también se puede dar con oraciones coordinadas: *Juan entró en la habitación y Ø llevaba un papel en la mano*, donde el sujeto de la segunda cláusula coordinada sería otro pronombre personal en tercera persona, sin conocimiento completo de género (masculino o femenino: *él* o *ella*).

Hay un principio general que obliga a elidir el pronombre en las oraciones coordinadas si éste representa al sujeto del conjunto precedente: *Luis_i llegó a casa y él_i habló con Teresa*. Sin embargo en determinadas circunstancias se admite la presencia del pronombre intensivo *él mismo*, como sucedería en la anterior frase: *Luis_i llegó a casa y [él mismo]_i habló con Teresa*.

En inglés también se pueden encontrar casos de este tipo aunque son los menos, tal y como Hirst muestra en [80] que considera algunos casos de elipsis como anáforas, indicando que algunas expresiones anafóricas son completamente vacías. Por ejemplo: *Ross carefully folded his trousers and Ø climbed into bed*.

Hay dos posturas para describir este fenómeno de la omisión pronominal:

- ◆ Para algunas gramáticas estas construcciones son elípticas en el sentido de que en el análisis sintáctico debe suponerse la presencia de un sujeto tácito o callado.

- ◆ Por el contrario, ciertos autores consideran que este tipo de oraciones no presentan defectividad o elipsis alguna, por cuanto el sujeto está presente ya en la morfología del verbo.

En nuestra opinión, podemos distinguir en su resolución tanto una parte de elipsis como una parte de anáfora, es decir, ha de desencadenarse un proceso de recuperación del sujeto pronominal elidido (resolución de la elipsis) para de este modo obtener la información morfológica (género, persona y número) correspondiente a la forma pronominal (anáfora).

Hay otro fenómeno lingüístico en el que tampoco está tan clara la separación entre anáfora y elipsis: la omisión del núcleo del sintagma nominal. Este problema ya lo hemos tratado en Ferrández et al [57], trabajo en el que proponemos un algoritmo para su resolución. Como ejemplo se puede analizar la siguiente frase: *Pedro toma un zumo de naranja y Luis toma uno Ø de pera*. En esta frase, aunque es indiscutible que se encuentra elidido el núcleo del sintagma nominal, *zumo*, algunos autores consideran que en realidad se trata de una anáfora, en la que el determinante *uno* es ascendido a la categoría de pronombre viniendo a sustituir a *un zumo*. Por ejemplo, en [64] Franchini considera que existe elipsis, contrariamente a lo que opinan Alcina y Blecua en [2] y [140] que abogan por sustantivizar estos determinantes. También en [7] Arrarte considera un fenómeno característico del español la presencia del determinante con carácter pronominal en sintagmas nominales en que el nombre es omitido: *el de todos los días* (pan), *los de Madrid* (habitantes), *el que vino ayer* (chico). En la gramática propuesta por Arrarte basada en el formalismo gramática de huecos, se considera que el núcleo de estas frases es el determinante modificado por un sintagma preposicional o por una cláusula de relativo. Sin embargo en los casos en que aparece acompañado de un adjetivo o participio nominalizado, se considera como núcleo a dicho adjetivo o participio: *el caro*, *los mencionados*.

Para el desarrollo del algoritmo para la resolución de este problema propuesto por nosotros en [57] nos hemos sumado a la propuesta de no sustantivizar ningún componente del sintagma nominal incompleto. Es decir, hemos considerado que ni el determinante de estos sintagmas nominales se analiza como pronombre, ni tampoco los sintagmas adjetivales se ven ascendidos a realizar la función de núcleo del sintagma nominal. El motivo de esta elección es porque el hecho derivado de admitir el valor pronominal de estas unidades lleva aparejado el duplicar el sistema de los pronombres que pueden funcionar como núcleo de un sintagma, punto que complicaría en gran medida el diseño de la gramática.

En nuestra propuesta [57], mediante el formalismo gramática de unificación de huecos (SUG, Slot Unification Grammar⁵) hemos definido subtipos del sintagma nominal en los que no se ha rellenado su núcleo (el nombre), y sí aparecen un determinante y otros modificadores del nombre, como un sintagma adjetival, preposicional o una oración de relativo especificativa. Por ejemplo, *ese azul* lo consideraremos un subtipo del sintagma nominal y lo denominaremos de tipo adjetivo. Estos subtipos de sintagma nominal los detectaremos, tal y como mostraremos más adelante en este trabajo, y estableceremos un proceso de búsqueda del núcleo del mismo de igual modo que si de una anáfora se tratase. Este proceso se realizará tras la fase de análisis sintáctico, trabajando sobre una estructura que aglutine toda la información sintáctica, morfológica y semántica que se necesita para su

⁵ Este formalismo lo desarrollaremos en profundidad posteriormente en el capítulo 4.

resolución, tal y como se mostrará en el capítulo 5. En la Figura 1 se muestra un extracto de las reglas gramaticales propuestas para determinar la formación de sintagmas nominales: una correspondiente al sintagma nominal completo que tiene como núcleo un nombre (sintagma nominal *tipoSustantivo*) y otra en la que el núcleo aparece elidido (sintagma nominal *tipoAdjetivo*).

```

sn (Numero, Genero, terc, tipoSustantivo, Funcion) ++>
  <<det(Numero, Genero)>>, <<adyAdj(Numero, Genero)>>,
  nombre(Numero, Genero, Tipo),
  << sp(sn, ISP) >>, << oracionRelativo >> .

snSimplePodado(Numero, Genero, terc, tipoAdjetivo, Funcion) ++>
  det(Numero, Genero), adyAdj(Numero, Genero),
  << sp(sn, ISP) >>, << oracionRelativo >> .

```

Figura 1. Reglas gramaticales *SUG* para la resolución de la omisión del núcleo del sintagma nominal.

Una de las ventajas de posponer esta fase de recuperación de los elementos elididos a una fase posterior al análisis sintáctico es la de permitir el tratamiento de los casos en que se hallen los dos sintagmas nominales en posiciones no paralelas, ya sea, dentro de la misma oración o no:

- ◆ Caso intraoracional: los dos sintagmas nominales están dentro de la misma oración:
 - ◇ Pertenecen a un sintagma común: *El hijo de Juan y el Ø de Pedro han pegado a un chico.*
 - ◇ Pertenecen a sintagmas diferentes: *El hijo de Juan ha pegado al Ø de Pedro.*
- ◆ Caso interoracional: los dos sintagmas nominales se hallan en oraciones diferentes:
 - ◇ Con la misma función: *El hijo de Juan va de excursión. El Ø de Pedro se queda en casa* (función de sujeto).
 - ◇ Con distinta función: *Hemos pillado al hijo de Juan. El Ø de Pedro se ha escapado* (función de objeto directo y sujeto).

Como resumen de esta subsección en la que hemos comparado los fenómenos de la elipsis y la anáfora, podemos concluir que tienen características comunes que permiten e incluso aconsejan un tratamiento conjunto, tal y como se propone en el trabajo de Hardt [76] donde se tratan los casos de elipsis verbal con referencias, por ejemplo en el siguiente texto: *When Harry drinks_i, I always conceal [my belief that he shouldn't Ø]_j. When he gambles, I can't conceal it_j.*

1.2.2 Extracción de información

Una de las aplicaciones del procesamiento del lenguaje natural, con amplia relación con la resolución de la anáfora es la conocida como *extracción de información* (*information*

extraction). Este campo tal y como se describe en Cowie y Lehnert [43] se diferencia de la *recuperación de información (information retrieval)*, en cuanto que ésta última supone el proceso de recoger material útil desde vastos campos de material en bruto, representando una posible primera fase de la extracción de información, en la cual con grandes cantidades de información potencialmente útil, podríamos transformar ese material en bruto, refinándolo y reduciéndolo en forma de plantillas. La extracción de información tiene numerosas aplicaciones potenciales. Por ejemplo, la información disponible en texto no estructurado podría ser trasladada a bases de datos tradicionales que los usuarios pasarían a probar mediante consultas estándar.

La resolución de la anáfora y el problema de la extracción de información comparten problemas comunes relativos al análisis del discurso. Los problemas que abarca el análisis del discurso se pueden resumir en estos tres puntos fundamentalmente:

- ◆ El análisis de oraciones nominales, referente al problema de reconocer sintagmas nominales complejos y el análisis semántico de los mismos.
- ◆ La resolución de correferencias, es decir, el problema de reconocer cuándo un nuevo sintagma nominal está haciendo referencia a una entidad previa, y cuándo es en realidad una nueva entidad del discurso.
- ◆ El reconocimiento de enlaces relacionales que se necesitan para estructurar los tokens de memoria en una red asociativa que contiene enlaces que se saben importantes para el soporte de los requerimientos de la extracción de información.

De los tres problemas nombrados, sin duda el segundo es, con diferencia, el más importante, ya que supone el proceso de resolución de anáforas en sí mismo. Las entidades importantes del discurso pueden mencionarse varias veces y puede que nunca sean descritas dos veces con el mismo sintagma nominal. La resolución de la correferencia es sensible a las características estructurales de todo el texto, así como a las características semánticas, el correcto análisis de la oración, y el análisis de sintagmas nominales complejos. Cualquier propagación de errores asociada con la identificación de estas características es rápidamente visible cuando intentamos reconocer sintagmas nominales correferentes.

Otro tema común en la extracción de información y la anáfora es el del análisis sintáctico parcial del texto. Este análisis parcial se refiere a un estilo de análisis de oraciones que no requiere un análisis sintáctico total de cada palabra del texto. El motivo de aparecer diversos trabajos que abogan por este estilo de análisis sintáctico es porque el análisis completo de cada oración en cada texto es demasiado exigente computacionalmente a la hora de obtener altos niveles de rendimiento. Los investigadores se encuentran atrapados dentro de un ciclo de desarrollo del sistema demasiado lento para soportar una adecuada experimentación y bucles de realimentación. Los analizadores sintácticos de oraciones operan habitualmente en tiempo polinomial y tienden a atascarse con oraciones de más de veinte o treinta palabras. Esto ha motivado la revisión de las líneas de investigación que necesitan árboles de análisis sintáctico completos. Por ejemplo, podemos encontrar tendencias en las que se separan oraciones relevantes de las que no lo son, reduciendo la cantidad de datos a procesar. Otros relajan sus restricciones sintácticas para producir fragmentos o árboles sintácticos incompletos. Y finalmente, otros han abandonado sus analizadores completos para conseguir verdaderos analizadores parciales. Nosotros aplicaremos un enfoque similar para el tratamiento de la anáfora puesto que realizaremos un análisis sintáctico parcial, analizando tan sólo los constituyentes que nos interesan para la resolución de referencias desechando los demás. Por ejemplo, en [87] Kameyama muestra

un algoritmo para el tratamiento de la anáfora que se integra dentro del sistema de extracción de información FASTUS [83], en el que no se realiza un análisis sintáctico completo del texto, justificándolo tanto desde el punto de vista computacional como desde el teórico con los problemas de resolución de la ambigüedad.

Para finalizar esta breve revisión de la relación entre anáfora y extracción de información, tan sólo mostrar el enfoque elegido en el sistema LaSIE desarrollado en la Universidad de Sheffield [66]. LaSIE es un sistema integrado que construye un modelo de un texto que puede utilizarse en diversas tareas, como por ejemplo para obtener resúmenes en lenguaje natural. En cuanto al análisis sintáctico, este sistema utiliza un analizador ascendente tipo *chart* implementado en Prolog. La gramática empleada se incluye dentro del grupo de las gramáticas incontextuales basadas en la unificación. El análisis sintáctico se realiza en dos fases, cada una de ellas empleando una gramática diferente. Así en una primera fase se utilizarán únicamente reglas gramaticales para identificar sintagmas nominales relevantes; y en la segunda y última fase se empleará una gramática más general que analizará el resto del texto. Respecto al módulo de resolución de la anáfora, este sistema trabaja oración tras oración después de realizar el análisis sintáctico y semántico, momento en el que compara las entidades que aparecen en la nueva oración con las que ya tenía almacenadas anteriormente, y en el caso que haya correferencia las funde en una sólo entidad.

1.2.3 Traducción automática

Tal y como cuenta Mitkov en [112], las aplicaciones de traducción automática también necesitan un correcto tratamiento de la anáfora con el objetivo de conseguir una traducción global más compacta a nivel de discurso en lugar de una traducción frase a frase. En esta traducción global se presentan dos alternativas, la primera consistiría en que la traducción tenga resueltas las anáforas (en la traducción en lugar de aparecer la expresión anafórica se pondrá su antecedente), y la segunda sería que se tradujesen las expresiones anafóricas explícitamente al lenguaje destino (por ejemplo traducir el pronombre *he* por *él*). La elección entre una u otra alternativa es una tarea importante dentro del módulo de tratamiento de la anáfora de un sistema de traducción automática.

El problema de la resolución de la anáfora introducida en el marco de la traducción automática revela una complejidad adicional debido fundamentalmente a las siguientes cuestiones: discrepancias de género entre diferentes lenguajes, discrepancias de número entre palabras que denotan el mismo concepto, discrepancias en género y número entre los pronombres, y finalmente, discrepancias entre los procesos de tratamiento de la anáfora en un lenguaje y en otro.

Estas discrepancias en el tratamiento de los pronombres de diferentes lenguajes se hace más explícita en caso que los pronombres de un determinado lenguaje no conserven la información correspondiente al número y género, por lo que si en el lenguaje origen aparece un pronombre con esta información, al traducirlos sería conveniente resolverlos previamente ya que de otro modo esta información se perdería. Una situación de éste estilo ocurriría en la traducción del inglés a coreano, en la que un pronombre inglés podría no aparecer en la frase (elipsis), o bien se podría traducir por su antecedente, o bien se podría traducir por uno o dos pronombres coreanos en función de la información sintáctica y rasgos

semánticos del núcleo del sintagma nominal al cual se refiere. Otro ejemplo más cercano a nosotros sucedería en la traducción español-inglés, en la que existe la ya comentada omisión del sujeto pronominal. De este modo, en la siguiente frase: *Marta está muy cansada. Ø Ha estado trabajando todo el día*, en la traducción al inglés de la segunda oración tendríamos dos posibles pronombres ingleses a elegir, ambos en tercera persona singular pero con género masculino o femenino, *he* o *she*: *Marta is very tired. (He / She) has been working all day*. Algo similar ocurre con los denominados *pronombres japoneses "cero"* (*Japanese Zero Pronouns*, tratados en numerosos trabajos por ejemplo el de Nakaiwa y Shirai [124]). Estos pronombres suelen omitirse en japonés cuando ocupan la función de sujeto u objeto, mientras que son obligatorios en inglés.

Un ejemplo en cuanto a la discrepancia en número entre palabras de diferentes lenguajes que denotan el mismo concepto podríamos encontrarlo en la siguiente palabra: *people*, es decir, *gente*. Esta palabra en inglés tiene número plural mientras que en español tiene número singular, tal y como se puede apreciar en la siguiente frase: *People are fed up with the situation* \Rightarrow *La gente está harta de la situación*. Evidentemente este hecho debería tenerse en cuenta cuando se nos presente un pronombre que referencie a alguna de estas entidades.

1.2.4 Resolución del ligamiento de sintagmas preposicionales

Hay algunos trabajos que estudian la relación entre el problema de la resolución del ligamiento de sintagmas preposicionales y la resolución de la anáfora, como por ejemplo en [8] Azzam propone un algoritmo para coordinar el proceso de resolución de la anáfora con el de la desambiguación del ligamiento de sintagmas preposicionales. Podemos observar uno de estos problemas en la frase: *Juan observaba al chico sobre el tejado*, en la que existe la ambigüedad por la que el sintagma preposicional *sobre el tejado* modifique o bien al verbo o bien al sintagma nominal *el chico*, ambigüedad que sería importante de determinar antes de resolver posteriores anáforas que se refieran a este sintagma nominal.

El motivo por el que Azzam propone un algoritmo que afronte ambos problemas de manera conjunta es porque considera que existe información común necesaria para la solución de ambos problemas de manera separada, es decir, intentar aprovechar la salida de cada componente en la solución del otro. Según nombra el mismo Azzam, la resolución de la anáfora y la resolución del ligamiento de un sintagma preposicional son de las ambigüedades más frecuentes en el procesamiento del lenguaje natural, por lo que sería interesante un tratamiento conjunto o coordinado puesto que pueden necesitar de información común. Azzam propone un algoritmo de resolución de la anáfora basado en la teoría del foco del discurso⁶, el cual se aplica sobre la representación conceptual de la frase y tiene como salida un conjunto de posibles antecedentes. En cuanto a la resolución del ligamiento de los sintagmas preposicionales, las preposiciones que no están ligadas suponen

⁶ Esta teoría la veremos en el apartado 3.2.2.4 de este trabajo.

roles⁷ vacíos en esta representación conceptual, por lo que el objetivo del algoritmo será el completar estos huecos.

El algoritmo propuesto por Azzam se aplica sobre el texto oración tras oración, y de forma resumida se puede comentar que supone una repetición de la aplicación alternada de los módulos de resolución de estos dos problemas. Por ejemplo, el módulo de tratamiento de la anáfora se saltaría una determinada expresión anafórica cuando esté precedida por una preposición sin ligar ya que puede haber un vacío en la estructura conceptual de la frase. El tratamiento de esta expresión anafórica se pospone a una siguiente fase en la que se vuelve a aplicar este módulo de tratamiento de la anáfora, pasando antes a ejecutar el módulo de resolución del ligamiento. Es decir, el algoritmo consiste en sucesivas llamadas al módulo de resolución de la anáfora y de la resolución del ligamiento de sintagmas preposicionales. La salida de cada llamada supondrá un conjunto de estructuras conceptuales que representan los resultados intermedios intercambiados entre cada módulo. El objetivo es rellenar los roles vacíos en cada estructura conceptual.

En los casos en que se encuentre una expresión anafórica antes de un sintagma preposicional considera que primero se ha de aplicar la resolución de la anáfora y después el del ligamiento, tal y como ocurriría en el siguiente ejemplo propuesto por Azzam: *The sale of Credito was first proposed last August and that of BCI late last year.* Sin embargo, cuando ocurre la anáfora después de sintagmas preposicionales sin ligar podría tratarse de una anáfora intraoracional, entonces su resolución puede depender de uno de los anteriores sintagmas preposicionales. En este caso se tratará la anáfora en una llamada posterior. También, cuando la anáfora está incluida en un sintagma preposicional la resolución del ligamiento necesitará información semántica sobre el objeto del sintagma preposicional. Cuando este objeto es un pronombre esta información semántica no está disponible, por lo que no se podrán aplicar las reglas de ligamiento, con lo que se tendrán que resolver primero los pronombres anafóricos para determinar la clase semántica del objeto al que se refieren.

En el ejemplo propuesto por Azzam: *UPHB shares have been suspended since October 29 at the firm's request following a surge in its share price on a takeover rumour,* el pronombre *its* no se puede resolver porque está precedido por un sintagma preposicional sin ligar por lo que su resolución se pospondría. Entonces se llamaría al módulo de resolución del ligamiento de los sintagmas preposicionales *since October 29* y *at the firm's request.* El ligamiento del sintagma preposicional *in its* se realizará después de la resolución del pronombre anafórico *its* siguiendo la regla comentada en el párrafo anterior. De este modo, se consigue la información semántica necesaria para tratar el ligamiento. Como resumen, una vez ligados los sintagmas preposicionales *since* y *at* se resolvería la anáfora *its*, y finalmente se volvería a llamar el módulo de resolución del ligamiento para los sintagmas preposicionales *in* y *on*.

⁷ Por estos roles Azzam entiende los roles temáticos de la oración: *agente, tema, instrumento, o localización.* Por ejemplo en *Juan rompió la ventana con un martillo,* el sujeto *Juan* ocupa el rol de agente, *la ventana* ocupa el de tema y *con un martillo* el de instrumento.

2. Definición de conceptos

En este capítulo realizamos un completo estudio de los conceptos y definiciones necesarios para poder abordar el problema de la resolución de la anáfora. Mediante este estudio pretendemos centrar el objetivo principal de esta Tesis, delimitando los tipos de anáfora que nos planteamos afrontar, y determinando en qué consistirá exactamente esta resolución.

Estos conceptos los organizaremos en tres secciones. En la primera se intenta centrar la definición de anáfora y el problema que supone su resolución, mientras que en la siguiente se presentan varias clasificaciones de la anáfora que permiten distinguir diferentes tipos de anáfora referenciados en la literatura existente sobre este fenómeno lingüístico. Estas clasificaciones se realizarán en función del marco en que se trata la anáfora, en función de la accesibilidad del antecedente, en función del tipo de referencia que se establezca entre la expresión anafórica y su antecedente, y finalmente en función del tipo de expresión anafórica. Se finaliza con una sección dedicada a centrar los objetivos que se plantea abordar con este trabajo, delimitando el tipo de anáfora a tratar.

2.1 Definición de anáfora

En la literatura actual sobre la anáfora podemos encontrar diferentes definiciones, aunque en todas ellas subyace el mismo contenido y es el modo en que la anáfora nos permite hacer referencia a una determinada entidad⁸, a la cual llamaremos *antecedente* o *referente*. A la referencia abreviada se la llama *expresión o elemento anafórico*. Por ejemplo, tomemos la definición que nos ofrece Hirst en [80]:

La anáfora es el mecanismo que nos permite hacer en un discurso una referencia abreviada a alguna entidad o entidades, con la confianza de

⁸ Por *entidad* entendemos cualquier objeto o persona que aparece implícita o explícitamente en el proceso comunicativo.

que el receptor del discurso sea capaz de desabreviar la referencia y por consiguiente determinar la entidad a la que se alude.

Franchini en [64] añade a esta definición algo importante en cuanto que nos permite diferenciar el fenómeno de la anáfora respecto al de la elipsis:

La pronominalización (un caso especial de anáfora) no es la sustitución por cero (caso de la elipsis), sino la sustitución por un elemento deíctico o proforma.

Detalle que se hace más evidente en la definición dada por Covington en [42]:

La anáfora es el uso de palabras especiales que representan a individuos, situaciones u otras cosas ya referenciadas anteriormente.

La definición dada por Rico en [143] nos resalta otra característica importante, y es que no es necesario que la entidad a la que se hace referencia esté mencionada explícitamente en el discurso:

La anáfora de manera general se define como la relación de referencia que se establece entre una forma lingüística y un objeto, una persona o una situación que ya han sido mencionados de manera implícita o explícita con anterioridad durante el proceso comunicativo.

Esta posibilidad de que no aparezca explícitamente en el texto el antecedente, determina la anáfora que se desarrolla en el contexto denominado *situacional*. Éste hecho nos sirve para introducir el siguiente matiz que distingue los diversos contextos o marcos en los que se puede desarrollar la anáfora. Estos contextos son el *convencional*, *situacional* o *lingüístico*:

- ◆ La anáfora desarrollada en un *contexto convencional* se trata de fórmulas fijas, lexicalizadas, con evidente pronominalización en las que resulta prácticamente imposible restituir exactamente el elemento a que el pronombre hace referencia: *darla con queso, arreglárselas* o *pasarlo bien*.
- ◆ En el *contexto situacional* los antecedentes de una determinada anáfora se deducen de la situación concreta y no aparecen explícitamente en el texto. Por ejemplo, en una situación concreta en la que tenemos delante de nosotros un grupo de objetos, al pronunciar la frase *Deme éste*, señalando con el dedo hacia uno de estos objetos, el antecedente será el objeto señalado, el cual no ha sido nombrado de manera explícita.
- ◆ Y finalmente en el *contexto lingüístico* los antecedentes sí que aparecen explícitamente en el texto, como por ejemplo en *Juan enjabona al bebé, y María lo, seca*, donde la expresión anafórica formada por el pronombre *lo* se refiere al sintagma nominal *el bebé*. En este trabajo, nos centraremos únicamente en la anáfora que sucede en este contexto lingüístico.

Igualmente en [51] Ersan y Akman distinguen entre la anáfora que sucede en el contexto situacional de la que sucede en el contexto lingüístico, llamándolas *referencias exóforas* y *referencias endóforas* respectivamente. En las referencias exóforas o también denominadas sencillamente como *exófora*, Ersan y Akman indican que se hace referencia a entidades del mundo exterior al mensaje lingüístico. Dentro de las referencias endóforas,

incluirá a la anáfora y la catáfora, ya que según explican Ersan y Akman, el elemento referente nos hace referencia a una entidad interna al mensaje lingüístico. La clasificación que realizan se resume en la Figura 2.

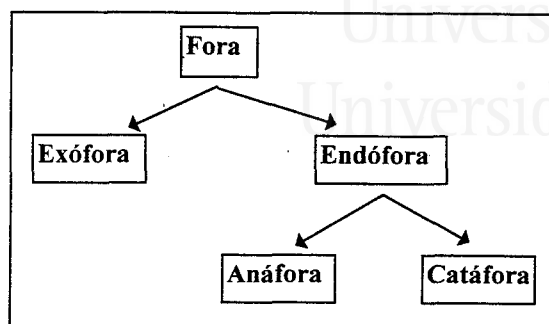


Figura 2. Clasificación de tipos de referencias.

También es conveniente distinguir dentro del problema de la anáfora el caso especial en el que el antecedente aparezca después que la referencia. Este caso especial se denomina *catáfora*, y como ejemplo mostramos el propuesto por Covington en [42]: *Cerca de él_i, Pedro_i, vio una serpiente*, o también: *Como la_i, vieron pasar, Ana_i, se paró a hablar con ellos*. La catáfora, al revés que la anáfora, anticipa o anuncia algo que luego va a nombrarse o decirse. La catáfora suelen realizarla los pronombres: *No lo_i, podrás creer: [Lolita va a casarse]_i*, o también en la frase: *Mi norma es ésta_i: [portarme con los demás mejor que se portan conmingo]_i*, y suele encontrarse el antecedente en la misma oración.

Y finalmente nos queda por definir exactamente en qué va a consistir el *problema de la resolución de la anáfora*. En [144] Ristad lo define como el modo de derreferenciar una expresión anafórica, y según él, la salida de un sistema que la resuelva debe ser una representación del conocimiento que se tiene del objeto al que hace referencia esa anáfora. Veamos a continuación, en qué consiste exactamente este proceso *derreferenciar*. La *referencia* es un concepto central del lenguaje que ha sido estudiado ampliamente. Esencialmente, el problema radica en determinar cómo las palabras son capaces de denotar conceptos y en particular cómo una cierta secuencia de palabras puede denotar un único concepto. Por ejemplo, una persona comenta a otra sin ningún propósito anterior: *El periodista fotografió a Aznar*. Lo más probable es que la persona que escucha la frase en el contexto adecuado sea capaz de determinar que *Aznar* es el presidente del Gobierno y no cualquier otra persona que se llame *Aznar*. Y cuando posteriormente en el texto, utilicemos un pronombre para hacer referencia a esta persona *Aznar*, estaremos realmente referenciando al presidente del Gobierno de una manera indirecta: *él* → *Aznar* → *Presidente del Gobierno*.

Por ello, hablando con más propiedad, una expresión anafórica no es que se refiera a su antecedente, sino al referente de la expresión que sirve de antecedente, por lo que ha de hablarse de *correferencialidad* entre expresión anafórica y antecedente. Es decir, ambos hacen referencia al mismo concepto: la expresión anafórica referencia indirectamente a través del antecedente, mientras que éste último referencia directamente. La correferencialidad suele indicarse poniendo el mismo subíndice al antecedente y al pronombre anafórico: *Mi amigo_i, me saludó cuando él_i, entró*. De este modo, Ristad distinguirá entre *expresiones referentes* (*referring expressions*) y *elementos anafóricos* (*anaphoric elements*), basándose en el nivel de referencia de cada uno. Las expresiones

referentes tienen una referencia directa ya que sus referencias no están mediadas por otros elementos lingüísticos. Mientras que los segundos, como por ejemplo los pronombres, mantendrán una referencia indirecta, ya que se apoyarán en otros elementos lingüísticos (las anteriores expresiones referentes). También destaca Chomsky en su teoría de *rección y ligamiento* (*Government and Binding Theory*) [35][33][34], que los nombres propios se caracterizan por no aceptar relación de correferencia con un antecedente: *Luis, dijo que Luis, vendría. *Luis, insultó a Luis,. Esto es debido a su contenido referencial pleno. Chomsky denomina a esta clase de palabras como *expresiones referenciales*.

Esta noción de correferencialidad, nos permite introducir un cierto matiz que puede hacer una distinción entre los conceptos de *antecedente* y *referente*, tal y como nos reseñan Brown y Yule en [22] y Rico en [143]. Según se define en estos trabajos, el *referente* constituiría la representación mental de los objetos evocados por el texto, mientras que el *antecedente* sería la representación lingüística que estos toman en el mismo. Nosotros en el presente trabajo no distinguiremos entre ambos conceptos, ni tampoco utilizaremos la denominación *expresión referente* o *referencial*, sino que siempre lo llamaremos *antecedente*.

También Hirst en [80], hace hincapié en este concepto indicando que una expresión anafórica y su antecedente se dice que son *correferentes* (*coreferential*), y el proceso de determinar el antecedente de una expresión anafórica se denomina *resolución*. Allen en [3], también utiliza esta definición, ya que cuando se dice que un pronombre tiene un determinado antecedente, realmente se está expresando que ambos se refieren al mismo objeto, es decir, ambos correferen (*co-refer*).

Algunos lingüistas han pensado que en realidad el pronombre anafórico no es más que un sustituto del sintagma nominal al que remite, por lo que el problema de la resolución de la anáfora consiste sencillamente en una sustitución. Sin embargo, esta visión es inadecuada en determinados casos como el que se muestra en la siguiente frase: [*El hombre que la, busca*], *tendrá [la vida que él, desea]*, donde si seguimos el proceso de sustituciones, observaremos que no tiene fin pues nunca se agotarán los pronombres.

Esta visión errónea de la resolución de la anáfora como una sustitución es del mismo modo remarcada por Convington en [42] indicando que la anáfora representa entidades del discurso, y no palabras o frases. Él mismo propone los siguientes ejemplos que ilustran los problemas de considerar la resolución de la anáfora como una mera sustitución: *Si cualquier cliente no paga, muestra su balance*. Aquí indudablemente no sería correcto resolver esa anáfora como: **Si cualquier cliente no paga, muestra el balance de cualquier cliente*. Igualmente en el texto: *Muestra el primer registro. Ahora muestra el siguiente registro. Ahora bórralo*. En la última oración tampoco sería correcto resolver la anáfora como **Ahora borra el siguiente registro*, sino que la referencia sería al registro mostrado en la oración anterior.

Es importante destacar que existe la posibilidad de que la expresión anafórica y su antecedente no establezcan una relación de correferencia, es decir, que *no* sean correferentes. En estos casos la expresión anafórica introduce un nuevo objeto que no aparece explícitamente en el texto, y que se relaciona con otro objeto anteriormente nombrado. Un ejemplo de este tipo de situaciones es la denominada por Allen en [3] como

anáfora superficial (surface anaphora) y la define como aquella que realiza una referencia parcial a parte de un antecedente. Allen propone el siguiente ejemplo⁹:

- A. Tell me [John's grade in CS]₁.
- B. Give me [it in MT]₁ as well.
- C. Give me [Mike's in GG]₁ too.

Los objetos referenciados por las expresiones anafóricas *it in MT* y *Mike's in GG*, hacen referencias a objetos que no han sido mencionados anteriormente de modo completo, sin embargo, el contexto creado por la frase *A* permite interpretarlos de manera correcta.

A modo de resumen, podemos concluir que la solución al problema de la anáfora debe ser una representación del conocimiento que se tiene del objeto al que hace referencia esa anáfora y no una mera sustitución de palabras.

Este problema de la anáfora, puede ser tratado desde diversas ciencias humanísticas, tal y como Rico destaca en [143]. Existen teorías de la anáfora en *Sintaxis*, *Semántica*, *Pragmática*, *Psicolingüística*, *Filosofía del lenguaje*, *Lógica* y *Lingüística Computacional*. Esto es así debido a que los procesos que intervienen en el establecimiento de relaciones anafóricas abarcan un amplio espectro de las ciencias humanísticas:

- ◆ En *Psicolingüística* se llevan a cabo experimentos para descubrir la manera en que los interlocutores consiguen crear y comprender las referencias anafóricas en el proceso comunicativo, o cuáles son los mecanismos que actúan en la selección de determinadas anáforas.
- ◆ En *Filosofía del Lenguaje* se proponen teorías que explican cómo una anáfora puede llegar a establecer una relación con el mundo real a través de la referencia a un antecedente.
- ◆ En el campo de la *Lógica* se crean gramáticas que tratan la anáfora entre otros problemas lingüísticos.
- ◆ En la *Lingüística Computacional* se tiene como objetivo el tratar la información lingüística mediante métodos informáticos a fin de facilitar el tratamiento automático del lenguaje natural. En [107], Meya y Hubert dicen que el objetivo de la Lingüística Computacional es comprender cómo se comunica el ser humano para luego crear modelos y sistemas que capaciten a los ordenadores simular un comportamiento inteligente.

Para finalizar esta importante sección en la que hemos intentado centrar el problema de la anáfora, concluimos que el objetivo de esta Tesis se encuadra en el campo de estudio de la Lingüística Computacional, y que pretendemos tratar con la anáfora que sucede en el contexto lingüístico, buscando como solución la entidad con la que se establece una relación de correferencia en su caso, o la nueva entidad que se introduce en el discurso.

⁹ En este trabajo denotaremos con subíndices numéricos (1, 2, 3, etc.) la relación de *no* correferencia entre una expresión anafórica y su antecedente. En el caso que sí exista correferencia entre ambos utilizaremos subíndices de tipo carácter (*i, j, k*, etc.).

2.2 Tipos de Anáfora

En la literatura actual sobre el tratamiento de la anáfora podemos encontrar diferentes tipos de anáfora: *anáfora intraoracional*, *anáfora discursiva*, *anáfora superficial*, *anáfora profunda*, etc. Estas variaciones miden diferentes matices del concepto original y general de anáfora, por ejemplo en función del marco en que se trata la anáfora, ya sea éste la propia oración o todo el discurso, hablaremos de anáfora intraoracional y anáfora discursiva respectivamente. En función de la accesibilidad del antecedente: *anáfora morfosintáctica*, *semántica* y *pragmática*, de tal modo que la morfosintáctica será la que tenga una mayor accesibilidad, es decir, la que se derreferencia más fácilmente, y la pragmática la que tenga menor accesibilidad. También en función del tipo de referencia que se establezca entre la expresión anafórica y su antecedente hablaremos de *anáfora superficial* y *anáfora profunda*, en la que la profunda establece una relación de correferencia, mientras que la superficial no. Y finalmente podemos encontrarnos distintos tipos de expresiones anafóricas: pronombres, sintagmas nominales definidos, sintagmas nominales formados por el pronombre *uno* (*one*) y algunos modificadores, sintagmas nominales cuyo núcleo sea un adjetivo, adverbios y complementos circunstanciales. Cada una de estas expresiones anafóricas definirá un determinado tipo de anáfora que en ocasiones se denomina de una manera específica: *anáfora pronominal*, *anáfora de tipo "one"*, *anáfora de tipo adjetivo*, *anáfora superficial numérica*, *anáfora verbal* y *referencias temporales o locales*.

Estos matices permiten delimitar la parte del fenómeno anáfora que se intenta resolver. En este caso enlazando con la sección anterior, nosotros trabajaremos sobre la anáfora que sucede en un contexto lingüístico, y dentro de los tipos de anáfora que existen, nos centraremos en la anáfora discursiva, es decir, consideraremos para su resolución no sólo la oración en la que se encuentra la expresión anafórica, sino todo el discurso. Nos centraremos en la anáfora morfosintáctica, o sea, la de mayor accesibilidad de su antecedente. Trataremos tanto referencias superficiales como profundas, tanto relaciones de correferencia como de no correferencia, e intentaremos resolver la anáfora pronominal y la de tipo adjetivo, es decir, expresiones anafóricas de tipo pronombre y sintagmas nominales cuyo núcleo sea un adjetivo.

En esta sección vamos a mostrar detalladamente esta clasificación de la anáfora, dividiéndola en cuatro subsecciones, cada una de ellas se corresponderá con uno de estos criterios de clasificación: en función del marco en que sucede, en función de la accesibilidad del antecedente, según el tipo de referencia y según el tipo de expresión anafórica.

2.2.1 En función del marco en que sucede

En función del marco en el que ocurre la anáfora, ya sea dentro de una oración o dentro de un discurso, podremos hablar de *anáfora intraoracional* o *anáfora discursiva* (*interoracional*).

2.2.1.1 ANÁFORA INTRAORACIONAL

La *anáfora intraoracional* ocurre dentro de la oración, es decir, tanto el antecedente como la expresión anafórica se encuentran en la misma oración. El problema de esta definición es cuando se intenta aplicar la resolución de la anáfora en componentes donde la oración no es un componente aislado, sino uno más dentro de una unidad mayor: el discurso. De este modo, en el caso que se supere el marco de una sola oración estaremos tratando el siguiente tipo, la *anáfora discursiva*.

2.2.1.2 ANÁFORA DISCURSIVA

En la *anáfora discursiva* la expresión anafórica y su antecedente no aparecen en la misma oración. Este tipo de anáfora actúa como un mecanismo de cohesión textual, es decir, la anáfora actúa como un instrumento lingüístico que contribuye a mantener el discurso como una unidad de sentido completa gracias a la creación de vínculos de unión entre diferentes partes del texto. De este modo contribuye a la conservación de un foco de interés previamente establecido mediante las entidades discursivas, las cuales crean unos puntos de atención en el discurso que son confirmados y consolidados gracias a las expresiones anafóricas.

En este trabajo nos centraremos en la anáfora discursiva ya que ésta es un fenómeno lingüístico básicamente discursivo, es decir, que no se puede reducir únicamente al marco de la oración. Como ejemplo mostramos las siguientes frases: *Anoté su dirección_i en un cuaderno. Te la_i daré luego por teléfono. Me saludó una chica_j. La_j conocía, pero no recordaba su nombre. [Hoy hablara el presidente]_k. La radio lo_k ha dicho. Recorrimos la arboleda_m despacio. Los niños no estaban allí_m.*

2.2.2 En función de la accesibilidad del antecedente

Continuando con el citado trabajo de Rico [143], la anáfora también se puede clasificar en tres tipos en función de la *accesibilidad del antecedente*, es decir, la facilidad con la que se puede derreferenciar (acceder al antecedente de una expresión anafórica concreta). Estos tres tipos son la *anáfora morfosintáctica*, *semántica* y *pragmática*, los cuales están resumidos en la Figura 3.

La expresión anafórica típica con accesibilidad alta es el pronombre, mientras que las que tienen accesibilidad más baja habitualmente son las anáforas semánticas y pragmáticas, ya que éstas son las expresiones que tienen mayor carga léxica y aportan mayor información sobre el antecedente, permitiendo antecedentes con menor accesibilidad. De este modo los pronombres habrán de referirse a entidades muy accesibles, ya que de otro modo al tener poca información léxica sería imposible resolverlos, mientras que las anáforas semánticas podrán referirse a antecedentes con menor accesibilidad con ciertas garantías de éxito. En [119] Mollá detalla más esta clasificación, dividiendo en tres grupos principales en función de la accesibilidad: el primer grupo lo formarán los nombres propios y sintagmas nominales definidos, el segundo los demostrativos, y como tercer grupo los reflexivos. El primer grupo será el de menor accesibilidad y el tercero el de mayor.

<i>Accesibilidad</i>	<i>Anáfora</i>
Máxima accesibilidad	<u>Anáfora morfosintáctica:</u> <ul style="list-style-type: none"> ■ <i>Nominal</i>: referencia a un sintagma nominal. ■ <i>Verbal</i>: referencia a un verbo o SV. ■ <i>Oracional</i>
Accesibilidad media	<u>Anáfora semántica:</u> <ul style="list-style-type: none"> ■ <i>Sinonimia</i>. ■ <i>Hiperonimia</i>. ■ <i>Anáfora contextual</i>
Mínima accesibilidad	<u>Anáfora pragmática</u>

Figura 3. Tipos de anáfora en función de la accesibilidad del antecedente.

El grado de accesibilidad de un antecedente se determina mediante la aplicación de cuatro parámetros básicos, tal y como también estudian Ariel [6] y Mollá [119]:

- ◆ *Distancia*, que consiste en el número de oraciones que hay desde la expresión anafórica hasta su antecedente (cuanta más distancia implica que habrá menos accesibilidad).
- ◆ *Prominencia*, según la cual los antecedentes que se encuentran en posiciones temáticas de la frase tendrán una prominencia alta, por lo que serán más fácilmente accesibles.
- ◆ *Competencia*, es decir, el número de entidades discursivas que compiten por ser el antecedente de la expresión anafórica. Cuanta más competencia obviamente existirá menos accesibilidad.
- ◆ *Unidad*: si la expresión anafórica y su antecedente están en el mismo párrafo, entonces habrá un mayor alto grado de accesibilidad.

En los próximos tres apartados estudiaremos en detalle cada uno de estos tipos de anáfora: morfosintáctica, semántica y pragmática.

2.2.2.1 ANÁFORA MORFOSINTÁCTICA: NOMINAL, VERBAL Y ORACIONAL

En la *anáfora morfosintáctica* las relaciones anafóricas se explican mediante criterios morfológicos y sintácticos, y son las que mayor accesibilidad presentan. Dentro de la *anáfora morfosintáctica*, se definen tres subtipos: la *anáfora nominal*, *anáfora verbal* y la *anáfora oracional*, las cuales pasamos a detallar a continuación:

- ◆ La *anáfora nominal* se entiende como aquella en la que se hace referencia a un sintagma nominal: *Tu hijo_i rompió el cristal. Yo le_i vi.*

- ◆ La *anáfora verbal* se construye principalmente en inglés, mediante la llamada eliminación del verbo o *gapping*, y mediante la utilización de un pro-verbo (por ejemplo los auxiliares *do* o *have*). Esta anáfora puede referirse a un verbo o a una frase verbal, como ocurre en la frase: *Pedro [jugó muy bien al tenis ayer], pero Juan [lo hizo], muy mal.*
- ◆ La *anáfora oracional* se construye mediante una expresión anafórica cuyo antecedente es toda una oración, como sucedería en *[No deberíamos salir esta noche], Yo no opino eso.*

En nuestro trabajo nos centraremos en la anáfora morfosintáctica (no trataremos con la anáfora semántica o pragmática), y más concretamente en la anáfora nominal, o sea, trataremos las expresiones anafóricas que tengan como antecedente un sintagma nominal.

2.2.2.2 ANÁFORA SEMÁNTICA: SINONIMIA, HIPERONIMIA, CONTEXTUAL Y FORZADA

En la *anáfora semántica* las relaciones anafóricas sólo pueden explicarse acudiendo a criterios semánticos. Habitualmente en esta anáfora, los antecedentes no están siempre explícitos en el discurso, y por tanto, son antecedentes muy poco accesibles. Dentro de este tipo de anáfora, podemos encontrar las siguientes variaciones:

- ◆ *Sinonimia*: la construcción de relaciones anafóricas a través de la sinonimia es posible cuando existen dos o más equivalentes semánticos que pueden intercambiarse, sin que este cambio introduzca entidades extrañas al lector. El equivalente semántico hace la función de anáfora, mientras que el término sustituido es el antecedente. Por ejemplo, en la siguiente frase: *Pedro se quitó sus gafas, ... Él limpió sus lentes,* aparecen dos sinónimos: *gafas* y *lentes*, realizando el segundo de ellos la función de expresión anafórica, es decir, forma una referencia a la misma entidad nombrada previamente en el discurso: *las gafas de Pedro.*
- ◆ *Hiperonimia*: cuando las relaciones anafóricas se establecen a través de hiperónimos¹⁰: *No sabía que ese coche, es tuyo. Opino que es un buen vehículo.*
- ◆ *Anáfora contextual*: supone el uso de una expresión anafórica que conecta con una idea o un hecho que se infiere de una parte del texto anterior. El antecedente puede ser una única entidad, implícita o explícita, puede ser también toda una idea evocada por el texto, o surgir de una presuposición. En cualquier caso, sólo a través del conocimiento y comprensión global del texto podemos interpretar la relación anafórica. Podemos encontrar un buen ejemplo en el siguiente texto: *Él se limpió las gafas, y se las ajustó a la nariz. Su montura, y cristales, estaban húmedas.* La interpretación de la referencia anafórica de *su montura* y *cristales*, exige que el lector conozca las partes que componen unas gafas. Sólo si sabemos que las gafas tienen montura y cristales podemos inferir correctamente que los

¹⁰ La *hiperonimia* es la relación de un término que abarca a otros semánticamente. Por ejemplo *semana* es el hiperónimo de *lunes, martes, miércoles*, etc., de la misma forma que *árbol* es el hiperónimo de *roble, pino*, etc. La relación inversa de la hiperonimia es la *hiponimia* que consiste en la relación de inclusión de un término en otro: por ejemplo *roble* es un hipónimo de *árbol*.

elementos que componen la expresión anafórica tienen como antecedente a *gafas*.

En [80] Hirst señala un tipo de anáfora que se podría incluir dentro de esta anáfora semántica: la *anáfora forzada* (*strained anaphora*). Este tipo de anáfora tiene como característica principal que sus antecedentes deben ser léxicamente similares a las palabras que ya aparecen en el texto. Por ejemplo, en *Pedro se convirtió en guitarrista porque pensaba que éste era un bonito instrumento*, la expresión anafórica *éste* se refiere a *la guitarra*, palabra léxicamente similar a la que aparece en el texto: *guitarrista*.

Para la resolución de la anáfora semántica se está aplicando actualmente la información almacenada en *WordNet*. Por ejemplo, Poesio, Vieira y Teufel [134] muestran cómo aplican WordNet a la resolución de la anáfora. WordNet se enmarca dentro del conjunto de diccionarios y corpus electrónicos que aportan información diversa de naturaleza léxica, sintáctica y semántica.

2.2.2.3 ANÁFORA PRAGMÁTICA

Y por último la *anáfora pragmática* supone una relación entre la expresión anafórica y su antecedente que no depende de factores contextuales, sino que, independientemente del contexto el lector es capaz de identificar el antecedente de la anáfora. Esto es posible gracias al conocimiento pragmático presupuesto por el discurso. De este modo, en el texto: *Esto es como en un libro: La isla del tesoro, La isla del coral, ...* se hace imprescindible poseer el conocimiento pragmático que ayuda a interpretar los títulos *La isla del tesoro* y *La isla del coral* como libros. Este tipo de anáfora necesita para la localización de su antecedente que se construya una representación del mundo.

2.2.3 En función del tipo de referencia

Anteriormente definíamos el proceso de resolución de una anáfora, como aquél que consiste en derreferenciar la expresión anafórica para determinar su antecedente, pudiéndose establecer entre ambos una relación de correferencia o no. En función del tipo de relación que se establezca entre la expresión anafórica y su antecedente Allen [3] nos distingue dos tipos de anáfora: *anáfora profunda* (existe relación de correferencia) y *anáfora superficial* (no existe correferencialidad).

Otros autores llaman a estos dos tipos de anáfora de manera diferente. Por ejemplo, Woods en [172][171] las denomina *anáfora completa* y *anáfora parcial* respectivamente. Y también han sido citadas en el trabajo de Hirst [80], pero llamándolas en este caso *anáfora con identidad de referencia* (*IRA, Identity of Reference Anaphora*) y *anáfora con identidad de sentido* (*ISA, Identity of Sense Anaphora*).

2.2.3.1 ANÁFORA PROFUNDA

El concepto de *anáfora profunda* engloba la referencia completa a un objeto que ha aparecido previamente en el discurso, como podría ser la frase: *[El chico de pelo rojo]*,

*compró un libro*_j. *Él*_i, *lo*_j *compró en la tienda de Juan*. Es decir, la expresión anafórica está referenciando el mismo objeto que el antecedente, o lo que es lo mismo, existe una relación de correferencia entre ambos.

2.2.3.2 ANÁFORA SUPERFICIAL

En [3] Allen define la *anáfora superficial* (*surface anaphora*) como aquella que realiza una referencia parcial a parte de un antecedente. Con ello, la expresión anafórica introduce un nuevo objeto que no aparece explícitamente en el texto y que se relaciona con otro objeto anteriormente nombrado, es decir, no se establece una relación de correferencia entre la expresión anafórica y su antecedente.

Es de resaltar que en muchas ocasiones este tipo de anáfora viene acompañada de adverbios del tipo *también* o *además* que explícitamente avisan de la comparación con algo previamente nombrado en el texto.

La anáfora superficial se puede dar con diversos tipos de expresiones anafóricas:

- ◆ Por ejemplo, en la anáfora pronominal (tipo de anáfora cuyas expresiones anafóricas están formadas por pronombres): *El hombre que da su sueldo*_i, *a su mujer es más sabio que el que lo*_j, *da a su amante*, donde el pronombre *lo* referencia al sueldo del segundo hombre y no al del primero. En [130] Partee destaca una clase de pronombres que se incluyen dentro de este tipo de anáfora superficial. A esta clase de pronombres los denomina *perezosos* (*pronouns of laziness*). En otros trabajos de la literatura sobre el tema también los podemos encontrar como *paycheck pronouns* precisamente por el anterior ejemplo del sueldo¹¹: *The man who gave [his paycheck]_i, to his wife was wiser than the man who gave it_j, to his mistress*.
- ◆ En la *anáfora verbal* también se dan estos tipos de referencias parciales, por ejemplo: *Pedro [besó a su mujer]_i, Juan también [lo hizo]_j*, en la que al resolver la anáfora verbal, queda la interpretación de si *Juan* besó a *su mujer* o bien besó a *la mujer de Pedro*.
Hardt en [76] denomina a este tipo de referencias *identificación descuidada* (*sloppy identity*). Este fenómeno ocurre cuando después de resolver la anáfora verbal o la pronominal queda una referencia intermedia sin resolver generalmente producida por adjetivos posesivos. Por ejemplo, en *Tomás quiere a su gato. Juan también*, después de resolverla nos quedaría: *Juan también [quiere a su gato]_j*, en la cual *el gato* referido podría ser el de *Tomás* o bien el de *Juan*.
- ◆ La *anáfora de tipo "one"* es un tipo especial de anáfora que se da en inglés en el que las expresiones anafóricas están formadas por el pronombre *one* junto con varios modificadores. Determinados casos de este tipo de anáfora realizan este

¹¹ Recordamos que en este trabajo denotaremos con subíndices numéricos (*1, 2, 3, etc.*) la relación de *no* correferencia entre una expresión anafórica y su antecedente, o sea, este tipo de referencias parciales o anáfora superficial. En el caso que sí exista correferencia entre ambos utilizaremos subíndices de tipo carácter (*i, j, k, etc.*), o sea, para las referencias completas o anáfora profunda.

tipo de referencias parciales, por ejemplo la siguiente frase: *Peter bought [a blue pen], yesterday. He has bought [another one], today*, en la que *another one* introduce un nuevo objeto en el discurso.

- ◆ Casos en los que el antecedente aparece cuantificado. Estos casos aparecen en la literatura, por ejemplo en Ersan y Akman [51], como *anáforas que funcionan como variables sin ligar (bound anaphora)*, como por ejemplo las siguientes frases: *[Ningún niño], admitirá que él, es somnoliento. [Ningún hombre], se culparía a [sí mismo]*. Según se desprende de estos ejemplos es evidente que no existe correferencia entre la expresión anafórica y su antecedente, sino que únicamente se ha de ligar la variable correspondiente al sintagma nominal cuantificado (*ningún niño*) con la del pronombre (*él*).

Dentro de este tipo de casos en los que el antecedente aparece cuantificado, se abre un nuevo campo de estudio tratado con asiduidad en la literatura actual, por ejemplo en Shalom y Nissim [149] y Chierchia [31], es el conocido como *donkey pronouns* o *donkey sentences*. El nombre de estos pronombres viene dado precisamente por el ejemplo del cual se parte: *Every man who owns a donkey beats it. Most farmers that have donkey beat it* o *If a man owns a donkey, he beats it*. O también con los siguientes ejemplos equivalentes: *Every person who has a credit card, will pay his bill with it* o *If a painter lives in a village, it is usually pretty*.

Para concluir este apartado, y siguiendo la reflexión hecha por Popowich en [138] para los pronombres reflexivos, podemos decir que tan importante es determinar el antecedente de una expresión anafórica, como el determinar la relación existente entre ambos. En el mencionado trabajo de Popowich, se muestran ejemplos en los que también para los pronombres reflexivos se pueden dar los dos tipos de anáfora: tanto profunda como superficial. Por ejemplo, en la frase *Sólo Juan se compadece a sí mismo*, al resolver la anáfora tendríamos que *Sólo Juan compadece a Juan*. La frase original se puede interpretar como que *Juan* es la única persona que se compadece a sí misma, mientras que en la segunda sería que *Juan* es la única persona que compadece a *Juan*. La diferencia entre ambas interpretaciones la resuelve Popowich por medio de los índices que asigna a cada sintagma nominal al resolver la anáfora: en el caso que asigne el mismo índice se tratará de anáfora profunda, y en caso contrario será anáfora superficial.

En nuestro trabajo también afrontaremos estos tipos de referencias parciales producidas por la anáfora superficial. Es decir, detectaremos determinadas situaciones en las que no se establece una relación de correferencia entre la expresión anafórica y su antecedente dentro de la anáfora pronominal y de tipo adjetivo, tipos de anáfora que trataremos con mayor detalle en la próxima subsección.

2.2.4 Según el tipo de expresión anafórica

Las expresiones anafóricas más habituales en lenguaje natural son los pronombres y sintagmas nominales definidos. Sin embargo, podemos encontrar otras expresiones. Por ejemplo, para el idioma inglés, tal y como resalta Covington en [42], podemos encontrar los llamados *verbos anafóricos* (los auxiliares como *do* o *will*) que se referirán a acciones o estados. Y también podemos encontrar los adverbios realizando referencias a tiempos y lugares. En esta subsección vamos a estudiar los siguientes tipos de expresiones anafóricas:

pronombres, sintagmas nominales definidos, anáfora de tipo "one", anáfora tipo adjetivo, anáfora superficial numérica, anáfora verbal, adverbios y complementos circunstanciales.

2.2.4.1 ANÁFORA PRONOMINAL

Las referencias producidas por los pronombres han sido tratadas profundamente en la literatura actual sobre la anáfora, pudiéndose producir tanto como anáfora intraoracional como anáfora discursiva. E igualmente pueden realizar tanto referencias parciales como completas, tal y como ya hemos visto en la subsección anterior. También vimos cómo estas referencias tienen una alta accesibilidad, es decir, una distancia pequeña con su antecedente ya que presentan una carga léxica muy pequeña.

Dentro de las referencias creadas por los pronombres destacan las producidas por los pronombres reflexivos, ya que como veremos posteriormente determinan cierta información especial que permite la identificación de su antecedente: han de tener su antecedente en la misma frase ocupando principalmente la función de sujeto: *Juan, se, peina*. Los reflexivos también se diferencian del resto de pronombres por la posibilidad de producir una lectura recíproca de la frase, tal y como ocurre en *[María y Luis], se, saludaron*.

Además existe la posibilidad de que un determinado pronombre no tenga antecedente. Estos pronombres los llamó Hirst en [80] *pronombres no referenciales*, a los que no considera como anáfora. Este es un fenómeno habitual en inglés principalmente con el pronombre *it* (se coloca en los casos de oraciones impersonales en las que no hay sujeto, ya que en inglés es obligatoria su presencia). Podemos encontrar varios ejemplos en las siguientes frases: *It is fortunate that Nadia will never read this book. It is half past two. It is raining*. Los casos equivalentes en castellano corresponderían a las oraciones impersonales en las que el sujeto semántico es desconocido, por lo que aunque esté omitido el sujeto pronominal, en realidad no constituyen una anáfora (no tienen antecedente): *Llaman a la puerta. Dicen que subirá la gasolina. Se está bien aquí*.

Y para concluir, resaltar que un pronombre puede hacer referencia a toda una oración o situación, y no solamente a uno de los sintagmas nominales anteriores: *[El Presidente fue disparado mientras iba en un coche], Esto, produjo pánico en Wall Street*.

2.2.4.2 DESCRIPCIONES DEFINIDAS

Las *descripciones definidas* son expresiones anafóricas formadas por sintagmas nominales. En [134] Poesio, Vieira y Teufel hacen una clasificación de las descripciones definidas como resultado de un estudio sobre un conjunto de artículos del Wall Street Journal. En esta clasificación se distinguen los siguientes tipos:

- ◆ *Descripciones anafóricas*: descripciones definidas que correferen con sus antecedentes y comparten el mismo nombre principal. En el corpus de estudio contaron un 30% de descripciones definidas de este tipo.
- ◆ *Referencias puente*: aquellos usos de descripciones definidas basadas en el discurso previo que requieren algún razonamiento en la identificación de su antecedente textual (más que el simple emparejamiento de nombres idénticos). Se contaron un 20% de descripciones definidas de este tipo.

- ◆ *Descripciones definidas basadas en el conocimiento común*: aquellas que usan conocimiento presupuesto en el discurso, como podría ser el caso del sintagma nominal *el gobierno*. Se contaron un 47% de descripciones definidas de este tipo.
- ◆ *Expresiones idiomáticas o casos dudosos*: aquellas que no tienen un uso anafórico, como el sintagma nominal *el pelo* en la siguiente frase: *No me tomes el pelo*. Se contaron un 3%.

En [3] Allen define las *descripciones definidas* como una referencia a un objeto que está normalmente determinado de forma única en el contexto, y realiza una importante distinción dependiendo de si son usadas de modo *existencial* o *referencial*:

- ◆ El uso *existencial* de una descripción definida según Allen correspondería a aquel que afirma la existencia de un único objeto que satisface la descripción y que previamente no ha sido introducido en el discurso. Por ejemplo, en *La luna ha salido esta tarde a las 7:05* se está hablando de *la luna* que gira alrededor de *la tierra*, y no de otras lunas a las que nos podríamos referir en un contexto adecuado.

En [42] Covington sugiere que en la fórmula lógica se le debe añadir esta lectura existencial. Por ejemplo, para la frase *El Rey es calvo* propone la fórmula lógica¹²: $(\exists ! X) (rey(X) \wedge calvo(X))$, es decir, existe un único rey.

- ◆ El uso *referencial* utiliza la descripción para referirse a un objeto previamente conocido. Es decir, con la lectura referencial de una descripción definida el oyente debe identificar el antecedente, y no simplemente aceptar que el referente existe (a diferencia de la lectura existencial). Por ejemplo, en la frase *La vaca está enferma* el oyente debe identificar qué *vaca* es de la que se está hablando. Por ello, en esta lectura se debe identificar el antecedente o de otro modo se considerará que la frase está mal formada. Sin embargo la identificación del antecedente no se ha de entender como una detección absoluta como ocurriría en el siguiente texto: *Ana se compró un coche, ayer. Ella pagó mucho por ese coche,* en el que no es necesario saber exactamente cuál es el coche al que se está referenciando, se considera suficiente el saber que se trata del *coche que compró Ana*.

En el siguiente ejemplo se destaca la importancia de estas descripciones definidas referenciales: *Un perro_i ladró y un gato_j maulló. Después el perro_i persiguió al gato_j.* En la segunda frase se está hablando del mismo perro y gato que en la primera, mientras que si esta segunda frase se hubiese formulado de la siguiente manera: *Después un perro_k persiguió a un gato_m,* esto sugeriría que no se trata del mismo perro y gato.

Hirst en [80] distingue un caso particular dentro de las descripciones definidas con las referencias producidas por los epítetos: *Juan_i perdió el dinero, [el pobre chico]_i, está hundido.*

En función del tipo de expresión anafórica, podremos obtener información importante para su resolución, como podría ser la posible posición de su antecedente. Por ejemplo, sabremos que los pronombres tendrán su antecedente preferentemente en la misma oración

¹² Covington utiliza el cuantificador $\exists ! X$ para expresar: “existe un único X ”.

o no muchas oraciones atrás¹³, y dentro de los pronombres, los reflexivos lo han de tener en la misma oración. Sin embargo en las descripciones definidas puede existir una mayor distancia con su antecedente. Esto es así ya que éstas tienen mayor carga léxica que los pronombres, por lo que la referencia se puede encontrar aunque haya una mayor distancia entre antecedente y expresión anafórica.

2.2.4.3 ANÁFORA DE TIPO "ONE", ANÁFORA TIPO ADJETIVO Y ANÁFORA SUPERFICIAL NUMÉRICA

Un tipo de anáfora tratado ampliamente dentro de la literatura actual de la anáfora es el conocido como *anáfora de tipo "one" (one-anaphora)*. Su nombre viene de los casos producidos dentro del inglés en los que la expresión anafórica se forma mediante un sintagma nominal con la siguiente estructura: el pronombre *one* acompañado de diversos modificadores. Un ejemplo podría ser el siguiente: *Wendy didn't give either boy [a green tie-dyed T-shirt]_i, but she gave Sue [a red one]_i*, en la que el sintagma nominal *a red one*, introduciría la siguiente entidad *a red tie-dyed T-shirt*.

Un problema que aparece con este tipo de anáfora, es la posibilidad que aparezca el pronombre *one* sólo (sin ningún modificador) y con un significado indefinido, es decir, sin referirse a ningún antecedente, ya que no se trata de anáfora. Podemos encontrar los siguientes ejemplos: *Smoking gives one cancer. My boss makes one work hard. John makes one sick*. Y en castellano podemos encontrar también ejemplos equivalentes como en *Uno se siente incómodo*.

La frase del primer ejemplo en castellano sería: *Wendy no dió a ningún niño [una camisa verde de manga corta]_i, pero a Sue le dió [una roja]_i*. A este tipo de anáfora la denominamos de *tipo adjetivo* precisamente porque la detectamos por la presencia de un sintagma nominal en el que ha sido elidido el núcleo nominal, función que el adjetivo realiza temporalmente (hasta que se resuelva).

Dentro de la anáfora tipo adjetivo destaca la formada por las expresiones anafóricas *el mismo, la misma* y sus variantes en plural. Este tipo de expresiones anafóricas se refieren a sintagmas nominales muy recientes, normalmente justo el sintagma nominal que le precede: *La gente que use tetera, para el café, debe limpiar la misma, después de cada uso*.

Un problema que nos plantea la anáfora de tipo "one" y la anáfora tipo adjetivo, es que permiten tanto referencias parciales como completas tal y como ya vimos en la sección 2.2.3. Por ejemplo, en *Compré [una pera verde]_i y [otra roja]_j. Yo prefiero [la verde]_i*, en la anáfora de tipo adjetivo producida por *otra roja* realiza una referencia parcial (introduce un nuevo objeto en el discurso) mientras que en *la verde* realiza una referencia completa (es

¹³ Se han realizado diversos estudios sobre el número máximo de oraciones anteriores a considerar en el proceso de búsqueda del antecedente de una anáfora pronominal. En nuestro trabajo hemos considerado dos oraciones anteriores tras un estudio previo del corpus sobre el que aplicamos nuestro sistema. Rico en [143] estudió el corpus *Susanne* (en lengua inglesa) observando que incluso se encontraban antecedentes tres oraciones atrás. En el apartado 3.1.5.2.1 estudiaremos con mayor profundidad este aspecto.

correferente con *una pera verde*). Un método para distinguir entre ambos tipos de referencia podría ser la siguiente regla simple propuesta por Hirst en [80]:

Se realizará una referencia parcial cuando venga acompañado de modificadores del sintagma nominal que lo diferencien del antecedente.

Es decir, en la siguiente frase: *Peter was watching [the car I bought], and John's car. He liked more [the one I bought],*, tendríamos una referencia parcial ya que aunque la expresión anafórica viene acompañada de modificadores, en realidad tiene exactamente los mismos modificadores que el antecedente al que se refiere (tan sólo se ha sustituido el nombre, *car*, por el pronombre *one*).

Un tipo de anáfora similar a la anterior de tipo adjetivo es la denominada *anáfora superficial numérica (surface count anaphora)*. En este tipo de anáfora las expresiones anafóricas presentan la siguiente estructura: *determinante + adjetivo + modificadores*. El componente adjetivo se especializa (para distinguirla de la anterior anáfora tipo adjetivo) en los ordinales, numerales y todas las expresiones anafóricas que signifiquen cierto grado de orden: *el primero, el último, el tercero, el tres*, etc., información que será especialmente útil para determinar su antecedente. Así podemos encontrar el siguiente ejemplo: *Pedro miró [al perro, y al gato], pero finalmente eligió el primero*. Para resolver este tipo de anáfora se requiere que además de tener el antecedente en la memoria del oyente se retenga la estructura superficial de la oración, es decir, el orden en el que aparecen los antecedentes dentro de la misma. Evidentemente, en caso que aparezcan demasiados antecedentes dentro de la oración el resultado sería inaceptable, tal y como ocurriría en: *Esta mañana compré un libro, un boli, un lápiz, un sacapuntas, un diccionario y una libreta. El cuarto que compré se rompió rápidamente*. Este tipo de anáfora en inglés tiene una estructura similar a la anterior anáfora de tipo "one" ya que también incluyen como núcleo el pronombre *one*, por ejemplo: *the second one* o *the last one*, sin embargo es conveniente distinguirla ya que realmente introduce un nuevo tipo de información (la relación de orden) indispensable para su correcta resolución.

2.2.4.4 ANÁFORA VERBAL

La *anáfora verbal* se construye mediante la llamada eliminación del verbo o gapping, y mediante la utilización de un pro-verbo (por ejemplo en inglés los auxiliares *do* o *have*). Esta anáfora puede referirse bien a un verbo bien a una frase verbal, como ocurre en la frase: *Peter danced with Jane at the party. John did it too*. También podemos encontrar la anáfora verbal en castellano habitualmente formada por un pronombre y un auxiliar, como por ejemplo en *Pedro [jugó muy bien al tenis ayer], pero Juan [lo hizo], muy mal*.

Este tipo de anáfora verbal tiene un cierto parecido con la ya citada *elipsis verbal*. La diferencia entre ambas es que en la elipsis se produce una sustitución por cero, mientras que en la anáfora existe la expresión anafórica con la que se realiza la referencia. El ejemplo anterior de anáfora verbal se podría conseguir de modo equivalente como la siguiente elipsis verbal: *Pedro jugó muy bien al tenis ayer, pero Juan \emptyset muy mal*. Otra diferencia entre la anáfora y la elipsis verbal es referenciada por Franchini en [64]: la anáfora verbal no es posible cuando el verbo anteriormente mencionado en el enunciado ya lleva un complemento directo: *Juan bebió vino y María [~~lo hizo~~] \emptyset agua mineral*.

2.2.4.5 ADVERBIOS Y COMPLEMENTOS CIRCUNSTANCIALES

Los adverbios y complementos circunstanciales también pueden formar expresiones anafóricas tal y como ocurre en el siguiente ejemplo: *La iglesia estaba [detrás de la librería]; Marga fue ahí, después del almuerzo.*

Hirst en [80] denomina a este tipo de expresiones anafóricas como *referencias temporales* o *locales*, indicando que sus antecedentes consisten siempre en la localización temporal (o local) más reciente en el texto, tal y como lo muestra en el siguiente texto: *El despertador suena a [las 6 de la mañana]. [Las siguientes dos horas], se pasan en tranquila meditación, y es entonces, cuando ...*, donde la expresión *las siguientes dos horas* toma su antecedente de la anterior expresión de tiempo: *a las 6 de la mañana*. Y la última anáfora, *entonces*, tomará su antecedente a partir de la anterior, es decir, *las siguientes dos horas*.

2.3 Objetivos de este trabajo

A modo de conclusión de este capítulo dedicado a exponer el problema de la anáfora, resumimos los objetivos que nos planteamos en el tratamiento de un tipo especial de anáfora:

- ◆ Nuestro objetivo es el encontrar la entidad (a la cual llamaremos **antecedente**) a la que se hace referencia (a la referencia abreviada la llamaremos **expresión anafórica**).
- ◆ Dentro de los diferentes tipos de anáfora que hemos descrito en este capítulo nos centraremos en:
 - ◇ Aquella que sucede en el **contexto lingüístico**, es decir, aquella en la que los antecedentes aparecen explícitamente en el texto.
 - ◇ Buscando como solución la entidad con la que se establece una relación de correferencia en su caso (**anáfora profunda**), o la nueva entidad que se introduce en el discurso (**anáfora superficial**).
 - ◇ Esta entidad la buscaremos tanto en el marco de la oración en la que aparece la expresión anafórica, como en el marco del discurso (**anáfora discursiva**).
 - ◇ Nos centraremos en la **anáfora morfosintáctica**, y más concretamente en la **anáfora nominal**, o sea, trataremos las expresiones anafóricas que tengan como antecedente un sintagma nominal.
 - ◇ Y entre las expresiones anafóricas existentes en el tipo de anáfora que acabamos de delimitar, nos interesarán la **anáfora pronominal** y la **de tipo adjetivo**, es decir, expresiones anafóricas de tipo pronombre y sintagmas nominales cuyo núcleo sea un adjetivo.
- ◆ Para finalizar, trataremos este tipo concreto de anáfora desde el punto de vista de la Lingüística Computacional, es decir, tratándola mediante la información lingüística mediante métodos informáticos a fin de facilitar el tratamiento automático del lenguaje natural.

En los próximos capítulos nos dedicaremos a desarrollar los objetivos que aquí planteamos. En el capítulo siguiente describiremos las principales estrategias que se están siguiendo en la actualidad para resolver el tipo de anáfora que aquí nos ocupa. Y después desglosaremos la solución que proponemos en este trabajo.

Universitat d'Alacant
Universidad de Alicante

3. Estrategias para la resolución de la anáfora

En este capítulo describiremos las diferentes estrategias que se están aplicando en la actualidad para el tratamiento y resolución de la anáfora. Estas estrategias se dividen en dos tipos: *estrategias integradas* y *estrategias alternativas*. Las primeras se basan en el conocimiento, es decir, manejan una serie de fuentes de información que se consideran necesarias para el correcto tratamiento de la anáfora, mientras que las segundas utilizan información estadística y principios de razonamiento con incertidumbre.

Dentro de las estrategias integradas se considera que el proceso de determinar el conjunto de posibles antecedentes de una determinada anáfora está gobernado por diversas fuentes de información: morfológica, léxica, sintáctica, semántica y pragmática. Un algoritmo genérico de este tipo de estrategia para tratamiento de la anáfora podría ser el siguiente:

En primer lugar identificar las expresiones anafóricas. A continuación buscar sus antecedentes utilizando diferentes fuentes de información.

En la literatura actual sobre el tema aparecen diversos métodos de coordinación de estas fuentes de información, sin embargo, hay un punto común entre todas ellas y es la distinción entre *restricciones* y *preferencias*. Las *restricciones* tienden a ser absolutas, y como tales eliminan antecedentes factibles de una determinada anáfora. Por otro lado, las *preferencias* tienden a ser relativas por lo que requieren el uso de información adicional y tienen carácter consultivo. Las preferencias se suelen aplicar después de las restricciones, y sólo en el caso en que quede más de un candidato posible. Éstas tienen como objetivo el seleccionar uno de estos candidatos posibles.

Una misma fuente de información puede ser tratada como restricción o preferencia en función del algoritmo o sistema en que se aplique para la resolución de la anáfora, o en función del tipo de expresión anafórica. Por ello, en la próxima sección trataremos en profundidad cada una de estas fuentes de información independientemente unas de otras, tanto si se tratan como restricciones o como preferencias, dejando para la siguiente sección el estudio de los algoritmos que las utilizan, momento en el que describiremos diversos métodos de coordinación de estas fuentes de información.

3.1 Fuentes de información

Según hemos comentado en la introducción de este capítulo se necesitan diferentes fuentes de información para permitir un correcto tratamiento de la anáfora. En función de la información que se utilice también clasificamos los algoritmos que se están aplicando en la actualidad: *integrados* y *alternativos*. En esta sección expondremos cada uno de estos tipos de información en diferentes subsecciones: información morfológica, léxica, sintáctica, semántica, pragmática, sobre la expresión anafórica e información estadística y sobre tratamiento de corpus.

3.1.1 Información morfológica

La información morfológica necesaria para el tratamiento de la anáfora se corresponde con la concordancia en número, género y persona entre la expresión anafórica y su antecedente:

- ◆ En *Luis compró varios libros_i, Los_i compró en la tienda de Marga*, el pronombre *los* necesariamente ha de referirse a los *libros* y nunca a *Luis* al no concordar en número.
- ◆ Por ejemplo, en el siguiente texto: *Pedro_i y Ana_j son buenos amigos. Sin embargo ella_i comenzó a evitarlo_i*, la concordancia en género constituiría suficiente información para determinar que el antecedente del pronombre *ella* ha de ser necesariamente *Ana* y no *Pedro*.
- ◆ En: *Juan y yo fuimos al cine* el pronombre *yo* al estar en primera persona se le asociará el antecedente del narrador de la historia, y nunca se le podría relacionar con *Juan* al estar en tercera persona.

Ésta es una información empleada tradicionalmente para la resolución de la anáfora y se ha considerado habitualmente como una restricción (y no como una preferencia) a la hora de seleccionar posibles antecedentes, tan sólo citar los algoritmos propuestos por Lappin y McCord en [97][96]. Sin embargo no ha pasado desapercibido para los investigadores que estas restricciones no son absolutas, es decir, no se cumplen siempre. Tal y como describen Rico en [143] y Hirst en [80] se destacan principalmente los siguientes casos:

1. Una expresión anafórica en plural puede tener un conjunto de antecedentes en singular, como por ejemplo en el siguiente texto: *[Juan y Pedro]_i fueron a Paris. Ellos_i estuvieron allí durante cinco días*, en la que la anáfora formada por el pronombre en plural *ellos* tiene como antecedente un grupo de antecedentes singulares: *Juan y Pedro*. Otro ejemplo más complejo que el anterior, sería cuando estos antecedentes singulares están separados, o sea, no aparecen coordinados en un sintagma nominal, como en *Juan_i le dijo a Pedro_j que ellos_{i,j} deberían llamar ahora*. Este último ejemplo determina un caso especial de anáfora denominado *antecedentes divididos (split antecedents)* en el que se incrementa la dificultad de resolución al tener que decidir qué antecedentes se incluyen en este conjunto y cuáles no, tal y como muestran Williams, Harvey y Preston en [168] sobre el siguiente texto:

*A move to stop Mr Gaitskell from nominating any more labour life peers is to be made at a meeting of labour MPs tomorrow. Mr Michael Foot has put down a resolution on the subject and he is to be backed by Mr Will Griffiths MP for Manchester Exchange. Though **they** may gather some left-wing support ...*

en el que el pronombre *they* tiene como antecedentes a *Mr Michael Foot* y *Mr Will Griffiths*, pero no hay nada que impidiese también incluir a *Mr Gaitskell*, por lo que en estos casos se necesita información semántica y del dominio del texto para determinar las relaciones aquí descritas.

2. El caso inverso al anterior, o sea, una expresión anafórica en singular puede tener un antecedente en plural, como en el siguiente ejemplo: *En el zoo, un mono correteaba entre dos elefantes, Uno, empujó al mono*, en el que la anáfora tiene como antecedente a uno de los elementos del conjunto referenciado por el sintagma nominal *dos elefantes*. Otro ejemplo sucedería cuando la expresión anafórica también está cuantificada por el cuantificador universal, en cuyo caso quedaría siempre con número singular, como en *Los niños, compraron un juguete. Cada niño, eligió el que quiso*.
3. Otro caso sucedería cuando se emplean nombres colectivos. Estos nombres pueden aceptar una referencia en singular o en plural, como sucedería en el texto: *The tribe, had been walking for two hours. (They, were / it, was) tired*, donde *they* designaría a los miembros del conjunto, mientras que *it* designaría al conjunto en su totalidad. Un ejemplo equivalente para el castellano ocurriría en: *El clero, es conservador. (A ellos, / a éste) no les gusta el cambio*, donde de nuevo, el pronombre plural (*ellos*) referenciaría a los miembros del conjunto, mientras que el pronombre singular (*éste*) referenciaría al conjunto en su totalidad.
4. Una expresión anafórica con género femenino puede tener un antecedente con género masculino, y el caso inverso. Un ejemplo sucedería en el siguiente texto: *Who is this Bresson,? Is she, a woman?* en el que el pronombre *she* se refiere a una persona de nombre *Bresson* que probablemente estará marcada como mujer en la consciencia del oyente. Estos casos vienen determinados habitualmente por nombres que se pueden aplicar tanto a entidades del mundo con género masculino como femenino, y se selecciona uno de estos géneros precisamente por los modificadores que le acompañan.

Para tratar estas excepciones de la concordancia en número, género y persona los diferentes algoritmos optan principalmente por dos soluciones: o bien tratan por separado cada excepción almacenándolas en el diccionario o en el propio sistema de tratamiento de la anáfora; o bien no definen este tipo de información como restricción absoluta sino como preferencia relativa.

3.1.2 Información léxica

Dentro de la información léxica se podría incluir aquella relativa al comportamiento de ciertas palabras o grupos de palabras en determinadas situaciones. Esta información se suele almacenar dentro del léxico o diccionario del sistema.

En [69] Grober y Beardsley proponen un ejemplo de la utilidad de esta información, a la que denominan *valencia implícita causal del verbo* (*implicit verb causality* o *causal valence*), indicando que la valencia causal de un verbo puede afectar a los antecedentes seleccionados de expresiones anafóricas cercanas. De este modo consideran que en general las oraciones con la siguiente estructura SN_1 VERBO SN_2 *porque* {*él / ella*}... tienen una tendencia natural por la que el pronombre *él* o *ella* se refiera a SN_1 para determinados verbos (VERBO), a SN_2 para otros, y neutrales (igual SN_1 que SN_2) para otros. Por ejemplo, en *Ana_i ganó a Elena porque ella_i es una jugadora muy hábil*, el pronombre se referirá a SN_1 , mientras que en *Pedro le pegó a Luis_i porque él_i se había portado mal*, el pronombre se referirá a SN_2 .

Esta información se suele tratar como preferencia, ya que es fácil encontrar ejemplos en los que no se cumpla, por ejemplo: *Ana ganó a Elena_i porque ella_i es una jugadora muy torpe*, y en *Pedro_i le pegó a Luis porque él_i es más fuerte*.

En [118] Mitkov y Stys utilizan información de cierto grupo de verbos ingleses para los que se prefiere como antecedente el primer sintagma nominal que los siga. Estos verbos son *discuss, present, illustrate, identify, summarise, examine, describe, define, show, check, develop, review, report, outline, consider, investigate, explore, assess, analyse, synthesise, study, survey, deal* o *cover*. Un ejemplo podría ser la siguiente frase: *This table shows a minimal configuration_i; it_i does not leave much room for* También utilizan información procedente de los sustantivos, como en el caso de situaciones en las que el núcleo del sintagma nominal que precede al verbo es *chapter, section* o *table*, entonces se escogerá el antecedente que siga al verbo, o del mismo modo, en caso que el sintagma nominal está en la cabecera de la sección y forma parte de la oración donde está la expresión anafórica, entonces se escogería éste como antecedente.

Otra información léxica que se ha utilizado habitualmente en el tratamiento de la anáfora (por ejemplo también la utilizan Mitkov y Stys en [118]) es la *reiteración* del antecedente. Según esta información se preferirá el antecedente que aparezca repetidas veces en el texto, con mayor grado de preferencia cuantas más veces aparezca repetido.

3.1.3 Información sintáctica

La información sintáctica ha sido utilizada ampliamente en la resolución de la anáfora. Esta información se obtiene a partir del análisis sintáctico del texto. Toda la información necesaria para realizar este análisis sintáctico se almacena en lo que se denomina *gramática*.

Una vez obtenida toda la información sintáctica del texto, se pueden formular diversas reglas que permiten determinar o eliminar antecedentes de una cierta expresión anafórica. Por ejemplo, la siguiente regla formulada por Hirst en [80]:

Un pronombre que aparezca en la oración principal nunca puede referirse a un sintagma nominal que aparezca en una oración subordinada a ésta.

Por ejemplo, en la siguiente frase: *Él llegó tarde al trabajo porque Juan no le despertó*, el pronombre *él* nunca podrá referirse a *Juan*. Para el caso inverso, cuando el

pronombre aparece dentro de una oración subordinada, no siempre se cumple esta regla. Por ejemplo, en *Juan llegó tarde al trabajo porque él se durmió*, se podría aceptar que *Juan* y *él* fuesen correferentes, aunque quizás un modo más correcto de uso obligaría a omitir el sujeto pronominal: *Juan, llegó tarde al trabajo porque \emptyset , se durmió*.

Este tipo de información sintáctica se utiliza de modo especial en la resolución de la anáfora pronominal bajo la denominación de restricciones *c-dominio* o *c-command* (*constituent-command*). Por ejemplo, Reinhart define en [141] la relación *c-dominio* entre dos constituyentes o nodos de la estructura sintáctica que representa a la oración:

Para dos nodos A y B de una estructura sintáctica, A c-domina a B si y sólo si el primer nodo que domine¹⁴ a A, también domina a B.

Y también define la relación por la que en un árbol o estructura sintáctica el nodo *A* domina al nodo *B* si y sólo si existe un camino en ese árbol desde *A* hasta *B* que los une en sentido descendente. En el mismo trabajo, Reinhart define las siguientes restricciones *c-dominio*:

1. Un sintagma nominal completo se debe interpretar como no correferencial con cualquier sintagma nominal completo al que c-domine.
2. Un pronombre debe interpretarse como no correferencial con cualquier sintagma nominal al que c-domine.
3. Un pronombre reflexivo debe interpretarse como correferencial únicamente con un sintagma nominal que lo c-domine dentro de un espacio sintáctico definido: su *categoría mínima de gobierno* (*minimal governing category*).

Se definirá como *categoría mínima de gobierno* de un nodo *A*, al menor nodo que determine la oración o un sintagma nominal que c-domine a *A* y a su nodo principal (el del constituyente al que pertenezca *A*, por ejemplo el nombre de un sintagma nominal, o el verbo de un sintagma verbal).

4. Un pronombre no reflexivo debe interpretarse como no correferencial con cualquier sintagma nominal que lo c-domine y que pertenezca a su categoría mínima de gobierno.

Podemos encontrar un ejemplo de la aplicación de la restricción 2 en la Figura 4. Aquí se puede observar cómo el pronombre *SN₃* no c-domina a *SN₁*, ya que el primer nodo que domina a *SN₃* es *SP*, el cual no domina a *SN₁*, por lo que esta restricción no impide que pueda considerarse como un antecedente posible. Sin embargo en la Figura 5, el primer nodo que domina a *SN₁* es *O*, el cual sí que domina a *SN₃*, por lo que han de considerarse no correferentes. Igualmente, otro hipotético caso de catáfora sería *Él vio a Pedro*, en el cual el pronombre nunca podría correferir con *Pedro* ya que *él* lo c-domina.

¹⁴ El primer nodo que domina a *A* es el propio padre de *A* en el árbol que representa la estructura sintáctica.

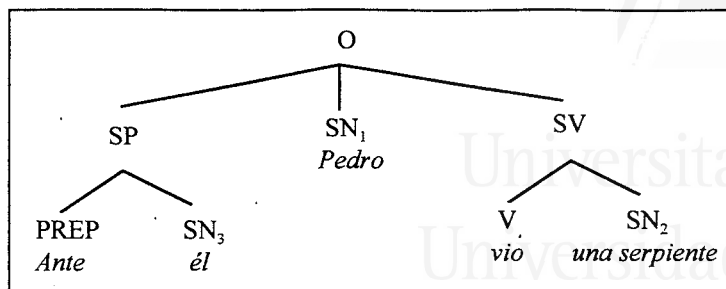


Figura 4. Ejemplo de aplicación de la restricción c-dominio número 2 en: “Ante él, Pedro vio una serpiente”.

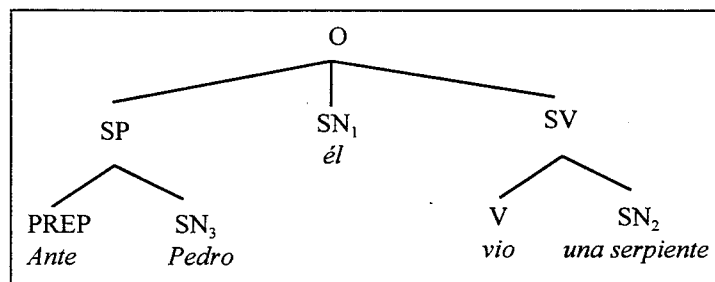


Figura 5. Ejemplo de aplicación de la restricción c-dominio número 2 en: “Ante Pedro, él vio una serpiente”.

Un ejemplo de la restricción 1 lo encontramos en la Figura 6, en la que podemos observar que tanto SN_3 como SN_1 son sintagmas nominales completos, y nunca podrán ser correferenciales ya que SN_1 c-domina a SN_3 .

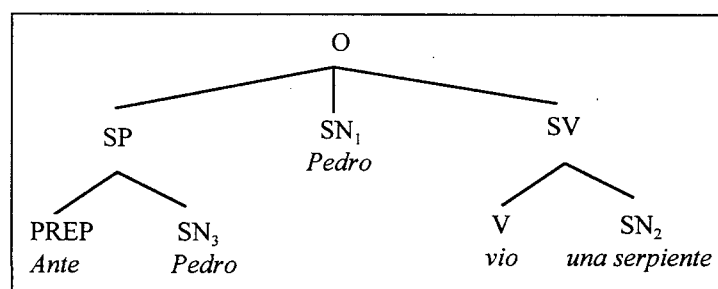


Figura 6. Ejemplo de aplicación de la restricción c-dominio número 1 en: “Ante Pedro, Pedro vio una serpiente”.

La restricción 3 se ilustra mediante el ejemplo de la Figura 7: *Luisa, se, aburre*. Esta restricción nos indica que un pronombre reflexivo debe ser interpretado como correferencial únicamente con un sintagma nominal que se encuentre en su categoría mínima de gobierno y que c-domine al pronombre. En esta figura SN_1 (*Luisa*) c-domina a SN_2 (*se*) y a su vez pertenece a su categoría mínima de gobierno determinada por el nodo O , por lo que correferen.

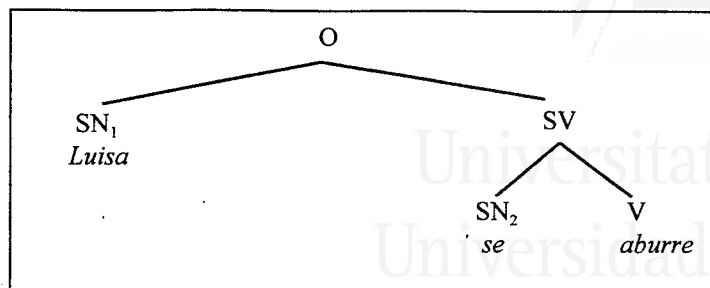


Figura 7. Aplicación de la restricción c-dominio número 3 en: "Luisa se aburre".

En la Figura 8 se muestra una frase similar a la anterior aunque se sustituye el pronombre reflexivo (*se*) por un pronombre personal (*le*): *Luisa le aburre*. En este caso aunque no se incumple la restricción 2 (ya que *le* no domina a *Luisa*), podremos rechazar el antecedente SN_1 gracias a la restricción 4 que nos exige que el antecedente no pertenezca a la categoría mínima de gobierno del pronombre (en este caso O). Encontramos otro ejemplo similar de la aplicación de esta restricción 4 en la frase *María llegó con ella*, en la que el pronombre no podría correferir con *María*, ya que aunque no c-domine a *María* (restricción 2), sí que se cumple la restricción 4: *María* pertenece a la categoría mínima de gobierno del pronombre. Igualmente se aplica esta restricción para evitar la correferencia en *el hijo, de él*, ya que el pronombre y el nombre *hijo* están dentro de la misma categoría mínima de gobierno que será el propio sintagma nominal que los incluye.

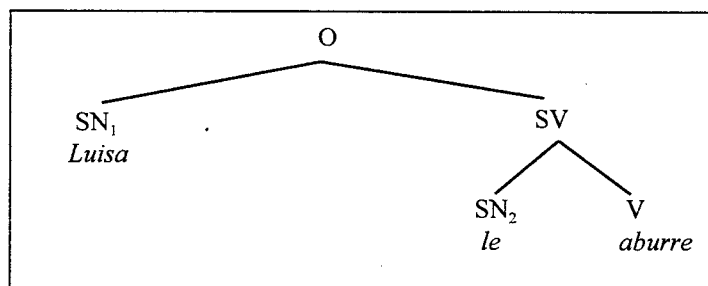


Figura 8. Aplicación de la restricción c-dominio número 4 en: "Luisa le aburre".

Finalmente, en el ejemplo de la Figura 9 se puede observar un caso en el que no se cumplen la restricción 2 ni la 4 (la categoría mínima de gobierno del pronombre *él* estará marcada por O_3), por lo que el pronombre puede correferir con *Juan*.

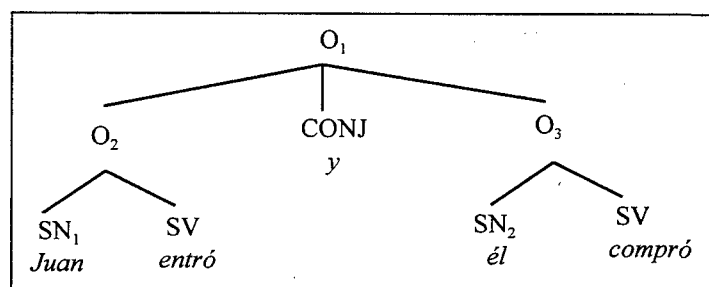


Figura 9. Ejemplo de correferencia en: "Juan entró y él compró".

Estas restricciones c-dominio se utilizan habitualmente para eliminar antecedentes que se saben que no van a correferir con el pronombre en cuestión. Por ello, se la denomina *disjoint reference filters* (filtros que determinan la no correferencia) en el algoritmo propuesto en [93] por Kennedy y Boguraev, o en el que a continuación detallamos propuesto por Lappin y McCord en [97][96]. Las restricciones propuestas en este último trabajo están basadas en las siguientes definiciones que suponen una adaptación de la definición de c-dominio para las gramáticas de huecos (SG, Slot Grammars):

- ◇ Un sintagma P está en el **dominio del argumento** (*argument domain*) de otro sintagma N , si P y N son ambos argumentos de la misma palabra núcleo H . Por ejemplo: *Pedro_N eligió_H éste_P*.
- ◇ P está en el **dominio adjunto** (*adjunct domain*) de N , si N es un argumento de una palabra núcleo H , P es el objeto de una preposición $PREP$, y $PREP$ es un adjunto de H . Por ejemplo: *María_N llegó_H con_{PREP} ella_P*.
- ◇ P está en el **dominio de sintagma nominal** (*noun phrase domain*), si D es el determinante de un nombre N y, o bien P es un argumento de N (*El_D perro_N suyo_P*) o bien P es el objeto de una preposición $PREP$ y $PREP$ es un adjunto de N (*El_D hijo_N de_{PREP} él_P*).

Según Lappin y McCord en [97][96] un pronombre P no es correferencial con un sintagma nominal N si se cumple alguna de las siguientes condiciones:

- ◆ P y N tienen características de concordancia incompatibles. Por ejemplo, en *La mujer_i dijo que él_j es gracioso*.
- ◆ P está en el dominio del argumento de N tal y como sucedería en la siguiente frase: *María_i eligió esta_j*, o en la siguiente: *Juan_i parece querer verle_j*.
- ◆ P está en el dominio adjunto de N . Ejemplos: *María_i llegó con ella_j*, *Who_i did John say wants to sit near him_j?*.
- ◆ P está en el argumento de una palabra núcleo H , N no es un pronombre, y N está contenido en H . Ejemplos: *Éste es el hombre_i de quien él_j escribió*, *Éste es el hombre_i que él_j vio*, *A él_i le gusta la mujer de Juan_j*, *Who_i did she_j say Mary_k kissed?*.
- ◆ P está en el dominio de sintagma nominal de N . Ejemplo: *[El hijo_i de él_j]*.
- ◆ P es el determinante de un nombre Q , y N está contenido en Q .

Las dos últimas condiciones evitan la correferencia en las siguientes frases: *Su_i retrato de Juan_j es interesante*, *John's portrait of him_j is interesting*, *Su_i descripción del retrato realizado por Juan_j es interesante*.

También en estos trabajos de Lappin y McCord, se refinan las anteriores reglas para formar el siguiente algoritmo para resolver las expresiones anafóricas formadas por pronombres reflexivos y recíprocos (*cada uno*). Para ello define el concepto usado en la formulación del algoritmo de **hueco argumento (o complemento) mayor** en base a la jerarquía de huecos argumento siguiente: *suj > agente > obj > (iobj | pobj)*, donde *suj* es el hueco sujeto, *agente* es el agente de una oración pasiva, *obj* es el hueco objeto directo, *iobj* es el hueco objeto indirecto y *pobj* es el objeto de un complemento preposicional de un verbo. El algoritmo consistirá en la siguiente regla: un sintagma nominal N es un posible antecedente para un pronombre anafórico A si N y A no tienen incompatibilidades de concordancia y se cumple una de las siguientes cinco condiciones:

- ◆ A está en el dominio del argumento de N , y N llena un hueco argumento mayor que el que llena A . Por ejemplo: *They_i wanted to see themselves_i. Mary knows the people_i who John introduced to [each other]_i.*
- ◆ A está en el dominio adjunto de N : *He_i worked by himself_i. Which friends_i plan to travel with [each other]_i.*
- ◆ A está en el dominio de sintagma nominal de N : *John liked Bill_i's portrait of himself_i.*
- ◆ N es el argumento de un verbo V , hay un sintagma nominal Q en el dominio del argumento o en el dominio adjunto de N de manera que Q no lleva determinante, y A es:
 - ◇ un argumento de Q , o
 - ◇ A es un argumento de una preposición $PREP$ y $PREP$ es un adjunto de Q : *They_i told stories about themselves_i.*
- ◆ A es el determinante de un nombre Q , y
 - ◇ Q está en el dominio del argumento de N y N llena un hueco argumento mayor que Q , o
 - ◇ Q está en el dominio adjunto de N : *[John and Mary]_i like [each other]_i's portraits.*

El siguiente ejemplo muestra el efecto combinado de las dos últimas condiciones: *[John and Mary]_i like [each other]_i's portraits of themselves_i.*

En el trabajo de Popowich [138] se trata ampliamente el fenómeno de las referencias producidas por los pronombres reflexivos, definiendo las siguientes restricciones sintácticas:

- ◆ *Restricciones de localidad*: el pronombre y antecedente han de aparecer en la misma oración, como por ejemplo en: *Mary_i loves herself_i. John_i found a picture of himself_i. Mary persuaded John_i to kiss himself_i. [A man who loves a woman]_i loves himself_i.* Esta restricción nos permite en la siguiente oración: *Jane said that Mary_i loves herself_i,* eliminar a *Jane* como posible antecedente del pronombre.
- ◆ *Restricciones c-dominio*: el pronombre debe ser c-dominado por su antecedente. Esta restricción es similar a la anterior de localidad, por lo que exige que antecedente y pronombre aparezcan en la misma cláusula.
- ◆ *Restricciones temáticas*: el rol temático del pronombre no debe ser mayor jerárquicamente que el de su antecedente. Esta jerarquía de roles temáticos sería la siguiente: *agente, localización, fuente, objetivo y tema*, donde *agente* tendría la mayor jerarquía y *tema* la menor. De este modo se considerarían agramaticales las siguientes oraciones: **Juan_i fue asesinado por [él mismo]_i,* ya que el pronombre toma el rol de agente, mientras que el sujeto toma el de tema. **Juan habló sobre Pedro_i, a [él mismo]_i,* ya que el pronombre es el objetivo, mientras que *Pedro* es el tema. Sin embargo sí que sería correcto en: *Juan_i habló a Pedro sobre [él mismo]_i.*

En [157] Stuckardt confirma estas restricciones de localidad proporcionando el siguiente ejemplo de correferencia: *The client_i appreciates that the barber shaves him_i,* y cuya siguiente variación sería incorrecta: **The client_i appreciates that the barber shaves himself_i.* Aunque tal y como destaca Stuckardt, también se permiten referencias a elementos externos a ese dominio local, como ocurre en la siguiente frase: *The barber hears his_i story about himself_i.*

Según se desprende de lo expuesto hasta ahora, las restricciones sintácticas para la resolución de pronombres difieren en función del tipo de pronombre: ya sea reflexivo o no. Chomsky en su *teoría de rección y ligamiento (Government and Binding Theory)* [35][33][34] realiza un estudio comparativo entre ambos tipos de pronombres. Él nos indica que los pronombres reflexivos y recíprocos precisan obligatoriamente de la presencia de un antecedente sin el cual no es posible interpretarlos. Por tanto son unidades carentes de referencia independiente: *Pedro_i se_i insultó a [sí mismo]_i. [Pedro y Luis]_i se_i miraban [el uno al otro]_i*. Y es característico de este tipo de unidades (*a sí mismo, el uno al otro*) que su antecedente sea el sujeto de su misma oración. Esto también se cumple en las oraciones de infinitivo con formas reflexivas o recíprocas, ya que el sujeto léxicamente vacío del infinitivo es obligatoriamente correferencial con el sujeto de la oración principal: *Pedro_i deseaba insultarse a [sí mismo]_i. [Pedro y Luis]_i decidieron mirarse [el uno al otro]_i*. También se cumple para las oraciones subordinadas con sujeto vacío: *Pedro_i aseguró que se había insultado a [sí mismo]_i. [Pedro y Luis]_i decidieron que nunca más se mirarían [el uno al otro]_i*.

El resto de pronombres según Chomsky, cuando tienen un antecedente es condición obligatoria que éste se encuentre en una oración o cláusula distinta de la del pronombre (lo cual coincide con las restricciones c-dominio vistas hasta el momento): *Luis_i habló con él_j. Luis_i lo_j lavó. Luis_i aseguró que él_i no había sido el autor de la fechoría. Luisa_i aseguró que ella_j la_j había perdonado*. Esto se podría contradecir en la siguiente frase: *Luisa_i criticó la devoción de Pedro por ella_j*, en la que la interpretación más probable sería la de no correferencia entre *ella* y *Luisa*, aunque quizás cabría la posibilidad de correferencia que se vería reforzada en el caso de utilizar el pronombre reflexivo *ella misma*. Por lo que respecta a su naturaleza semántica, el pronombre tiene cierta autonomía referencial ya que puede aparecer sin antecedente.

Chomsky muestra los siguientes casos en los que el límite dentro del cual se desarrollan los fenómenos de ligamiento es el propio sintagma nominal: *Luisa criticó la devoción de Pedro_i por [sí mismo]_i. Luisa criticó la devoción de Pedro_i por él_j. Luisa_i criticó la devoción de Pedro por ella_j. *Luisa_i criticó la devoción de Pedro por [sí misma]_i*.

En [137] Pollard y Sag distinguen a su vez entre dos tipos de pronombres reflexivos: los reflexivos ordinarios y los reflexivos del tipo "foto" (*picture noun reflexives*, denominación tomada del ejemplo: *John_i found a picture of himself_i*). Pollard y Sag argumentan que ambos tipos de reflexivos no tienen el mismo tratamiento. Como ejemplos de reflexivos ordinarios muestra los siguientes¹⁵: *John_i hates himself_i. The men_i admired [each other]_i. Mary_i explained Doris_j to herself_{ij}. Dana_i talked to Gene_j about himself_{ij}. The men_i introduced the women_j to [each other]_{ij}*. Y como ejemplos de reflexivos del tipo foto: *John found [a picture of himself]. The women selected [pictures of each other]. The men admired [each other's trophies]. The men introduced the women to [each other's spouses]*. En estos últimos Pollard y Sag indican que no requieren un antecedente que los c-domine tal y como ocurre en las siguientes frases: *A fear of himself_i is John_i's problem. The pictures of [each other]_i with Ness made [Capone and Nitty]_i somewhat nervous*.

¹⁵ Denotaremos los casos ambiguos en los que cabe la posibilidad de diferentes antecedentes para la misma expresión anafórica por medio del subíndice de cada antecedente separados por una barra (/). Por ejemplo: *ella_{ij}*.

Finalizando con el apartado de restricciones sintácticas para el tratamiento de la anáfora pronominal, Hirst [80] destaca otro tipo de información sintáctica necesaria para el tratamiento de ciertos casos de anáfora: el *paralelismo sintáctico* entre constituyentes. Según Hirst, un sistema que pretenda resolver la anáfora necesita un conocimiento de paralelismo que es particularmente importante para resolver la *anáfora superficial numérica* ya explicada en el apartado 2.2.4.3 (aquellas que están formadas por ordinales y expresiones que signifiquen cierto grado de orden: *el primero, el último, el tercero*, etc.). En este tipo de anáfora se requiere almacenar la estructura superficial de la frase o al menos el orden de los posibles antecedentes. Por ejemplo, en la oración: *Pedro miró al perro_i, al gato_j y al conejo_k, pero finalmente eligió el primero_i*, es necesario conocer el orden de los sintagmas coordinados (*el perro* será el sintagma nominal número uno, *el gato* el número dos, etc.) para poder resolver la anáfora *el primero*.

En muchos trabajos (por ejemplo los de Mollá [119] y Rico [143]) también se utiliza la noción de paralelismo sintáctico consistente en la preferencia de la expresión anafórica por aquellos candidatos que se encuentren en *posiciones paralelas*. Un ejemplo de la tendencia de que los antecedentes y expresiones anafóricas ocupen posiciones sintácticas paralelas podría ser el siguiente texto: *Pedro_i visitó a su hermano_j. Él_i le_j enseñó su_i coche nuevo*, en el que además se destaca cómo los roles semánticos de los pronombres reproducen los correspondientes roles de sus antecedentes. Carbonell y Brown en [24] nos resaltan que las posiciones paralelas pueden ser además de sintácticas, también semánticas y pragmáticas tal y como veremos en los próximos apartados en los que estudiaremos en detalle estos nuevos tipos de información.

Otro tipo de información sintáctica interesante en el tratamiento de la anáfora es la correspondiente a la estructura gramatical de los constituyentes analizados, información especialmente útil a la hora de resolver la *anáfora de tipo "one"* que recordamos consiste en aquellos casos producidos en el inglés en los que la expresión anafórica tiene la siguiente estructura: el pronombre *one* acompañado de diversos modificadores. Por ejemplo, en *Wendy didn't give either boy [a green tie-dyed T-shirt]_i, but she gave Sue [a red one]_i*, el sintagma nominal *a red one* introduce la siguiente entidad: *a red tie-dyed T-shirt* para lo cual se está utilizando la información correspondiente a la estructura sintáctica del sintagma nominal antecedente de la expresión anafórica (*a red one*) para construir la nueva entidad del discurso a la que se está referenciando (*a red tie-dyed T-shirt*).

3.1.4 Información semántica

La información semántica utilizada en el tratamiento de la anáfora se estructura habitualmente en forma de rasgos semánticos que se asignan a cada entidad del discurso. Esta información se suele utilizar obligando a que haya compatibilidad entre los rasgos semánticos de la expresión anafórica y su antecedente. Como ya hemos expuesto anteriormente, los pronombres disponen de poca carga léxica por lo que además tampoco dispondrán de estos rasgos semánticos. Estos rasgos semánticos los heredarán generalmente del verbo al que modifican. Por ejemplo, en la siguiente frase: *El ratón_i se paró cerca del coche. Éste_i encontró un trozo de queso_j y se lo_j comió*, se puede determinar que el pronombre *éste* se referirá al *ratón* y no al *coche* debido a que el verbo *encontrar* obliga a que el sujeto o agente de la acción sea del tipo *ser vivo (el ratón)*, y no del tipo *inorgánico (el coche)*. Estos rasgos semánticos se trasladan al pronombre ofreciendo una información

muy valiosa que en determinadas situaciones nos permite determinar el antecedente correcto. Un caso similar ocurrirá con el pronombre *lo*. Este pronombre hace la función de objeto directo del verbo *comer*, por lo que hereda el rasgo semántico de ser algo *comestible*, rasgo que únicamente será compatible con el antecedente *queso*.

Estas características de compatibilidad de rasgos semánticos se pueden determinar de diversas maneras. Por ejemplo, en [80] Hirst propone sencillamente tomar un valor de distancia semántica entre la expresión anafórica y sus posibles antecedentes para elegir el que tenga menor valor, es decir, esta distancia semántica se considerará únicamente como una métrica de qué grado de similaridad hay entre ambos.

En [24] Carbonell y Brown nos hablan de utilizar este tipo de información como restricción, es decir, eliminarán todas aquellas entidades discursivas cuyos rasgos semánticos no concuerden con las restricciones semánticas exigidas por la expresión anafórica. Por ejemplo, en la frase: *Juan cogió el pastel_i de la mesa y se lo_i comió*, aquí seleccionaría *el pastel* en lugar de *la mesa* ya que el verbo *comer* impone en su objeto directo la restricción de una entidad comestible.

También en el mismo trabajo de Carbonell y Brown se habla de *preferencias por posiciones semánticas paralelas*, según las cuales las expresiones anafóricas tienden a seleccionar el antecedente que comparten la misma categoría semántica, aunque se manifiesten en diferentes posiciones sintácticas en la oración. Por ejemplo, en *Juan llevó la caja de Pedro a Luis. Él también le_i llevó los libros de María*. Aquí el pronombre *le* y el sintagma nominal *Luis* comparten el rasgo semántico de receptor.

En [143] Rico observa que siempre debe existir *consistencia semántica* en la relación anafórica, y define tres tipos de consistencia semántica:

- ◆ Consistencia entre la expresión anafórica y su antecedente que ocurre mediante el empleo de sinónimos, hiperónimos¹⁶, relaciones del todo a una parte, del grupo al individuo y de hipónimos¹⁷. Este tipo de consistencia semántica, se manifiesta normalmente en aquellas expresiones anafóricas que tienen mayor carga léxica (descripciones definidas).
- ◆ Consistencia entre el verbo y la expresión anafórica. La expresión anafórica debe obedecer siempre las restricciones semánticas impuestas por el verbo. Por ejemplo, en: *The jury recommended that Fulton legislators act to have these laws, studied and revised to the end of modernizing and improving them.* El pronombre *them* referencia a *these laws* gracias a la necesidad de mantener la consistencia semántica con los verbos *improve* y *modernize*. Por el contrario, si en lugar de estos dos verbos, introducimos el verbo *fulfil* (cumplir), *them* se referirá a *Fulton legislators*.

¹⁶ Se dice que un objeto es *hiperónimo* de otro si lo abarca semánticamente, es decir, es más general. Por ejemplo *semana* es el hiperónimo de *lunes*, *martes*, *miércoles*, etc., de la misma forma que *árbol* es el hiperónimo de *roble*, *pino*, etc.

¹⁷ Se dice que un objeto es *hipónimo* de otro si éste es una especialización del segundo. Por ejemplo *roble* es un hipónimo de *árbol*.

- ◆ Consistencia dentro de la oración en la que se encuentra la expresión anafórica. Ésta es la más compleja porque utiliza el conocimiento del mundo para poder identificar el antecedente. En la oración: *The jury said it found the court has incorporated into its operating procedures...*, la expresión anafórica *its operating procedures* (procedimientos operativos) se refiere a *the court* (el tribunal), ya que *the jury* (el jurado) no tiene *operating procedures*. Aunque, como esta información es la más compleja de obtener, Rico encontró en su análisis del corpus *Susanne* que las relaciones anafóricas en las que el criterio semántico de inferencia es el definitivo para identificar el antecedente son muy poco numerosas (alrededor de un 6% del total de expresiones analizadas).

3.1.5 Información pragmática

Dentro de esta subsección comentaremos la información pragmática dividida en dos tipos que compondrán los dos siguientes apartados: información del mundo exterior e inferencia, y la información referente a la construcción del discurso.

3.1.5.1 INFORMACIÓN DEL MUNDO EXTERIOR E INFERENCIA

La información *pragmática* nos permite introducir cierta información del mundo exterior e inferencia de conocimiento aparte del que nos traen las propias palabras del lenguaje y sus usos. Esta información es fundamental en ciertas ocasiones en las que hay una gran ambigüedad a la hora de elegir entre diversos antecedentes, como ocurre en: *Pedro_i le dijo a Juan_j que él_{ij} era el hombre más feo que jamás había visto*, ejemplo en el que no queda claro quién es el hombre más feo, si *Pedro* o *Juan*. Habitualmente se escogería *Juan* ya que suele ser más común el comportamiento de crítica ajena que la auto-crítica (empleando conocimiento del mundo exterior).

Podríamos encontrar otro ejemplo en la siguiente frase: *No les des plátanos_i a los monos_j porque estos_i están verdes*, en la que se puede concluir que el pronombre *estos* se referirá a los *plátanos* porque sólo ellos y no los *monos* tienen la capacidad de estar *verdes*. O la siguiente: *Si [una bomba incendiaria]_i cae cerca tuyo_j, no pierdas [la cabeza]_j. Ponla_i en una cubeta y cúbrela_i de arena*, en la que es necesario saber que el objeto peligroso en este contexto es *la bomba* y no *la cabeza*. O en la frase muy utilizada en los textos sobre anáfora: *If the baby_i does not thrive on raw milk_j, boil it_j*, donde se ha de conocer que *la leche* puede ser hervida, mientras que los *bebés* no.

El siguiente ejemplo propuesto por Mollá en [119] nos da una idea de la importancia de esta información del mundo exterior. En este ejemplo aparecen dos frases que tan sólo se diferencian en una palabra, aunque esta palabra es suficiente como para que cambie el antecedente del pronombre *él*:

- ◆ *Juan_i vendió a Pedro su coche porque él_i lo odiaba.*
- ◆ *Juan vendió a Pedro_i su coche porque él_i lo necesitaba.*

En [24] Carbonell y Brown también hablan de *restricciones pragmáticas*, definiéndolas como restricciones impuestas por la acción que se realiza en la oración. Por

ejemplo, en *Juan le dio a Pedro una manzana. Él se comió la manzana*, el verbo *dar* impone la condición de que la persona que da algo a alguien deja de poseer ese algo. Si el antecedente de *él* fuera *Juan* se crearía un conflicto entre la condición del verbo *dar* y la de *comer* que establece que para que alguien coma algo es necesario que ese alguien posea ese algo.

3.1.5.2 INFORMACIÓN REFERENTE A LA CONSTRUCCIÓN DEL DISCURSO

Dentro de esta información pragmática también se incluye cierta información referente a la construcción del discurso en el que se desarrolla la anáfora. Por ejemplo, en [24] Carbonell y Brown consideran como un criterio de preferencia pragmática según el cual se establece que debe darse mayor prioridad a los posibles antecedentes que se encuentren en posiciones temáticas o topicalizadas. Como ocurriría en *María dijo a Ana de ir a Nueva York. Por qué hizo ella esto?*, en la que *María* está en una posición temática, por lo que sería el antecedente preferido por el pronombre *ella*. Igualmente ellos consideran que son preferibles a igualdad de condiciones los antecedentes que se encuentren en la misma oración que la expresión anafórica y además seleccionan como antecedente preferido aquel que ayude a mantener la cohesión discursiva. Por ejemplo, en *María condujo del parque al club. Pedro fue allí también*, la primera parte del discurso crea una acción de movimiento del parque al club. En la segunda oración debemos interpretar *allí* como refiriéndose al *club* si queremos mantener la cohesión discursiva.

Este tipo de información toma un papel fundamental en los algoritmos basados en la denominada *teoría del foco del discurso*, teoría que plantea un tratamiento especial de la anáfora a partir del estudio de la estructura del discurso. Estos algoritmos se basan en considerar como primer antecedente posible el foco del discurso, conceptos que expondremos con todo detalle en el apartado 3.2.2.4.

Un trabajo muy interesante es el realizado en [143] por Rico, en el que estudió sobre un corpus de frases en inglés la relación de diversos aspectos del discurso con el tipo de expresión anafórica. Los aspectos estudiados fueron la distancia, competencia, prominencia y unidad. Los resultados de este estudio los mostramos en los próximos apartados.

3.1.5.2.1 Distancia

Los resultados del estudio referenciado respecto la distancia entre la expresión anafórica y su antecedente se muestran en la Figura 10. Aquí se considera la distancia como el número de frases existente entre la expresión anafórica y su antecedente. Según se puede apreciar en estos resultados, las descripciones definidas y descripciones con demostrativo permiten mayor distancia con su antecedente. Esto es así debido a que son precisamente estas expresiones anafóricas las que tienen mayor carga léxica, esto es, comparten más rasgos con su antecedente por lo que éste se identifica con mayor facilidad. Los pronombres permiten generalmente muy poca distancia con su antecedente. Sólo lo permiten (16,3%) cuando entre la expresión anafórica y el antecedente no se encuentra ningún otro antecedente que pueda competir.

<i>Distancia</i>	<i>Expresión anafórica</i>			
	<i>Descripción definida</i>	<i>Descripción demostrativo</i>	<i>Pronombre personal</i>	<i>Pronombre posesivo</i>
0	6,35%	4,4%	30,8%	63,1%
1	39,4%	75,5%	46,8%	27,8%
2	10%	13,3%	5,9%	3,2%
3	44,1%	6,6%	16,3%	5,7%
TOTAL	99,8%	99,8%	99,8%	99,8%
<p>0 = El antecedente se encuentra en la misma oración 1 = El antecedente se encuentra una oración antes 2 = El antecedente se encuentra dos oraciones antes 3 = El antecedente se encuentra más de dos oraciones antes</p>				

Figura 10. Variación de la distancia en el tipo de expresión anafórica.

Estos resultados justifican el criterio de preferencia habitualmente utilizado en abundantes trabajos sobre la anáfora, por ejemplo el de Brown y Yule [22], que declaran que un pronombre tiende a referirse al último sintagma nominal introducido. Afirmación que generaliza Allen [3] para el resto de expresiones anafóricas denominándola *recency constraint*: tomar los antecedentes más cercanos que satisfacen todas las restricciones.

3.1.5.2.2 Competencia

En la Figura 11 se muestran los resultados asociados a la influencia de la competencia sobre el tipo de expresión anafórica. Esta competencia se mide en función del número de antecedentes posibles para una determinada expresión anafórica.

<i>Competencia</i>	<i>Expresión anafórica</i>			
	<i>Descripción definida</i>	<i>Descripción demostrativo</i>	<i>Pronombre personal</i>	<i>Pronombre posesivo</i>
1	5,9%	31,1%	88,4%	94,2%
2	10,5%	13,3%	6,4%	4,1%
3	83,6%	55,5%	5,2%	1,6%
TOTAL	100%	99,9%	100%	99,9%
<p>1 = Existe un único antecedente posible 2 = Existen dos posibles antecedentes 3 = Existen más de dos antecedentes posibles</p>				

Figura 11. Relación entre competencia y expresión anafórica.

De esta tabla se extrae que cuando existe un único antecedente, éste es altamente accesible para la expresión anafórica por lo que puede emplearse como medio de referencia anafórica una expresión con poca carga léxica, como sería un pronombre. En el caso de que

hubiesen varios antecedentes la expresión anafórica debe tener rasgos más marcados para que la identificación del antecedente correcto sea posible.

3.1.5.2.3 Prominencia

La influencia de la prominencia del antecedente en el texto, se realiza dependiendo de dos factores: en primer lugar la función sintáctica que el antecedente tenga en la oración, y en segundo lugar la repetición o recurrencia que éste muestre a lo largo del texto. Las funciones sintácticas que marcan mayor prominencia en inglés Rico las ordena del siguiente modo: en primer lugar el sujeto, a continuación el objeto directo y el objeto indirecto, y en último lugar los circunstanciales y los complementos regidos por el verbo. Mediante el segundo factor se permite la identificación de un antecedente que a pesar de no estar en posición sintáctica prominente consigue mediante la repetición hacerse prominente.

3.1.5.2.4 Unidad

En la Figura 12 se muestran resultados sobre la influencia de la unidad o localización de la expresión anafórica y su antecedente dentro de un mismo párrafo. Este concepto de unidad determina el efecto que la estructura discursiva tiene en la elección de las expresiones anafóricas. En el análisis el criterio de unidad se reveló como poco importante ya que la mayoría de las expresiones anafóricas se agrupan dentro de los valores 0 y 1, es decir, en el mismo párrafo o en el anterior.

<i>Unidad</i>	<i>Expresión anafórica</i>			
	<i>Descripción definida</i>	<i>Descripción demostrativo</i>	<i>Pronombre personal</i>	<i>Pronombre posesivo</i>
0	60,5%	73,3%	73,9%	91%
1	26,5%	22,2%	23,9%	8,1%
2	4,6%	4,4%	0,7%	0%
3	8,4%	0%	1,5%	0,8%
TOTAL	100%	99,9%	100%	99,9%
<p>0 = El antecedente se encuentra en el mismo párrafo 1 = El antecedente se encuentra en el párrafo anterior 2 = El antecedente se encuentra dos párrafos antes 3 = El antecedente se encuentra más de dos párrafos antes</p>				

Figura 12. Relación entre unidad y expresión anafórica.

3.1.6 Información sobre la expresión anafórica

Según vimos anteriormente, las características del proceso de resolución de la anáfora varían en función del tipo de expresión anafórica. En la Figura 13 se muestran los resultados del estudio realizado por Rico en [143] sobre la importancia de cada tipo de

información en la resolución de diferentes tipos de anáfora en inglés. Como se puede observar, se confirman nuevamente las anteriores afirmaciones según las cuales los pronombres tienen menor carga léxica que las descripciones definidas, con lo que la información léxica y semántica es menos importante.

<i>Tipo de información</i>	<i>Expresión anafórica</i>			
	<i>Descripción definida</i>	<i>Descripción demostrativo</i>	<i>Pronombre personal</i>	<i>Pronombre posesivo</i>
Sinonimia	84%	82%	0%	8%
Inferencia	8%	9%	0%	0%
Prominencia	5%	0%	21%	10%
Competencia	0%	0%	15%	0%
Distancia	0%	0%	50%	81%
Inf. verbo	0%	0%	13%	0%
Otros	3%	9%	0%	0%
TOTAL	100%	100%	99,9%	99,9%

Figura 13. Información necesaria para resolver cada tipo de expresión anafórica.

3.1.7 Información obtenida a partir del estudio del corpus

La aplicación de información obtenida a partir de corpus al problema de la resolución de la anáfora está creciendo rápidamente. Por ello, cada vez es mayor la cantidad de corpus disponibles para la evaluación de los sistemas de tratamiento de la anáfora (por ejemplo los corpus: *LOB*, *Susanne* o el *Penn Treebank*). Estos corpus son de uso general, es decir, pueden estar anotados sintácticamente y morfológicamente pero no están anotados respecto al discurso (roles temáticos o focos del discurso) ni respecto a las relaciones anafóricas entre diferentes constituyentes del corpus. La utilidad de los corpus anotados anafóricamente reside en poder utilizar información cuantitativa para restringir la búsqueda de antecedentes en sistemas reales *pobres en conocimiento*¹⁸ (*knowledge poor systems*).

En [106] McEnery, Tanaka y Botley muestran un breve repaso de los trabajos que se están realizando actualmente sobre la anotación de este tipo de información en corpus. Dentro de estos corpus y técnicas de anotación de los mismos destacan los siguientes:

- ◆ El *UCREL Anaphoric Treebank* consiste de 100.000 palabras anotadas morfosintácticamente provenientes de artículos de prensa. Este corpus fue

¹⁸ Tipo de sistema de tratamiento de la anáfora que restringe la cantidad de información necesaria para resolver la anáfora. Veremos con mayor detalle este tipo de sistemas en el apartado 3.2.2.3.

desarrollado como parte de un proyecto de colaboración entre UCREL e IBM, y fue anotado utilizando la técnica de anotación de discurso IBM/UCREL desarrollada por Fligelstone en [63]. Aquí se codificaron una amplia gama de características anafóricas y cohesivas. Estas características se asocian con unos determinados símbolos de anotación que codifican el tipo de relación y la dirección de la referencia en caso de ser relevante. Asimismo también es posible marcar grados de certeza en las anotaciones (tanto de la dirección de la referencia como del antecedente seleccionado), detección de múltiples antecedentes y rasgos semánticos para los pronombres en segunda y tercera persona.

El problema de este corpus es que sólo permite marcar relaciones anafóricas con antecedentes que aparecen explícitamente en el texto, y no cuando los antecedentes están relacionados con la expresión anafórica (por ejemplo cuando la expresión anafórica se refiere a un trozo extenso de texto).

- ◆ La técnica de anotación desarrollada por Rocha en [145] se aplica sobre textos obtenidos de conversaciones en inglés y portugués, los cuales se anotan de acuerdo a la estructura de tópicos del texto (concepto similar al de la teoría del foco del discurso que desarrollaremos en el apartado 3.2.2.4). En primer lugar Rocha establece para cada fragmento del discurso un *tópico global* o *tópico del discurso*, el cual puede permanecer durante todo el texto o cambiar en diferentes puntos del discurso. Este tópico estará formado por un sintagma nominal. A continuación divide el texto en segmentos de acuerdo a la continuidad del *tópico local*. Esto se realiza asignando un *tópico de segmento local* que es sólo válido en ese segmento, lo cual quiere decir que en el momento en que este tópico cambie también se considera que empieza un nuevo segmento. Estos cambios de segmento y de tópico local se anotan manualmente en el corpus. Finalmente Rocha anota cada caso de anáfora que se produzca en el texto especificando cuatro propiedades: *tipo de expresión anafórica*, *tipo de antecedente*, *grado de topicalidad del antecedente* e *información empleada para la resolución*.

Este corpus incorpora la innovación de intentar codificar la relación entre tópico (o foco del discurso) y los casos de anáfora. Asimismo clasifica las expresiones anafóricas y antecedentes de acuerdo a unos determinados criterios (las cuatro propiedades comentadas en el párrafo anterior), anotando además el tipo de conocimiento que considera necesario para resolver la anáfora. También este corpus está preparado para manejar otras relaciones al tener previsto la incorporación de textos multilingües. Sin embargo la principal desventaja de este corpus es que no utiliza estándares en los símbolos utilizados para la notación.

- ◆ En [65] Gaizauskas y Humphries superan la anterior desventaja del método propuesto por Rocha al utilizar las etiquetas *SGML* (*Standard Generalised Markup Language*) para anotar los casos de anáfora. Estas etiquetas tienen la ventaja de ser actualmente un estándar para la codificación electrónica de textos que permite su intercambio entre diferentes sistemas. Sin embargo tiene la desventaja de que sólo permite anotar cierto número de expresiones anafóricas y también presenta el inconveniente de su diseño orientado a tareas de resolución automática, donde el éxito de cada anotación ha de ser medido, es decir, no se diseñó con el objetivo de etiquetar grandes corpus.
- ◆ En [17] Botley desarrolla una técnica para describir los diferentes modos en que las expresiones demostrativas funcionan anafóricamente en textos escritos y hablados. Para ello clasifica las expresiones anafóricas demostrativas de acuerdo a

cinco criterios: *grado de recuperabilidad del antecedente*, *dirección de la referencia*, *tipo de referencia* (endófora o exófora), *función sintáctica* y *tipo de antecedente*. Este método al igual que el de Rocha presenta la ventaja de permitir marcar mayor información sobre los casos de anáfora que el método de UCREL.

En este mismo trabajo de McEnery, Tanaka y Botley [106] se comenta cómo se está trabajando en la universidad de Lancaster en colaboración con IBM Yorktown Heights en el desarrollo de un corpus en el que estén anotadas las referencias anafóricas. Actualmente este corpus no está terminado, aunque muestran algunos de los resultados obtenidos en el análisis. Entre estos resultados destacan que efectivamente los antecedentes de las expresiones anafóricas estudiadas presentan un modelo común, el cual varía en función del tipo de texto que se trate. Esta información puede resultar muy útil para adaptar la estrategia general de resolución anafórica al tipo de texto que se esté tratando, aunque tal y como expresan McEnery, Tanaka y Botley todavía estamos lejos de tener a nuestra disposición tal información.

Esta información obtenida a partir de análisis del corpus se está aplicando en diversos trabajos principalmente para la obtención de reglas de resolución especializadas para cada tipo de expresión anafórica y personalizadas para cada tipo de texto. Para obtener estas reglas se utiliza la información probabilística obtenida tras el estudio del corpus.

El principal inconveniente de la información anafórica obtenida a partir de corpus es la dificultad de conseguirla ante la ya comentada falta de corpus anotados adecuadamente. Además, actualmente esta anotación se está realizando prácticamente de modo manual, lo cual dificulta todavía más su progreso y su posterior empleo en aplicaciones reales de procesamiento del lenguaje natural. Un trabajo que intenta superar esta dificultad es la propuesta de Mitkov [115], en la que esboza un sistema para la anotación semiautomática de la anáfora pronominal en corpus. Para ello utiliza un sistema de resolución pobre en conocimiento seguido de una posterior fase de anotación y revisión manual de los resultados obtenidos. El sistema de resolución utilizado está detallado en el mismo trabajo de Mitkov [116] que desarrollaremos con mayor detalle posteriormente en la subsección 3.2.3, aunque de modo resumido podemos adelantar que trabaja sobre la salida de un etiquetador¹⁹, sin realizar un análisis sintáctico, y utiliza una serie de *indicadores de antecedente*, es decir, un grupo de reglas heurísticas dependientes del dominio que sirven para la detección del antecedente correcto. El dominio sobre el que trabaja es el de manuales de informática y equipos de sonido de alta fidelidad. Este sistema de anotación semiautomática no está desarrollado y se plantea como una posible futura colaboración entre la Universidad de Lancaster, British Telecom y la Universidad de Wolverhampton.

3.2 Algoritmos

Una vez expuestas con detalle y por separado cada fuente de información utilizada en el tratamiento de la anáfora, ahora entramos a describir diferentes algoritmos o sistemas que

¹⁹ *Part of speech tagger* o también conocido como *POS tagger*.

usan estas fuentes. La diferencia fundamental entre todos estos sistemas es o bien la cantidad o tipo de información que incluyen, o bien el modo en que organizan o coordinan estos tipos de información. Por ello, comenzaremos por reseñar las primeras aproximaciones al tratamiento de la anáfora que comparten la característica común de limitarse prácticamente a una sola fuente de información. A continuación dividiremos los sistemas restantes en dos grupos en función del tipo de información que emplean: sistemas integrados y sistemas alternativos. La diferencia fundamental entre ambos es que los alternativos se basan en la información obtenida a partir del estudio de corpus, mientras que en los integrados utilizan el resto de fuentes de información. A su vez el grupo de sistemas integrados lo dividiremos en cuatro subgrupos en función del modo en que coordinan las diferentes fuentes de información:

- ◆ Sistemas democráticos basados en restricciones y preferencias.
- ◆ Sistemas democráticos basados únicamente en preferencias.
- ◆ Sistemas pobres en conocimiento (knowledge poor systems).
- ◆ Sistemas consultivos basados en la teoría del foco del discurso.

De estos cuatro subgrupos los tres primeros dan igual protagonismo a cada fuente de información, mientras que en el último aparece una fuente de información que propone candidatos a antecedente, y las restantes fuentes de información se limitan a confirmar o rechazar estos candidatos.

3.2.1 Primeras aproximaciones al tratamiento de la anáfora

En esta subsección repasaremos brevemente las primeras aproximaciones al problema de la anáfora. Incluiremos en este apartado a la primera generación de investigaciones en el tratamiento del lenguaje natural y la resolución de la anáfora, cuya característica común es la de centrarse prácticamente en una sola fuente de información, no tratando de manera conjunta cada una de las vistas en la sección 3.1, y basándose principalmente en la utilización de reglas heurísticas para la resolución de ciertos casos de anáfora.

Las aproximaciones al problema de la anáfora realizadas en torno a los años sesenta y setenta tienen como inconveniente principal el que tienden a centrar la búsqueda de posibles antecedentes a aquellos encontrados dentro de la misma oración. Para ello utilizan únicamente conocimiento de tipo sintáctico para la localización del antecedente, encontrándose un gran número de casos de anáfora que no pueden tratar.

Un ejemplo de estas primeras aproximaciones es el sistema *STUDENT* desarrollado por Bobrow en [16]. Este sistema resuelve problemas de álgebra e incorpora unos pocos procedimientos heurísticos para resolver algunos tipos de anáfora y repeticiones incompletas. Por ejemplo, es capaz de detectar el antecedente del pronombre en el siguiente texto: *The number of soldiers the Russians have is half the number of guns they have. The number of guns is 7000. What is the number of soldiers they have?*. Sin embargo, estos procedimientos son fácilmente burlados ya que la oración no se analiza sintácticamente.

Algo similar ocurre con *ELIZA*, un programa desarrollado por Weizenbaum en [166]. Éste es uno de los primeros programas escritos para simular el comportamiento humano, el cual es capaz de imitar a un psiquiatra que dialoga con su paciente. Este programa no

consiste en ningún sistema experto con conocimiento de psiquiatría, sino que sencillamente se trata de un emparejado de estructuras (*pattern-matching*) con el objetivo de enlazar con la conversación del paciente (que el programa sea capaz de continuar la conversación a partir de la última frase del paciente). En cuanto al tratamiento de la anáfora se reduce al intercambio entre pronombres (por ejemplo será capaz de cambiar de primera y segunda persona, es decir, cuando el usuario utiliza el pronombre *yo*, al contestar ELIZA lo intercambia por el pronombre *tu*).

En [170][169] Winograd propone el sistema interactivo *SHRDLU* que permite al usuario entablar un diálogo con el ordenador para obtener información acerca de determinadas cuestiones. Los diálogos que el usuario puede mantener con el programa giran en torno a un campo temático concreto: el mundo de las figuras geométricas y las posiciones, formas y colores que éstas pueden tomar. Además *SHRDLU* tiene la posibilidad de aceptar instrucciones y ejecutarlas, para ello dispone de un brazo de robot para mover los bloques, una mesa donde se colocarán estos bloques y los bloques propiamente dichos. Este sistema puede tratar algunos pronombres personales, sintagmas nominales definidos y determinados casos de anáfora tipo "one".

Se compone de tres módulos: un analizador sintáctico o parser, un analizador semántico y un módulo que incorpora técnicas de razonamiento que permite resolver las cuestiones más complejas. Éste incorpora técnicas heurísticas mucho más complejas que las desarrolladas en el sistema *STUDENT*, permitiendo referencias a partes anteriores de la conversación entre el programa y el usuario. Para ello, almacena en una pila todos los grupos nominales que encuentra asignándoles a cada uno un valor numérico que indique la preferencia de la expresión anafórica por éste, y en caso de empate se solicita la intervención del usuario para determinar la referencia. Winograd utiliza la concordancia en número, género y persona entre la expresión anafórica y su antecedente, eligiendo el antecedente más reciente que pase estas restricciones de concordancia. Algunas de las reglas que emplea son fácilmente engañables, como sería la siguiente: "En el caso que aparezcan dos veces los pronombres *it* o *they* en la misma oración o en dos oraciones adyacentes, se considerará que estos pronombres son correferenciales", la cual no se cumple en la siguiente frase: *He put the box on the table. Because it_i wasn't level, it_j slid off.*

Como aspectos positivos de este sistema, se encuentra que es capaz de detectar e interpretar correctamente expresiones anafóricas que forman parte de su propio antecedente, como por ejemplo en: *...a block which is bigger than anything which supports it.* También maneja ciertos usos de *one* con significados opuestos, como los casos de pares de palabras como *grande* y *pequeño*, que se usan frecuentemente como opuestos, por ejemplo en *...a big green pyramid and a little one*, aquí la expresión *little one* significa *little green pyramid* y no *little pyramid* o *little big green pyramid*. Sin embargo esta detección fallaría en el caso de adjetivos que no tengan significados opuestos, como en el caso de: *a big blue one, a big green one and a little blue one.*

En [172][171] Woods describe un interfaz en lenguaje natural para una base de datos sobre minerales procedentes de las piedras traídas de la luna por la expedición *Apolo-11*. Este sistema se le conoció como *LSNLIS* (*Lunar Sciences Natural Language Information System*) o también como *LUNAR*. Tiene tres componentes básicos que procesan el lenguaje natural: un analizador sintáctico, un módulo de interpretación semántica y un último módulo encargado de la gestión de la base de datos. De modo similar al *SHRDLU*, para realizar el tratamiento de la anáfora almacena cada entidad que aparece en el discurso junto con sus respectivas representaciones sintácticas y semánticas. Distingue dos tipos de anáfora:

parcial y completa. La anáfora parcial se caracteriza por el empleo de un pronombre que se refiere a una parte de un sintagma nominal aparecido con anterioridad, por ejemplo en: *Give me [all analyses of sample 10046 for hydrogen]_i. Give me [them for oxygen]_i*. Estos casos los detecta por la presencia de una oración de relativo o un sintagma preposicional que modifica al pronombre (*for oxygen*). Para resolverlos efectúa una búsqueda entre todos los grupos nominales que han sido mencionados anteriormente, e intenta localizar aquél que tenga una estructura sintáctica y semántica paralela a la expresión anafórica empleada. En la anáfora completa aparece un pronombre que se refiere a un sintagma nominal completo, tal y como ocurriría en: *Which [coarse-grained rocks]_i have been analyzed for cobalt? Which ones_i have been analyzed for strontium?*. La estrategia de resolución depende de si la anáfora se construye mediante un demostrativo o un pronombre. Cuando la expresión anafórica la constituye un sintagma nominal introducido por un demostrativo, el método aplicado consiste en buscar un sintagma nominal cuyo nombre sea el mismo que acompaña al demostrativo: *Do any breccias_i contain aluminium? What are [those breccias]_i?*. En el caso que la anáfora se construya mediante un pronombre, entonces obtendrá información semántica de la frase en que aparece, información que procesará su módulo de interpretación semántica.

Este sistema sufre de las mismas limitaciones que los anteriormente descritos, es decir, sólo puede resolver expresiones anafóricas que tienen la misma estructura que sus antecedentes, por ejemplo en *Give me [all analyses of sample 10046 for hydrogen]_i*, ninguna de las dos siguientes frases se podría detectar como anáfora parcial: *Give me [the oxygen ones]_i*, *Give me [those that have been done for oxygen]_i*. Además de ello, también tiene la limitación de no poder tratar con la anáfora intrasentencial ya que los sintagmas nominales de esa oración no estarán disponibles hasta que se haya analizado completamente la oración.

En [30] Charniak describe un sistema cuyo dominio son historias infantiles. Éste tiene como objetivo principal estudiar el tipo de inferencias sobre el mundo real que son necesarias para hacer comprender al ordenador pequeñas historias, contestar a preguntas sobre estas historias y resolver problemas de ambigüedad anafórica. Trabaja sobre sintagmas nominales definidos y pronombres. Para realizar inferencias y razonamientos el sistema emplea un método esencialmente heurístico. La estrategia se basa en la aplicación de los llamados *demons* o *rutinas* que codifican hechos sobre el mundo que deben satisfacerse para llegar a la comprensión del texto. Estas rutinas se van activando a medida que se procesa la historia y aparecen conceptos nuevos. Para resolver anáforas, en primer lugar construye una lista de posibles antecedentes de la expresión anafórica, a continuación, si se comprueba que esta lista contiene un único candidato, éste se acepta como antecedente; en el caso de que la lista contenga más de uno, entonces la expresión anafórica se representa como una variable y se aplican diferentes *demons* con esta variable a la lista de antecedentes; finalmente, la variable que representa a la expresión anafórica quedará ligada a aquel antecedente que sea válido para el *demon* que contiene la variable. Por ejemplo, para el siguiente texto: *Janet put some money on the sink. Mother said, 'If you leave the money there it may fall in the drain'*, para encontrar el antecedente del pronombre *it*, el sistema consulta la lista de posibles antecedentes, en este caso *money* y *sink*. Para decidirse por uno de ellos, convierte al pronombre en una variable y la incluye dentro del *demon* que represente la información *it may fall in the drain*, interpretándose para ambos candidatos. Obviamente se obtiene como solución que el dinero sí que puede colarse por el desagüe, mientras que no ocurre así con el propio fregadero.

Este sistema presenta la desventaja de que a la hora de aplicar varios *demons* para resolver una relación anafórica siempre tiene preferencia aquel que se haya empleado antes,

lo cual no produce habitualmente resultados correctos. Además tiene el inconveniente de que aunque el dominio sea restringido se necesita gran cantidad de *demons* para codificar esta información.

En [167] Wilks construye un sistema de traducción del inglés al francés en el que implementa su teoría de la semántica de preferencias. Esta teoría utiliza patrones semánticos para interpretar las palabras en sus contextos. Para ello concibe el texto como un conjunto de bloques semánticos. De este modo, el texto se convierte en una unidad semántica que no necesita la sintaxis para su análisis. Cada bloque semántico o texto está compuesto por plantillas, las cuales se unen mediante patrones y reglas de inferencia de sentido común. El vínculo de unión entre estos elementos lo constituyen las fórmulas. Este sistema utiliza cuatro niveles de resolución de la anáfora pronominal dependiendo del tipo de pronombre y del mecanismo necesario para resolverlo. En el primer nivel, se aplican las plantillas y patrones existentes, utilizando únicamente conocimiento de las palabras que aparecen en el texto. Por ejemplo, en: *Give the bananas_i to the monkeys_j, although they_i are not ripe, because they_j are very hungry*, cada pronombre *they* es interpretado correctamente utilizando el conocimiento de que los monos son seres animados, por lo que tienen la posibilidad de estar hambrientos, y los plátanos al ser una fruta tienen la posibilidad de estar (o no) maduros. En el segundo nivel, se construyen nuevas plantillas semánticas que de alguna manera están implícitas en las plantillas que ya existen. Este nivel se aplicará en caso que el nivel anterior falle en encontrar un único antecedente del pronombre, por lo que se hace necesario utilizar métodos de inferencia para obtener conocimiento del mundo real. El tercer nivel se aplicará en caso que todavía haya más de un antecedente, en cuyo caso se intenta encontrar el tópico o foco de la oración que se considerará como antecedente. Este nivel supone el empleo de reglas de inferencia sobre el mundo real que van más allá de las definiciones y significados codificados en las fórmulas. En caso que los tres niveles anteriores fallen entonces es necesario seleccionar el antecedente por defecto (nivel cuatro), lo cual, equivale a suponer que todavía se está hablando de lo mismo que se había hablado hasta ese momento.

Al final de los años 70 se realizan diversos estudios sobre el uso de conocimiento del dominio en el discurso. Los sistemas construidos en esos momentos no son de uso general. Algunos ejemplos son *SAM* [44], *GUS* [15], y el *Task Dialogue Understanding System* de Grosz [71]. Este último realiza la distinción entre conocimiento del dominio, información del discurso y la intención del mismo. Grosz construye ese sistema como parte de su tesis doctoral. Su trabajo trata con la estructura del discurso y la noción de foco del discurso más que con el tema de la resolución de la anáfora, aunque supone que una correcta formulación de la estructura del discurso contribuye a la comprensión de la anáfora. Esta formulación de la estructura del discurso se englobará dentro de la denominada *teoría del foco del discurso*, que desarrollaremos en mayor profundidad en el apartado 3.2.2.4.

En [82] Hobbs desarrolla dos aproximaciones para resolver las referencias pronominales. Una de ellas utiliza únicamente información sintáctica, mientras que la otra realiza un análisis semántico del texto. En la primera aproximación Hobbs desarrolla un sencillo algoritmo que trabaja sobre los árboles sintácticos de las oraciones del texto. Estos árboles sintácticos representan la estructura gramatical del texto y se utilizan para realizar la búsqueda de los sintagmas nominales antecedentes de una determinada expresión anafórica, para ello utilizará las restricciones típicas como la concordancia en número, género y persona. Hobbs comprobó su algoritmo escogiendo 100 frases con diferentes ocurrencias de los pronombres *he*, *she*, *it*, y *they*, obteniendo un porcentaje de éxito del 81,8%. A pesar de

su porcentaje de éxito, Hobbs considera necesario el añadir información semántica al algoritmo, ante los casos en los que fracasa precisamente para conseguir un éxito del 100%.

En su segunda aproximación, Hobbs describe un sistema que comprende ciertas operaciones semánticas para desarrollar ciertos mecanismos de inferencia a partir de una base de conocimiento que almacena datos del dominio del texto. Aparecen cuatro operaciones básicas: detectar o verificar relaciones entre oraciones, interpretación de predicados, eliminación de redundancias e identificación de entidades. Estas operaciones se diseñan con el objetivo de reconocer la estructura e interrelaciones implícitas en el texto. Esta aproximación presenta problemas computacionales, como por ejemplo la complejidad exponencial en los procedimientos de búsqueda. Otro inconveniente de este algoritmo es precisamente la ausencia del módulo de análisis sintáctico que construya el árbol sintáctico correcto para todas las oraciones.

En la tesis doctoral de Webber [165] se describe una aproximación computacional de la anáfora en la que considera el discurso como una colección de diferentes tipos de entidades (individuales, conjuntos, acontecimientos o acciones). Estas entidades del discurso serán los antecedentes de las expresiones anafóricas. En el modelo de Webber aparecen lo que él denomina *invoking descriptions (ID)* de cada entidad del discurso, que significan la primera mención de una entidad del discurso, y las expresiones anafóricas tendrán como antecedentes estas ID. Webber trata con la anáfora de tipo "one", pronombres y algunos casos de cuantificación.

En [4] Alshawi desarrolla un sistema de resolución de la anáfora que se basa en dos mecanismos fundamentales: el de memoria y el contextual. Ambos mecanismos se emplean para la identificación de antecedentes, para resolver ambigüedades, y para generar enunciados a partir de un texto y almacenarlos en una base de datos.

El primero de estos mecanismos, el de memoria, se utiliza para almacenar información a medida que el sistema analiza un texto. Para poder llevar a cabo su tarea, emplea una serie de reglas semánticas formales que se definen mediante dos tipos de función (*ref* y *rel*) y dos tipos de afirmación (*specialisation* y *corresponds*). La función *ref* establece una referencia entre una entidad y el conjunto de objetos del mundo que ésta representa. Por ejemplo, la regla *ref (COMPUTER)* crea la referencia entre la entidad *computer* y los objetos del mundo real que corresponden a ella. La función *rel* establece una relación entre un par de entidades y el conjunto de pares de objetos que corresponden a éste en el mundo real. Respecto a los tipos de afirmación, la *specialisation* indica la referencia de un subconjunto a un conjunto mayor y la *corresponds* expone la relación que existe entre un par de entidades y otro par de entidades correspondientes a un conjunto mayor.

El segundo mecanismo, el contextual, señala la información más relevante en el texto. Para ello se definen siete factores contextuales que se aplican al texto a medida que éste se analiza:

- ◆ Factor que señala la oración más reciente en el texto.
- ◆ Factor que indica qué párrafo es el más reciente.
- ◆ Factor que indica la relevancia del sujeto en las oraciones construidas con el verbo *to be* y las oraciones de pasiva (*énfasis sintáctico*).

- ◆ Factor que indica el camino o historia que han seguido las operaciones del mecanismo de memoria.
- ◆ Factor que señala el uso de *deixis*²⁰ (que sirve para dirigir la atención del receptor desde un interés pre-existente a otro nuevo).
- ◆ Factor de asociación entre nodos semánticos cuyos vecinos hayan sido activados anteriormente.
- ◆ Factor que asocia un enunciado creado en la base de datos con todas las entidades correspondientes.

Cada factor consiste en un grupo de entidades a las que se asocia un valor numérico que determina su peso o importancia en el conjunto del texto. Para localizar un antecedente en una relación anafórica se lleva a cabo una búsqueda entre todas las entidades cuyo peso supere un umbral determinado. Si sólo existe una entidad que supere este umbral se aceptará ésta como antecedente. Si son varias las entidades propuestas se tomará como antecedente aquella cuyo peso sea mayor.

3.2.2 Sistemas integrados basados en el conocimiento

En la subsección anterior hemos revisado brevemente las primeras aproximaciones al problema de la anáfora en las que no se trata de manera conjunta cada una de las fuentes de información vistas en la sección 3.1, basándose principalmente en la utilización de reglas heurísticas para la resolución de ciertos casos de anáfora. A continuación estudiaremos los sistemas que utilizan diversas fuentes de conocimiento con el objetivo de tratar la anáfora. A estos sistemas se los puede encontrar en la literatura como *sistemas integrados basados en el conocimiento* (*integrated knowledge-based systems* o sencillamente *integrated systems*). Estos sistemas se diferencian de los *sistemas alternativos*, que expondremos en la siguiente subsección 3.2.3, en que no utilizan información obtenida a partir de corpus.

Una posible clasificación de los sistemas integrados es según el modo en que utilizan o coordinan estas fuentes de información. Según Carter en [27] y Rico en [143] se distinguen dos enfoques para la coordinación de todas estas fuentes: *democrático* y *consultivo*.

En el enfoque democrático todas las fuentes de conocimiento tienen asignado el mismo papel en la selección del antecedente anafórico, es decir, que aquellas entidades que pueden ser susceptibles de convertirse en antecedente pueden surgir por igual de aportaciones de la información morfológica, sintáctica, semántica o pragmática. Un claro ejemplo de enfoque democrático es el propio trabajo de Rico [143] justificando este enfoque según sus propias palabras, “porque de este modo los valores que aporta cada fuente de información son siempre relativos ya que no podemos decidir de antemano qué fuente es la

²⁰ Se usa el término *deixis* para designar la codificación en los enunciados de los datos referidos al contexto situacional, es decir, del lugar, tiempo y participantes. Tienen valor deíctico los pronombres demostrativos y personales, los adverbios de lugar, tiempo y modo y los tiempos verbales.

que aportará la información decisiva". Dentro del enfoque democrático distinguiremos cuatro apartados. En los dos primeros apartados estudiaremos diversos ejemplos de sistemas democráticos agrupados en función de cómo se considere cada fuente de información: como restricciones y preferencias, o sólo como preferencias. En el siguiente apartado hemos agrupado aquellos sistemas democráticos aplicados a la resolución de la anáfora pronominal que comparten la característica común de restringir la cantidad de información que necesitan para el tratamiento de este tipo de anáfora.

En el segundo enfoque, el consultivo, únicamente una fuente de información está capacitada para proponer antecedentes, y la función del resto de fuentes de información es confirmar o rechazar los antecedentes propuestos. Un ejemplo habitual de este segundo enfoque es el de los algoritmos basados en la teoría del foco del discurso, los cuales veremos en profundidad en el último apartado.

3.2.2.1 SISTEMAS DEMOCRÁTICOS BASADOS EN RESTRICCIONES Y PREFERENCIAS

En [24] Carbonell y Brown reconocen la necesidad de tratar la anáfora en toda su amplitud como un fenómeno en el que intervienen muchas y diferentes fuentes de información. Dividen los tipos de conocimiento en dos: *restricciones* y *preferencias*. Las restricciones sirven para eliminar candidatos, para las cuales Carbonell y Brown elegirían las siguientes fuentes de información: morfológicas, semánticas (eliminan todas aquellas entidades discursivas cuyos rasgos semánticos no concuerden con las restricciones semánticas exigidas por la expresión anafórica) y restricciones impuestas por la acción que se realiza en la oración. La última restricción va más allá de la anterior semántica, ya que utiliza información inherente de la acción que realiza el verbo, por ejemplo en *Juan le dio a Pedro una manzana. Él se la comió*, el verbo *dar* impone la condición de que la persona que da algo a alguien deja de poseer ese algo. Si el antecedente de *él* fuera *Juan* se crearía un conflicto entre la condición del verbo *dar* y la de *comer*, que establece que para que alguien coma algo es necesario que ese alguien posea ese algo.

Las *preferencias* facilitan la selección del antecedente correcto indicando qué candidatos se consideran mejores que otros. Como criterios de preferencia utilizaron la preferencia de la expresión anafórica por aquellos candidatos que se encuentren en posiciones paralelas, la preferencia por los antecedentes que aparecen en posiciones temáticas y preferencia por los antecedentes que se encuentren en la misma oración que la expresión anafórica.

Carbonell y Brown consideran que la preferencia por candidatos que se encuentren en posiciones paralelas se refiere tanto a paralelismo sintáctico, semántico o pragmático. Como posiciones sintácticas paralelas entienden aquellas en las que la expresión anafórica y su antecedente desempeñan la misma función sintáctica aunque la unidad lingüística que las representa sea distinta: *The robot gave the dog, a bone. John also gave it, some water*. Por posiciones semánticas paralelas consideran las relaciones que se dan entre una expresión anafórica y un antecedente que comparten la misma categoría semántica, aunque se manifiesten en diferentes posiciones sintácticas en la oración: *Juan llevó la caja de Pedro a Luis, Él también le, llevó los libros de María*. Aquí *le* y *Luis* comparten el rasgo semántico de receptor. Y finalmente por posición pragmática interpretan aquella que obliga a seleccionar como antecedente el que ayude a mantener la cohesión discursiva: *María condujo del parque al club. Pedro fue allí también*, la primera parte del discurso crea una

acción de movimiento del parque al club. En la segunda oración debemos interpretar *allí* como refiriéndose al *club* si queremos mantener la cohesión discursiva.

En cuanto a la preferencia por los antecedentes que aparecen en posiciones temáticas se establece que debe darse mayor prioridad a los posibles antecedentes que representen los tópicos o temas del discurso. Por ejemplo, en: *María dijo a Ana de ir a Nueva York. Por qué hizo ella esto?*. Aquí *María* está en posición temática por lo que se la prefiere como antecedente de *ella*.

El sistema propuesto por Carbonell y Brown tiene un módulo de tratamiento de las relaciones anafóricas que actúa después del análisis sintáctico y semántico con la siguiente secuencia de pasos:

- ◆ Considera como posibles antecedentes todos los sintagmas nominales que hayan aparecido con anterioridad.
- ◆ Aplica sobre ellos las restricciones y elimina así todos los candidatos que no las cumplan.
- ◆ Sobre los candidatos que quedan aplica las preferencias.
- ◆ Como resultado, cada candidato tendrá asignado un peso o valor correspondiente a la relevancia que le otorga cada preferencia. Por último, el sistema selecciona como antecedente el que mayor peso haya obtenido.

En [142] Rich y LuperFoy siguen la misma línea de trabajo de Carbonell y Brown, afirmando que deben crearse diferentes teorías para explicar los diversos problemas y aspectos que componen el tratamiento de la anáfora. Para ello, crean distintos módulos que se corresponden a su vez con diferentes teorías de la anáfora. Cada uno de estos módulos aborda un aspecto distinto de la identificación de antecedentes, e impone una serie de restricciones, por lo que reciben el nombre de *fuentes de restricción* (*constraint sources* o *CS*). Aparte de cada uno de estos módulos independientes hay un módulo superior, el *controlador*, que coordina las propuestas e intercambios entre todos los módulos. Cada *CS* tiene cuatro tipos de funciones:

- ◆ **Modelado:** es la función que mantiene el modelo de interpretación creado por cada *CS*. Por ejemplo, supongamos que el módulo activo es el que contiene información sobre el discurso. Para poder aplicar las restricciones sobre los posibles antecedentes se construye un modelo de interpretación discursiva que permanece activo mientras es necesario.
- ◆ **Exhibición de restricciones:** establece todas las restricciones que interactúan en la identificación de antecedentes asegurando así que la interpretación será correcta.
- ◆ **Formulación de hipótesis:** se encarga de proponer una lista de candidatos a antecedente mediante la asociación de una puntuación a cada uno de ellos en función de los resultados obtenidos tras aplicar las restricciones.
- ◆ **Evaluación:** evalúa los antecedentes propuestos por la función anterior.

Esta estrategia se lleva a la práctica en *LUCY*, un sistema de comprensión de textos. Para asegurar la independencia entre los diferentes módulos utilizan la llamada *arquitectura de pizarra*. Para ilustrar el funcionamiento de esta arquitectura imaginemos un grupo de especialistas dedicados a resolver un problema determinado. Todos ellos están reunidos ante una pizarra con el objetivo de comunicar sus conocimientos y buscar una solución al

problema. Por turnos cada uno escribe en la pizarra su contribución, de modo que el siguiente especialista pueda utilizarla y añadir nuevos datos. Este proceso continúa hasta que el problema está resuelto o ninguno de los especialistas puede añadir más información. Esta arquitectura se ha utilizado sobre todo en reconocimiento del habla, y se caracteriza porque utiliza diversas fuentes de información para proponer y evaluar diferentes hipótesis.

El algoritmo utilizado se resume en estos pasos:

- ◇ Análisis sintáctico y semántico.
- ◇ Actualización de la estructura discursiva.
- ◇ Se activan las funciones del *modelado* y el de *exhibición de restricciones*.
- ◇ Se activa el *controlador de la anáfora*, que a su vez, llama al de *formulación de hipótesis* y al de *evaluación*.
- ◇ El resultado es una lista de posibles antecedentes ordenados según la puntuación obtenida tras la aplicación de los CS.

Para la asignación de puntuaciones a cada candidato, emplean una función que calcula la media entre la puntuación absoluta que otorga cada CS a cada candidato (esta puntuación se sitúa entre -5 y +5) y el índice de confianza o seguridad de que el antecedente elegido es correcto (entre 0 y 1).

3.2.2.2 SISTEMAS DEMOCRÁTICOS BASADOS ÚNICAMENTE EN PREFERENCIAS

En el anterior apartado hemos visto sistemas que clasifican las fuentes de información que intervienen en el tratamiento de la anáfora en restricciones y preferencias. Las restricciones tienen como objetivo eliminar antecedentes para una determinada expresión anafórica, mientras que las preferencias ordenan los antecedentes que han quedado tras aplicar las anteriores restricciones para seleccionar el antecedente *preferido*. En este apartado veremos otra alternativa de manejo de estas fuentes de información en la que se eliminan las restricciones, considerándose todos los tipos de información como preferencias.

Un ejemplo de sistema basado únicamente en preferencias es el propuesto por Rico en su tesis doctoral [143]. Este sistema está basado en el producto escalar de vectores y se fundamenta en los siguientes tres procesos:

- ◆ Tratamiento simultáneo de toda la información lingüística (sintáctica, morfológica, semántica y pragmática).
- ◆ Asignación de relevancia a cada fuente de información en función del contexto lingüístico.
- ◆ Comparación en términos de igualdad de cada antecedente posible y su expresión anafórica, es decir, todas las entidades discursivas tienen las mismas posibilidades de ser consideradas como antecedente, por lo que no se excluye a priori ningún candidato (todas las fuentes de información se consideran como preferencias).

Esta estrategia parte de la idea de codificar de forma numérica cada fuente de información representándola en forma de vector. De este modo cada expresión anafórica y posible antecedente tendrán asignado un vector numérico que simboliza los valores que tienen para cada fuente de información. Para realizar la comparación entre dos de estos vectores utilizará el producto escalar, ya que el número que se obtiene de este producto permite fijar la posición relativa entre ambos, permitiéndonos un mecanismo de

comparación entre vectores. Para saber qué posición relativa tiene un vector respecto del otro necesitamos conocer el ángulo que separa ambos para lo cual se utiliza la fórmula mostrada en la Figura 14.

$$v \cdot w = \sum_{i=1}^n v_i \cdot w_i$$

$$\cos\theta = \frac{v \cdot w}{\|v\| \cdot \|w\|} \quad , \quad \|v\| = \sqrt{\sum_{i=1}^n v_i^2}$$

Figura 14. Cálculo del ángulo de separación entre dos vectores basado en el producto escalar entre ambos.

La codificación de las distintas fuentes de información mediante vectores ya se utilizó en [160] por Sutcliffe para representar significados de palabras con el objetivo de llevar a cabo comparaciones entre distintas palabras. Después de realizar el cálculo del ángulo de separación entre ambos vectores, el que tenga menor ángulo indica que sus significados comparten muchos rasgos. Para realizar esta codificación Rico define un conjunto de atributos anafóricos aplicables de manera general tanto a entidades discursivas como a expresiones anafóricas, los cuales deben cumplir las siguientes condiciones:

- ◊ Deben contener toda la información lingüística necesaria para consolidar una relación anafórica.
- ◊ Se deben poder codificar en forma de valores numéricos.
- ◊ El conjunto de atributos debe ser el mismo tanto para las expresiones anafóricas como para las entidades discursivas.
- ◊ Debemos asignar los valores siguiendo siempre los mismos parámetros y criterios independientemente de que estemos tratando con una expresión anafórica o con una entidad discursiva.
- ◊ El conjunto de atributos debe ser finito y abarcar todos los rasgos posibles.
- ◊ El conjunto de atributos debe representar con exactitud el tipo de información necesaria para la identificación de antecedentes, y sólo dicha clase de información.

Rico elegirá los siguientes atributos anafóricos:

- ◆ Información morfológica: género (*neutro, masc, fem*), número (*sing, pl*), persona (*prim, seg, terc*).

Aquí resuelve el problema de una referencia a un nombre colectivo, según el cual la expresión anafórica con la que establecen referencia puede estar tanto en singular como en plural. Para resolver este problema el atributo de número del nombre colectivo lo pone tanto en singular como en plural, dejando que sean otras fuentes de información las que actúen de manera decisiva.

- ◆ Información sintáctica: sujeto, objeto (*directo, indirecto*), complemento (*circunstancial, preposicional*).
- ◆ Información semántica:
 - ◊ Rasgos semánticos de carácter general que aseguran la consistencia entre antecedente y expresión anafórica: *humano / no humano, animado / no*

animado, abstracto / no abstracto, objeto / no objeto, contable / incontable, sinonimia e hiperonimia.

- ◇ Rasgos que aseguran las restricciones de selección (dependen del tipo de verbo). Por ejemplo, el verbo de la expresión anafórica *modernize them* impone la restricción de que su objeto directo sea un objeto físico, por ello en el vector de *them* aparecerá el rasgo *objeto físico* en este campo.
- ◆ Información pragmática:
 - ◇ Prominencia (*alta, baja*)
 - ◇ Distancia oracional (*misma oración, diferente oración*), distancia clausal (*misma cláusula, diferente cláusula*).

Ejemplo de distancia. Para la expresión anafórica *himself* se exige que esté en la misma cláusula y en la misma oración, por ello su vector contendrá los siguientes valores: (*distancia oracional = misma oración, distancia clausal = misma cláusula*). Para los posibles antecedentes, se les pondrá uno u otro valor en función de si están en la misma oración o cláusula que la expresión anafórica. Sin embargo, para la expresión anafórica *him* se exige que esté en distinta cláusula y misma oración por lo que su vector será: (*distancia oracional = misma oración, distancia clausal = diferente cláusula*).

Ejemplo de prominencia. *Sandy walked her dog near a bull one day. It walked quietly along*, aquí *dog* tendrá más prominencia que *bull*, por lo que tendrá valor *alto*, mientras que *bull* tendrá valor *bajo*, y el pronombre *it* tendrá valor *alto*.

Respecto a los grados de aplicación de cada atributo se calcularán en función de la información lingüística que se extrae de la expresión anafórica y su antecedente. Para ello, se codificarán los grados de aplicación mediante valores numéricos, de tal modo que si un atributo se aplica en mayor grado que otro el valor correspondiente será más alto. Además clasificará los atributos anafóricos en dos tipos: *restrictivos* y *no restrictivos*. Para los primeros, su aplicación restringe las características que la entidad discursiva y la expresión anafórica pueden tener. Tendríamos como ejemplos los atributos morfológicos, semánticos y el atributo pragmático de distancia clausal. Los segundos, los atributos no restrictivos, no imponen ningún tipo de restricción a la relación que la entidad discursiva y la expresión anafórica puedan establecer. Como ejemplos aparecen los atributos sintácticos, la prominencia y la distancia oracional.

En cuanto a la asignación de valores numéricos a cada atributo, se definen dos reglas: una para los atributos restrictivos y otra para los no restrictivos. Para la primera, cualquier valor superior a 0 indica que el atributo está presente en la entidad discursiva cuyo vector se está definiendo. El valor 0 se asigna a aquellos atributos que no pueden aplicarse a la entidad discursiva en cuestión. Por ejemplo, *María* tendrá el siguiente vector en los atributos de género: (*neutro=0, masc=0, fem=3*), y *él* tendrá el siguiente: (*neutro=0, masc=3, fem=0*). De este modo al hacer el producto escalar entre *él* y *María* dará como resultado 0. En la segunda regla, la de los atributos no restrictivos, estos atributos sólo tienen un valor posible que será asignado sólo si se cumplen las condiciones necesarias para su aplicación y que no será asignado en caso contrario. Además este valor será único y diferente para cada uno de los atributos. En el caso de los atributos sintácticos podrían considerarse los siguientes valores: (*sujeto = 4, objeto directo = 3, objeto indirecto = 2, complementos = 1*). De este modo en *La casa es grande*, el antecedente *la casa*, tendrá el valor 4 (*sujeto*).

De este modo los valores numéricos asignados al conjunto de atributos que definen una entidad discursiva concreta constituyen el vector que representa esa entidad. Y la comparación con otros posibles antecedentes se realizará mediante el producto escalar entre los vectores de cada uno de los antecedentes y el de la expresión anafórica. Como resultado se obtendrá una lista de las entidades ordenadas según su proximidad con el vector correspondiente a la expresión anafórica. Para mantener los resultados del producto escalar dentro de una escala fija se efectúa una normalización de las componentes del vector, con el objetivo de conseguir que el peso de los elementos que componen cada vector se pondere a una misma escala para todos los vectores. Para ello, se multiplica cada componente del vector por sc (Figura 15), de modo que el resultado de multiplicar el cuadrado de los elementos normalizados sea igual al cuadrado de una constante C , que se mantiene igual en todo el sistema. De este modo, el producto escalar de dos vectores es siempre menor que 1 , a menos que los dos vectores sean idénticos (será más próximo a 1 cuanto más parecidos sean los vectores).

$$sc = \frac{C^2}{\sum_{i=1}^n P_i^2}$$

Figura 15. Constante a aplicar para la normalización de vectores.

En su implementación computacional, utiliza una gramática de cláusulas definidas, aprovechando el analizador sintáctico desarrollado por Amores en [5]. Para la identificación de unidades anafóricas recorre el enunciado de izquierda a derecha y anota en dos listas diferentes la aparición de una expresión anafórica o una entidad discursiva, cada una de ellas con su información sintáctica y morfológica.

Con referencia a la evaluación del sistema la principal crítica que surge está basada en la posible arbitrariedad a la hora de asignar manualmente estos valores. De todos modos, también apunta la posibilidad de probarlo con métodos conexionistas que asignen automáticamente estos valores en versiones posteriores. Este sistema resuelve manualmente los pronombres personales, reflexivos y las descripciones definidas, y computacionalmente sólo los pronombres personales y reflexivos. No resuelve computacionalmente las descripciones definidas porque necesitan fundamentalmente información semántica sobre su antecedente, y en muchas ocasiones conocimiento de inferencias sobre el mundo, concluyendo que el construir un módulo semántico que construya inferencias sobre conocimiento del mundo supone una gran complejidad. Respecto a sus limitaciones destaca que hay muchos tipos de anáfora por tratar, que no se ha implementado un módulo de gestión de la estructura discursiva y que no existe el ya mencionado mecanismo de generación de inferencias sobre conocimiento del mundo.

Una vez vistos varios ejemplos de sistemas integrados democráticos que utilizan dos distintos enfoques para el manejo de la información: o bien restricciones y preferencias, o bien solamente preferencias, es interesante comentar el trabajo de Mitkov [114] en el que estudia la importancia del enfoque elegido para coordinar todas estas fuentes de

información. Para ello comparará dos aproximaciones computacionales, cada una con diferentes estrategias de aplicación de estos factores²¹: una basada en restricciones y preferencias, y otra que utiliza únicamente preferencias sin descartar ningún antecedente asumiendo inicialmente que el antecedente que se examina es el correcto. La segunda estrategia a su vez utilizará la fórmula del *razonamiento con incertidumbre* (*uncertainty reasoning*) para aceptar o rechazar el antecedente en cuestión.

Ambas aproximaciones han sido estudiadas por Mitkov en diversos trabajos, la primera basada en restricciones y preferencias en [110] y la segunda basada únicamente en preferencias en [111]. En la primera se utiliza información morfológica, sintáctica, semántica y pragmática. Como restricciones escoge la concordancia, las restricciones c-dominio y la consistencia semántica. Y como preferencias aplica justo en este orden (los primeros criterios de preferencia serán los que más peso tengan en la elección final): la teoría del foco del discurso, paralelismo sintáctico, semántico y distancia entre la expresión anafórica y su antecedente.

En la segunda aproximación [111] se utilizan técnicas de *razonamiento con incertidumbre*, justificando el uso de estas técnicas en la base de que en el proceso del lenguaje natural en muchas ocasiones partimos de información incompleta (por ejemplo en situaciones en las que no se ha entendido el texto completamente). Este sistema trabaja con los mismos factores que el sistema anterior pero sin distinguir entre restricciones y preferencias, asignando a cada factor un valor numérico que indique su aportación a la identificación del antecedente. A este valor numérico le llamaré factor de certeza (*certainty factor* o *CF*). Estos factores de certeza (*CF*) tendrán valores entre $0 < CF_i < 1$, en caso que se cumpla ese factor, y valores entre $-1 < CF_f \leq 0$, en caso que no se cumpla. Esto quiere decir, que si por ejemplo se cumple la concordancia entre la expresión anafórica y el antecedente se le asignará el valor CF_c , y en caso contrario se le asignará CF_f a este factor (concordancia). El proceso de identificación del antecedente emplea esta estrategia de razonamiento con incertidumbre: cada antecedente se evalúa como correcto o no según cada factor, asignándole su valor CF_c o CF_f según hemos visto en el anterior ejemplo. Progresivamente se va aplicando cada uno de los factores, calculándose la suma total de los valores *CF* de cada uno de ellos. Los antecedentes se prueban de derecha a izquierda (en primer lugar los que estén más cercanos a la expresión anafórica) mientras no se alcance uno cuya suma total de sus *CF* sobrepase un determinado valor umbral. La suma total de los *CF* se calcula por medio de la expresión aritmética mostrada en la Figura 16, en la que CF_i representa el nuevo valor a partir del valor anterior CF_0 tras aplicar el factor *s*, el cual nos ha devuelto el valor CF_s . Por ejemplo, supongamos que un cierto antecedente ha alcanzado un valor $CF_0 = 0,5$ tras aplicar algunos factores, si a continuación comprobásemos el factor restricción c-dominio y nos devolviese $CF_s = 0,45$, el valor final $CF_i = 0,5 + 0,45 - 0,5 * 0,45 = 0,725$.

²¹ Mitkov denomina en sus trabajos a estas fuentes de información como *factores*.

$$CF_1(s, CF_0) = CF_s + CF_0 - CF_s * CF_0 \Leftrightarrow CF_s > 0, CF_0 > 0$$

$$\frac{(CF_s + CF_0)}{[1 - \min(|CF_s|, |CF_0|)]} \Leftrightarrow CF_s < 0 \text{ ó } CF_0 < 0$$

Figura 16. Esquema de cálculo del factor de certeza.

Ambas aproximaciones [110] y [111] utilizan los mismos factores: concordancia en género y número, paralelismo sintáctico, preferencia sobre las entidades que están en posiciones temáticas de la oración, consistencia semántica, paralelismo semántico, preferencia por determinadas posiciones sintácticas (en primer lugar el sujeto de la oración anterior y después el objeto directo), preferencias inducidas por verbos o sustantivos, preferencia por sintagmas nominales que aparecen repetidos en el texto o que aparecen en las cabeceras de sección o capítulo y por último la distancia respecto la expresión anafórica.

La comparación de ambas aproximaciones se llevó a cabo sobre textos de informática, obteniendo un 83% de precisión para la basada en restricciones y preferencias, y un 82% para la basada en la técnica de razonamiento con incertidumbre con un $CF_{umbral} = 0,7$. La elección del valor umbral será un elemento determinante del éxito del sistema tal y como comprobó Mitkov, ya que al aumentarlo hasta 0,8, la precisión bajó hasta el 71%. Sobre estos resultados Mitkov saca las siguientes conclusiones: ambos métodos funcionan bien para la mayoría de los casos presentándose muy poca diferencia en cuanto a la precisión, aunque el primero parece ser más exacto pero menos seguro, poniendo como ejemplo las situaciones de excepción de la concordancia en número y género, es decir, situaciones en las que la expresión anafórica y su antecedente no concuerdan. Además, el segundo funciona mejor que el método basado en restricciones cuando se dispone de menos cantidad de conocimiento, aumentando su precisión cuantas más fuentes de información se emplean en la detección del antecedente.

Finalmente Mitkov sugiere la combinación de ambos métodos en una estrategia conjunta tal y como también propuso en [117]. Ésta consiste en que cada candidato se evalúa simultáneamente por ambos métodos, y en el momento que la respuesta de ambos coincide se detiene el proceso (acortando el proceso de búsqueda en comparación con el uso independiente de ambos métodos). De este modo, Mitkov considera que se ahorra un 10% del proceso de búsqueda en el caso del primer método, y un 33% en caso del segundo además de ganar en exactitud.

Como conclusión a su estudio, Mitkov considera que no es sólo importante el disponer de un conjunto adecuado de factores o fuentes de información lo que nos lleva a un correcto tratamiento de la anáfora, sino que también es necesario una clara estrategia de coordinación de estos factores. Continúa indicando que es difícil definirse por un sistema en el que hayan restricciones o por otro en el que todo sean preferencias, pero sí teniendo claro que en el caso de utilizar el primero, hay que tener muy en cuenta las excepciones a esas restricciones (caso de la concordancia en número y género).

Respecto al orden de aplicación tanto de restricciones como de preferencias, para las primeras considera que el orden es indiferente puesto que cada una de las restricciones va a

eliminar candidatos, mientras que para las preferencias sí que es importante puesto que algunas de esas preferencias van a entrar en conflicto en cuanto al antecedente seleccionado (casos de preferencia por antecedentes en la misma posición sintáctica y preferencias por el sujeto de la oración anterior). Del mismo modo, también considera que no todas las preferencias se pueden aplicar a cualquier tipo de texto, como por ejemplo una preferencia que propone Mitkov por la que se seleccionan los antecedentes que estén en las cabeceras de sección o capítulo, preferencia que evidentemente se aplicará en textos que estén organizados en forma de secciones o capítulos. En definitiva, considera que queda mucho trabajo por hacer en cuanto al estudio de la interrelación entre todas estas fuentes de información.

3.2.2.3 SISTEMAS POBRES EN CONOCIMIENTO

Dentro de este apartado vamos a describir diversos sistemas democráticos que comparten la característica de intentar reducir la cantidad de información necesaria para la resolución de la anáfora pronominal. La cantidad de información que utilizan se limita prácticamente a la morfológica y sintáctica, eliminando la información semántica, inferencia de conocimiento e información del dominio o mundo exterior. El motivo de eliminar este tipo de información es fundamentalmente a que no se dispone habitualmente de la misma en aplicaciones reales de procesamiento del lenguaje natural, y también debido a que aún sin incorporarla se obtiene una precisión suficientemente aceptable: por encima del 80%.

Una de estas primeras aproximaciones corresponde al algoritmo ya comentado anteriormente propuesto en [82] por Hobbs, el cual utiliza únicamente información sintáctica y morfológica. Este algoritmo pone en práctica diversas restricciones sintácticas para las referencias pronominales. Para ello busca en el árbol sintáctico resultado del análisis de una oración para encontrar el antecedente de un pronombre. Una vez que encuentra el primer sintagma nominal que cumple estas restricciones, éste se acepta como antecedente. El orden que sigue para atravesar el árbol sintáctico es el siguiente: en primer lugar analiza la última oración que aparece en el texto y una vez que está dentro de ella recorre el árbol de izquierda a derecha y examina primero los nodos que están a un mismo nivel, para pasar después a un nivel más profundo (*breath-first*). Hobbs comprobó su algoritmo escogiendo 100 frases con diferentes ocurrencias de los pronombres *he*, *she*, *it*, y *they*, obteniendo un porcentaje de éxito del 81,8%.

Ya que este algoritmo trabaja únicamente sobre esta información sintáctica, sin tener en cuenta la información semántica y pragmática, resulta ser un algoritmo computacionalmente eficiente. Sin embargo, tiene la desventaja de estar muy limitado en el tratamiento de otros tipos de anáfora, y además no está implementado el módulo de análisis sintáctico que construya el árbol sintáctico correcto para todas las oraciones (este algoritmo no fue implementado, sino que los resultados fueron obtenidos tras un estudio manual).

En [95] Lappin y Leass describen un algoritmo para la resolución de referencias pronominales que trabaja únicamente sobre información sintáctica con un alto porcentaje de análisis correctos: un 85%. Kennedy y Boguraev [93] proponen un algoritmo para la resolución de la anáfora pronominal que es una versión modificada del anterior trabajo de Lappin y Leass. En contraste con ese anterior trabajo este algoritmo no necesita un análisis completo del texto. En su lugar trabaja sobre la salida de un etiquetador (*part-of-speech tagger* o *POS tagger*) enriquecido tan sólo con las funciones gramaticales de determinadas

palabras. Al trabajar sobre la salida de un etiquetador presenta la ventaja de permitir su aplicación sobre sistemas de procesamiento del lenguaje natural que no puedan emplear componentes de análisis sintáctico robustos y fiables.

El etiquetador devuelve un análisis muy simple de la estructura del texto: por cada palabra devuelve un conjunto de valores que indican sus características morfológicas, léxicas, gramaticales y sintácticas, dependiendo del contexto en el que aparece la palabra. A esta salida del etiquetador, su algoritmo añade la información correspondiente a la posición numérica de cada palabra en el texto, es decir, asigna un número secuencial a cada palabra en el texto. Esta información se utilizará para implementar las relaciones de precedencia.

La identificación de sintagmas nominales se obtiene a partir de reglas gramaticales que definen la composición de un sintagma nominal. Aparte de esta información gramatical se utilizará información referente al contexto en el que aparece el sintagma nominal, por ejemplo si se encuentra dentro de un sintagma preposicional o dentro de una oración de relativo. Además utilizará información concerniente al contexto de determinados pronombres, como podría ser el caso de *it*, e información como la del caso que apareciese como sujeto de un determinado grupo de verbos (*seem*, *appear*, ...), información que nos pueda dar determinadas pistas sobre el posible antecedente a elegir.

También utiliza otra estructura de información que almacena las entidades del discurso, cada una de ellas representadas por información sobre ella misma y el contexto en el que aparece. Sin embargo, la única información con la que contará sobre su relación con otros referentes del discurso es en forma de precedencia respecto a la posición numérica que anota en el texto. La ausencia de información explícita sobre relaciones configuracionales marcan la diferencia fundamental entre este algoritmo y el de Lappin y Leass [95] en el que está basado. Este conjunto de entidades del discurso es el que utilizará el algoritmo de resolución de la anáfora.

En primer lugar este algoritmo activa el procedimiento de interpretación que realiza un análisis oración tras oración, interpretando los antecedentes del discurso en cada oración de izquierda a derecha. Cada nueva entidad del discurso presenta dos posibles interpretaciones: o bien como un nuevo participante en el discurso, o bien como una referencia a otra entidad previamente descrita en el texto.

La correferencia se determina al eliminar primero aquellos antecedentes a los que la expresión anafórica no se puede referir, para después aplicar unos criterios de preferencia sobre los antecedentes restantes. Los criterios de eliminación de antecedentes son básicamente dos. El primero basado en la información morfológica: concordancia en número, género y persona. Y el segundo basado en las restricciones c-dominio ya comentadas en la subsección 3.1.3. Para implementar estas restricciones c-dominio, ya que no se tiene información configuracional, este algoritmo realiza inferencias de las funciones gramaticales según la precedencia entre constituyentes.

Con respecto a los criterios de preferencia a aplicar, Kennedy y Boguraev emplean diez condiciones basadas en información contextual, gramatical y sintáctica para determinar la elección de uno u otro candidato. Cada una de estas condiciones tiene asignada un valor numérico que se muestra en la Figura 17. La suma total de estos valores numéricos dará el valor total según el cual se ordenarán todos los candidatos para la correferencia. Este valor total lo denominan *saliency weight*. Tal y como reseñan Kennedy y Boguraev, estos valores son totalmente arbitrarios, considerándose tan sólo importante la relación estructural entre

ellos, es decir, lo que de verdad es importante es que *SENT-S* tenga el máximo valor y que el segundo máximo sea el correspondiente a *CNTX-S*, y así sucesivamente. Ellos justifican la relación estructural elegida por el estudio lingüístico y la experimentación realizada.

<i>SENT-S: 100</i>	<i>si está en la misma oración</i>
<i>CNTX-S: 50</i>	<i>si está en el mismo contexto</i>
<i>SUBJ-S: 80</i>	<i>si su función es la de sujeto</i>
<i>EXST-S: 70</i>	<i>si está en una construcción existencial</i>
<i>POSS-S: 65</i>	<i>si GFUN=possessive</i>
<i>ACC-S: 50</i>	<i>si GFUN=direct object</i>
<i>DAT-S: 40</i>	<i>si GFUN=indirect object</i>
<i>OBLQ-S: 30</i>	<i>si es el complemento de una preposición</i>
<i>HEAD-S: 80</i>	<i>si EMBED=NIL</i>
<i>ARG-S: 50</i>	<i>si ADJUNCT=NIL</i>

Figura 17. Valores numéricos asignados a las preferencias.

La suma total de estas preferencias para una determinada entidad no es fija durante la vida de la misma, sino que varía en función de su prominencia o reiteración en el discurso. De este modo, cuando un pronombre se refiere a una determinada entidad, su *saliency weight* se ve incrementado. En caso contrario, es decir, conforme pase el tiempo y la entidad no sea referenciada, este peso se verá disminuido hasta llegar a anularse.

Los candidatos son ordenados en función de este peso final, y el que tenga mayor peso es el que se determina antecedente del pronombre bajo consideración. En el caso de un empate, el antecedente más cercano al pronombre será el que se seleccione.

En cuanto a la evaluación de este algoritmo, lo aplican exclusivamente para la anáfora pronominal con un porcentaje de éxito del 75%, que aunque es menor que el del algoritmo en el cual se basa, Kennedy y Boguraev lo justifican indicando que han trabajado sobre textos más variados y menos formales (el de Lappin y Leass [95] conseguía un 85% con textos basados únicamente en manuales de informática).

En [48] Declerck propone un algoritmo basado en la unificación y que se lleva a cabo en la fase de análisis semántico. Este algoritmo resuelve la anáfora pronominal interoracional. La operación de unificación es la que se utilizará para comparar la información contenida en el pronombre con la de los posibles antecedentes, y ésta se aplicará en la fase de interpretación semántica de la frase, dentro del módulo de la gramática denominado de *refinamiento* (*refinement component of the grammar*). El módulo de interpretación semántica está basado en el lenguaje de representación *DPL* (*Dynamic Predicate Logic*). Este lenguaje DPL es fruto de la investigación de la interpretación del lenguaje de *cálculo de predicados de primer orden*, considerándose como un primer paso para conseguir una teoría de la semántica del discurso. DPL está basado en la sintaxis de la lógica de predicados estándar, pero propone una nueva (dinámica) interpretación de los cuantificadores y conectivas que permiten el ligamiento de variables dentro y fuera de su ámbito, dependiendo de las interpretaciones de las correspondientes expresiones del lenguaje natural. De este modo considerará que los pronombres actuarán como variables, suposición no compartida por algunos trabajos como por ejemplo el la *teoría de representación del discurso* o el de Heim [77].

El modo que DPL interpreta los distintos cuantificadores y conectivas es el que a continuación exponemos. Los cuantificadores existenciales y las conjunciones se consideran *dinámicos externamente*. Estos pueden ligar variables tanto dentro como fuera de su ámbito. Un ejemplo de este tipo de cuantificador sería el siguiente: *Un hombre, camina en el parque y él, silba. Él, es feliz*, en el que se mantendrá ligada la variable correspondiente al sintagma *un hombre*, fuera de su ámbito, unida con el pronombre *él*. El cuantificador universal y las implicaciones se considerarán *dinámicas internamente*, y sólo podrán ligar variables dentro de su ámbito. Por ejemplo, en: *Cada hombre, camina por el parque. Él, silba*, el pronombre *él* y su antecedente no compartirán la misma variable. La negación y la disyunción se considerarán como *estáticos*, por lo que no podrán ligar variables anafóricas. Por ejemplo, en la frase *Ningún hombre, camina por el parque. Él, silba*, o en la siguiente: *Un hombre, camina en el parque o él, silba*, nunca podrían compartir la misma variable, es decir, nunca podrían correferir, el pronombre y el antecedente.

En [157] Stuckardt propone un algoritmo para la resolución de la anáfora pronominal (tanto para pronombres reflexivos como no reflexivos) y la anáfora producida por sintagmas nominales definidos. Este algoritmo está basado principalmente en las restricciones sintácticas derivadas del trabajo de Chomsky en su *teoría de rección y ligamiento (Government and Binding Theory)* [35][33][34]. Stuckardt afirma que la implementación computacional de las restricciones sintácticas propuestas por Chomsky es inviable, ya que Chomsky las propone como un mero instrumento teórico. Chomsky asigna índices de modo aleatorio a cada sintagma nominal, y es durante la fase de interpretación semántica al traducirlos a su correspondiente fórmula lógica cuando los principios del ligamiento sirven como restricciones para filtrar la distribución de índices que se considerarán válidos para interpretar las correferencias. Una implementación directa de este procedimiento supondría una complejidad exponencial. La solución propuesta por Stuckardt supera esa complejidad exponencial, y también permite la interdependencia (*interdependency*) entre antecedentes. Esta interdependencia la interpreta que puede suceder fundamentalmente en tres situaciones. La primera cuando uno de los posibles antecedentes de un pronombre ya no está accesible, con lo cual se incumple el principio B de la teoría de Chomsky, como por ejemplo en la siguiente frase: **El barbero, contó al cliente, una historia mientras él, le, afeitaba*. La segunda corresponde a la situación en la que se ha de escoger entre antecedentes del discurso (fuera de la frase actual). Y la tercera sucede como consecuencia del ligamiento de cláusulas relativas.

El algoritmo aquí propuesto trabaja con una complejidad cúbica en su peor caso en función del número de sintagmas nominales, resolviendo aproximadamente un 90% de los pronombres aparecidos en textos sobre biografías de arquitectos. Este algoritmo consiste en los siguientes pasos:

- ◆ Por cada expresión anafórica *Y* se determinan el conjunto de posibles antecedentes *X*. Para ello se verifica la concordancia en número, género y persona entre *Y* y *X*. Además, en caso que el candidato *X* sea intraoracional se comprueban las restricciones sintácticas ya comentadas anteriormente.
- ◆ Por cada antecedente que quede en *X*, se le asigna un valor numérico que determina su admisibilidad (*plausibility*) como antecedente final. Este valor se basa en criterios ya comentados como el del paralelismo (*preferencia* de la expresión anafórica por aquellos candidatos que se encuentren en *posiciones paralelas*), proximidad, preferencia por el sujeto, etc. En función de este valor, se ordenarán todos estos antecedentes (de mayor a menor).

- ◆ También se ordenarán descendentemente las expresiones anafóricas en función del valor numérico de su mejor antecedente.
- ◆ Siguiendo el orden de expresiones anafóricas propuesto en el paso anterior, se irán asignando sucesivamente los mejores antecedentes de cada expresión anafórica siempre que no aparezcan problemas de interdependencia entre ellos. Por ejemplo, en la frase anterior del barbero, al determinar que el primer pronombre *él* referencia a *el barbero*, cuando posteriormente se resuelva el pronombre *le*, nunca se podrá correferir con el mismo antecedente.

En [118] Mitkov y Stys proponen otro sistema que necesita poca cantidad de conocimiento para la resolución de la anáfora pronominal en manuales técnicos tanto en inglés como en polaco. Utiliza la concordancia en número, género y persona como restricción y una serie de *indicadores de antecedente* (*antecedent indicators*) a modo de preferencias. Este sistema es una modificación del expuesto por Mitkov en [116] escogiendo sólo un subconjunto de los indicadores de antecedente expuestos en ese trabajo (los que tras un estudio previo considera más adecuados para los manuales técnicos). Cada uno de estos indicadores asignará valores numéricos a los antecedentes, escogiéndose finalmente como antecedente de la expresión anafórica el que tenga mayor suma de estos valores.

Mitkov y Stys trabajan sobre la salida de un etiquetador gracias al cual obtienen la información léxica y morfológica. El motivo que reseñan por el que desechan la información semántica y pragmática es la propia complejidad de introducirlas en aproximaciones prácticas a la resolución de la anáfora, tanto desde el punto de vista humano como computacional. El conocimiento que utilizarán se limita a una serie de reglas gramaticales correspondientes a sintagmas nominales, información morfológica (número, género y persona), una lista de términos y un conjunto de indicadores de antecedente (los cuales variarán en función del tipo de texto). En caso de empate se acudirá a dos criterios para seleccionar el antecedente correcto: en primer lugar se escogerá el antecedente con mayor valor para el indicador de reiteración léxica, y en segundo lugar en caso que todavía persista el empate, se escogerá el más cercano.

Los indicadores de antecedente se aplicarán sobre los sintagmas nominales encontrados en una distancia máxima de dos oraciones respecto la expresión anafórica, y los utilizados por Mitkov y Stys son los siguientes:

- ◆ Se preferirán como antecedente los sintagmas nominales definidos (los que están modificados por un artículo, demostrativo o posesivo) antes que los indefinidos. Para ello asignarán valor *0* si el sintagma nominal es definido, y valor *-1* si es indefinido.
- ◆ Son preferidos los sintagmas nominales que representan el tema (*theme*) del texto (información dada o conocida), en cuyo caso se les asigna valor *1* y en caso contrario valor *0*. La nueva información que se añade al texto, o *rheme* proporciona información sobre el tema del texto. Aquí aplicará la siguiente regla heurística para determinar el tema del texto: se escogerá el primer sintagma nominal de cada oración como tema de la misma.
- ◆ Los sintagmas nominales que representan términos del dominio del texto son más probables de ser el antecedente correcto (valor *1*, caso contrario *0*).

- ◆ Si el verbo es uno de los siguientes: *discuss, present, illustrate, identify, summarise, examine, describe, define, show, check, develop, review, report, outline, consider, investigate, explore, assess, analyse, synthesise, study, survey, deal* o *cover*, entonces se elige el primer sintagma nominal que los siga (valor 1 ó 0).
- ◆ Si el núcleo del sintagma nominal que precede al verbo es *chapter, section* o *table*, entonces se escogerá el que siga al verbo (valor 1 ó 0).
- ◆ Se preferirá un sintagma nominal si éste se repite varias veces. Se le asignará valor 2 si se repite dentro del mismo párrafo dos o más veces, 1 si se repite una vez y 0 en caso contrario.
- ◆ Si el sintagma nominal está en la cabecera de la sección y forma parte de la oración donde está la expresión anafórica, entonces se le asigna valor 1, caso contrario 0.
- ◆ Se escogerá el que ocupe la misma posición respecto al verbo que la ocupada por la expresión anafórica (valor 2 ó 0).
- ◆ Se preferirán en primer lugar los que estén en la cláusula previa (valor 2), después los que estén en la oración anterior (valor 1), después los que estén dos oraciones atrás (valor 0) y, finalmente, los que estén tres oraciones atrás (valor -1).
- ◆ Se le dará mayor preferencia (valor 0) a los sintagmas nominales que no formen parte de un sintagma preposicional (valor -1).
- ◆ Dada la estructura $(you) V_1 NP_1 \dots conj (you) V_2 it (conj (you) V_3 it)$, se preferirá como antecedente del pronombre *it* a NP_1 (valor 1).

Este sistema se evaluó sobre un manual técnico en inglés de una fotocopiadora Minolta obteniendo una precisión del 95,8%, justificando los errores por falta de información sintáctica y semántica. Para el polaco también lo aplican obteniendo una precisión del 92,1%.

En [158] Stuckardt parte de la idea según la cual el conseguir un análisis sintáctico único es un objetivo poco acorde con el estado actual de desarrollo del campo del procesamiento del lenguaje natural a causa de la ambigüedad estructural (ligamiento de constituyentes como por ejemplo sintagmas preposicionales, adverbiales u oraciones de relativo; determinación de roles o funciones sintácticas como podrían ser la de sujeto u objeto directo; etc.). Por ello considera que un sistema que trate la anáfora ha de ser robusto en el sentido de ser capaz de superar esa inexactitud en el análisis sintáctico, o sea, que sea capaz de trabajar con estructuras sintácticas deficientes o incompletas. Stuckardt clasifica los sistemas robustos que tratan con esas estructuras deficientes en dos tipos:

- ◆ *Modelos de descripciones superficiales (shallow description model)*, en los que se utilizan reglas heurísticas para reconstruir estas estructuras sintácticas. En estos casos el proceso de tratamiento de la anáfora se adapta al tipo de entrada de que dispone.
- ◆ *Modelos de descripciones deficientes (deficient description model)*, los cuales extienden el tratamiento de la anáfora para trabajar con este tipo de estructuras, con lo que se utiliza la información sintáctica hasta donde es posible.

El trabajo de Kennedy y Boguraev [93] comentado anteriormente lo encuadra dentro del primer tipo. También se comparan dos sistemas que trabajan con estructuras sintácticas

incompletas o deficientes encuadrados en el segundo tipo, con el objetivo de aprovechar la salida del sistema de resolución de la anáfora para aplicarlo a una posterior fase de desambiguación estructural. Ambos sistemas utilizan fragmentos lingüísticos para aplicar las restricciones basadas en la información sintáctica, deduciendo a partir de reglas heurísticas el resto de información configuracional no obtenida en la fase de análisis sintáctico. El primero de estos sistemas que utiliza reglas heurísticas se implementa para el alemán en el sistema de tratamiento de la anáfora descrito en un trabajo anterior del mismo Stuckardt en [157]. Con este sistema obtiene un 82% de precisión, en principio superior al 75% obtenido por Kennedy y Boguraev, aunque de nuevo no pudiéndose comparar ambos resultados al aplicarse sobre distintos textos e idiomas.

En [13] Baldwin presenta un sistema de resolución de la anáfora pronominal denominado *CogNIAC* con un 90% de precisión y un alcance (*recall*) superior al 60%. El sistema resuelve pronombres que no necesitan conocimiento del mundo ni un sofisticado tratamiento lingüístico. Lo que distingue a este sistema de los restantes es que únicamente resuelve pronombres de los que está completamente seguro, para los cuales no encuentra ninguna ambigüedad. Para ello exigirá que sólo quede un antecedente después de aplicar el algoritmo de resolución. Baldwin justifica esta afirmación indicando que hay frases definitivamente ambiguas, las cuales son también interesantes detectar, como por ejemplo la siguiente: *Pedro y Juan estaban trabajando juntos cuando de repente él se cayó.*

Este sistema utiliza la siguiente información: detección de oraciones, POS tagging, reconocimiento de sintagmas nominales (corregido manualmente), información del número y género, y árboles sintácticos parciales, es decir, elimina la información semántica y el conocimiento del mundo. Éste trabaja en el siguiente orden: en primer lugar se aplica el etiquetado palabra a palabra, para después reconocer sintagmas nominales simples (que no tengan anidados nuevos sintagmas nominales). Después identifica manualmente las cláusulas de cada oración, utilizando expresiones regulares para identificar sujeto, objetos y verbo. A continuación aplica una serie de restricciones (número, género y c-dominio) que limitan el número de antecedentes sobre los que se aplicarán las siguientes preferencias heurísticas:

1. *Único en el discurso*: si hay un sólo antecedente se escoge éste como solución (evaluación: 8 correctos, 0 incorrectos).
2. *Reflexivos*: se escoge el antecedente más cercano de la oración actual en caso que se trate de un pronombre reflexivo (16 correctos, 1 incorrecto).
3. *Único en la oración actual y previa*: si hay un único antecedente en la oración actual y la previa, se escoge éste (114 correctos, 2 incorrectos).
4. *Posesivos*: si la expresión anafórica es un pronombre posesivo y hay uno con la misma estructura en la oración anterior, se consideran ambos correferenciales (4 correctos, 1 incorrecto).
5. *Único en la oración actual*: si sólo hay un antecedente en la oración actual se escoge éste (21 correctos, 1 incorrecto).
6. *Antecedente y expresión anafórica son sujetos* de la oración anterior y actual respectivamente, entonces se escoge este antecedente (11 correctos, 0 incorrecto).

CogNIAC resuelve los pronombres de izquierda a derecha en el texto. Para cada pronombre se aplican las anteriores reglas en el mismo orden que se han citado, de tal

manera que si se cumple una de ellas no se continúa probando las demás. Si no se cumple ninguna de las reglas, entonces se deja sin resolver el pronombre.

También Baldwin compara su sistema con el algoritmo ya comentado anteriormente de Hobbs [82]. Este algoritmo de Hobbs presenta una precisión del 81,6%, pero a diferencia del método propuesto por Baldwin, intenta resolver todos los pronombres sin detectar las situaciones de ambigüedad y además ordena todos los antecedentes (en el de Baldwin sólo se escoge uno, sin evaluar los restantes). Baldwin aplicará ambos métodos sobre un texto narrativo en el que hay gran ambigüedad: una tercera persona cuenta la historia de otras dos personas del mismo sexo, y tratará únicamente pronombres singulares en tercera persona. Baldwin aplicará su método corrigiendo manualmente la detección de los sintagmas nominales con la intención de no degradar su sistema respecto al de Hobbs, el cual se ejecutaba manualmente. También evitará situaciones en las que se puedan arrastrar errores, es decir, en caso que al resolver un pronombre se produzca un error, lo corregirá antes de continuar el análisis. Con estas condiciones el algoritmo de Hobbs obtuvo un 78,8% de precisión y un 100% de alcance mientras que el de Baldwin obtuvo un 92% de precisión y un 64% de alcance. Baldwin también realizó pruebas ampliando el alcance de su sistema hasta un 100% para poderlo comparar con el de Hobbs, al añadir dos nuevas reglas a las anteriormente descritas:

7. Siguiendo la teoría del foco del discurso, la cual la implementó Baldwin en [12], cada oración tiene un centro focal que mira hacia atrás (*backward centre* o *Cb*), el cual une el enunciado al discurso precedente, y supone la entidad focalizada en la oración. Si existe un antecedente en la cláusula actual que representa a este *Cb*, entonces se escoge éste como solución del pronombre.
8. Se escoge el antecedente más cercano a la expresión anafórica en caso que no se cumpla ninguna de las reglas anteriores.

Con estas nuevas reglas el algoritmo de Baldwin obtiene un 77,9% de precisión, o sea, incluso inferior al de Hobbs.

Además Baldwin presenta los resultados de la aplicación de CogNIAC a la extracción de información dentro de la evaluación del MUC-6 [14]. Para ello se integró con otros módulos, como el analizador parcial desarrollado por Collins en [39], y también se modificó CogNIAC para adaptarlo al dominio sobre el que se trabajaría. Por ejemplo, se eliminaron la cuarta y octava regla de preferencia, y se le añadieron módulos para detectar usos no anafóricos del pronombre *it*. Con ello obtuvo un 73% de precisión bastante inferior a otros resultados, pero atribuyendo este bajo rendimiento a los errores inducidos por el resto de módulos (mal etiquetado de palabras, errores de detección de sintagmas nominales, etc.).

3.2.2.4 SISTEMAS CONSULTIVOS: TEORÍA DEL FOCO DEL DISCURSO

En los anteriores tres apartados hemos descrito sistemas integrados que utilizan un enfoque democrático para coordinar las distintas fuentes de información que intervienen en la resolución de la anáfora. En este apartado expondremos los sistemas que utilizan el otro enfoque, el consultivo. Este enfoque consultivo se caracteriza por la selección de antecedentes gracias a la información recogida por una única fuente. Una vez que se ha propuesto un antecedente se consultarán otros tipos de conocimiento con el fin de recoger información suficiente para confirmar o rechazar dicho antecedente.

La fuente de información que generalmente selecciona antecedentes en este tipo de sistemas consultivos es la correspondiente a la estructura del discurso, y por esto mismo se suelen englobar bajo este epígrafe de *sistemas basados en la teoría del foco del discurso*, tal y como por ejemplo Ersan y Alkan nos definen en su trabajo [51]. Ellos cuentan que los investigadores que trabajan en este tipo de sistemas intentan modelizar la compleja estructura del discurso para que la anáfora, aceptada como un fenómeno a nivel del discurso, se resuelva utilizando esta estructura.

Esta *teoría del foco del discurso* está basada intuitivamente en la siguiente idea:

Los participantes de un discurso coherente centran su atención sobre ciertas entidades del mismo. Algunas entidades se destacan sobre las demás conforme se desarrolla el discurso, por lo que ciertas expresiones anafóricas se utilizan como técnica para referirse a esas entidades. Para que se puedan resolver, el antecedente debe estar en la consciencia del oyente; si esto no fuese así, el discurso estaría mal formado desde el punto de vista del oyente.

Según esta afirmación y suponiendo al discurso bien formado, si conseguimos obtener un modelo completo del discurso, entonces conseguiremos resolver estos tipos de anáfora. Por lo tanto, esta teoría se basa en este modelo del discurso y ha sido desarrollada inicialmente por Grosz en [71][70], quien construye un modelo de focalización basado en tres mecanismos:

- ◆ Representación del *foco explícito*. Se utiliza para localizar los elementos de mayor relieve en el discurso, los cuales quedarán almacenados en los *espacios focales*.
- ◆ Representación del *foco implícito*. Contiene información sobre conceptos asociados con el elemento que está focalizado en ese momento.
- ◆ *Cambio de foco*. Su función es actualizar los elementos contenidos en los espacios focales a medida que avanza el discurso, es decir, el foco del discurso va a ser dinámico. Estos cambios se producen fundamentalmente de dos modos: explícita e implícitamente. El primer modo sucede cuando aparecen expresiones que nos avisan explícitamente del cambio de foco, tales como *hablemos sobre...*. El segundo modo corresponde a la propia estructura de la frase o del texto en general, y es su propia información la que nos indica el cambio de foco.

Estos mecanismos de focalización limitan la búsqueda a las entidades relacionadas con elementos focalizados, dejando en segundo término otras entidades menos accesibles. Esta teoría se aplica fundamentalmente a la resolución de referencias pronominales.

En [154][152][153] Sidner modifica la teoría del foco esbozada por Grosz, al añadir las nociones de *foco discursivo* y *foco agente* que corresponden a entidades mencionadas en el enunciado que son focalizadas debido a la estructura sintáctica y a las relaciones temáticas de la oración. En estos trabajos, trata con referencias pronominales y sintagmas nominales definidos (sintagmas nominales precedidos por el artículo definido *the*, *this* o *that*), proponiendo diferentes algoritmos para cada tipo de anáfora. Las expresiones anafóricas se tratan de izquierda a derecha en la oración.

Por ejemplo, en *Juan sacó el perro a pasear*, el sintagma nominal *Juan* es el foco agente y *el perro* el foco discursivo. Sidner propone el siguiente algoritmo para la interpretación de la anáfora pronominal:

- 1) Si el pronombre está en una posición temática que no es la de agente, seleccionamos el foco discursivo como antecedente.
- 2) Si el pronombre está en una posición de agente, se tomará como antecedente el foco agente.

Podemos observar un ejemplo de la aplicación de este algoritmo sobre la siguiente frase: *Pedro y Juan compraron un libro. Ellos suelen comprar uno cada semana.* Aquí *Pedro y Juan* son el foco agente, mientras que *un libro* es el foco discursivo. De este modo *ellos* (posición de agente) referenciará a *Pedro y Juan*. Igualmente, relacionaremos *uno* con el foco discursivo *un libro*.

Sidner reconoce la necesidad de aplicar restricciones sintácticas, semánticas y pragmáticas que puedan impedir la selección de un antecedente aunque la regla lo haya identificado como tal. Por ejemplo, en: *Pedro y Juan compraron un libro. Después, ellos se tomaron varios helados. Ellos pensaron que éstos sabían realmente bien,* aquí el pronombre *éstos* no puede relacionarse con *un libro* porque las restricciones sintácticas y semánticas lo impiden. Se selecciona entonces el siguiente foco discursivo, esto es, *varios helados*. De este modo, la entidad escogida como antecedente según la teoría del foco del discurso también ha de superar filtros semánticos y sintácticos, y no ha de provocar ninguna contradicción en el mecanismo de inferencia. Los filtros sintácticos incluyen la concordancia morfológica y restricciones c-dominio. El mecanismo de inferencia lo aplica en ejemplos como el siguiente: *Llevé a mi perro al veterinario ayer. Él le mordió en la mano,* en el que elige al *perro* como antecedente del pronombre *él* puesto que los perros tienen mayor tendencia que las personas a morder, además de que no tienen *manos*.

En la primera frase del discurso, el algoritmo de Sidner intenta “adivinar” el posible foco del discurso, el cual lo ratificarán las oraciones posteriores. Para ello, utiliza una serie de reglas, como son el tipo de oración (por ejemplo, en el caso que se trate de una oración del tipo *is-a* o *there-is*, entonces escogerá el sujeto) o las relaciones temáticas del verbo. En las frases posteriores por cada expresión anafórica encontrada le aplica las restricciones sintácticas y semánticas, y los antecedentes resultantes los envía al mecanismo de inferencia, el cual confirmará o rechazará estos antecedentes.

En [74] Grosz, Joshi y Weinstein presentan un mecanismo de resolución de la anáfora que contrasta en algunos aspectos con el algoritmo de Sidner. La regla básica de interpretación anafórica está basada en un proceso de identificación de lo que denominan *centros focales* que vienen a sustituir a las nociones de foco discursivo y foco agente. Cada oración tendrá un centro focal que mira hacia atrás (*backward centre* o *Cb*) que une el enunciado al discurso precedente y supone la entidad focalizada en la oración. Y también cada oración dispondrá de un conjunto de centros focales que miran hacia delante (*forward centre* o *Cf*) que proporcionan un conjunto de entidades a las que pueden hacer referencia los siguientes enunciados. Este conjunto se encuentra ordenado por las propiedades gramaticales que se consideran importantes para obtener el grado de focalización en la frase. Un elemento del conjunto *Cf* puede convertirse posteriormente en el discurso en un *Cb*. El proceso de identificación de antecedentes, al que denominan *centering*, consiste en comprobar si el centro del enunciado que se está procesando es el mismo que el centro del enunciado anterior, en cuyo caso se podría haber empleado un pronombre.

En [72] Grosz y Sidner consiguen unir todas las teorías desarrolladas anteriormente, gracias a la definición de unos principios generales sobre la estructura del discurso, los cuales indican que la estructura básica del discurso consta de tres componentes:

- ◆ Estructura lingüística de la oración. El análisis lingüístico de la oración ayuda a localizar ciertos marcadores lingüísticos o *cue phrases* a partir de los cuales se produce un cambio de foco en el discurso.
- ◆ Estructura de intenciones. Determina la finalidad del discurso (*discourse purpose*). Todo discurso tiene una finalidad propia que lo diferencia de otros discursos y junto a ésta, existen una serie de *finalidades secundarias* que se distribuyen a lo largo del discurso y contribuyen al cumplimiento de la finalidad principal.
- ◆ Estructura de atenciones. Es una abstracción del foco de atención de los participantes en el discurso a medida que éste se desarrolla. Como podemos suponer, se trata de una estructura dinámica que cambia y se actualiza durante el transcurso del discurso.

Con estos tres elementos construyen la estructura discursiva y la organizan en espacios focales y pilas donde se almacena la información. De este modo se restringe la búsqueda de antecedentes a los que estén dentro de los espacios focales.

El aspecto más importante de esta teoría es el de la ordenación de las entidades de los centros focales que miran hacia delante o Cf. Estas entidades se ordenan por criterios sintácticos, semánticos y léxicos. Los defensores de esta teoría afirman que ésta trabaja correctamente para cualquier lenguaje siempre que se aplique una correcta ordenación del conjunto Cf. Por ejemplo, en [126][85][164] se utiliza el siguiente criterio de ordenación de la lista de entidades Cf para el japonés: *tópico > empatía > sujeto > objeto2 > objeto > otros*, definiendo la *empatía* como la propiedad gramatical que indica la posición del hablante a la hora de describir la situación. En [18][73] se considera el siguiente criterio de ordenación de la lista Cf como el más adecuado para los lenguajes occidentales: *sujeto > objeto1 > objeto2 > sintagmas preposicionales*.

En la Figura 18 se modelizan los cambios de centro focal en función de los centros focales de una oración O_n y los de su oración previa O_{n-1} . Para ello se utiliza el concepto de *centro preferido* o C_p que consiste en el primer elemento de la lista Cf, y representará la predicción sobre el Cb (centro focal que mira hacia atrás) de la próxima oración.

	$Cb(O_n) = Cb(O_{n-1})$	$Cb(O_n) \neq Cb(O_{n-1})$
$Cb(O_n) = C_p(O_n)$	<i>Continuar</i>	<i>Cambio ligero</i>
$Cb(O_n) \neq C_p(O_n)$	<i>Retener</i>	<i>Cambio radical</i>

Figura 18. Cambios de foco del discurso.

Estos cambios de foco están guiados por las siguientes reglas:

- ◆ Si en una oración O_n hay un sólo pronombre, y este pronombre referencia algún elemento de $C_f(O_{n-1})$, entonces éste elemento constituirá el $C_b(O_n)$.
- ◆ Si en O_n no hay ningún pronombre o más de uno, el $C_b(O_n)$ será:

- ◇ $Cb(O_{n-1})$ si $Cb(O_{n-1})$ aparece en O_n .
- ◇ En caso contrario, será el primer elemento de $Cf(O_{n-1})$ que aparece en O_n .

El orden habitual de preferencia en cuanto al cambio de foco sería el siguiente: *continuar, retener, cambio ligero y cambio radical*. Veamos un ejemplo que ayudará a entender estos cambios de foco:

- ◆ Inicialmente el texto comienza por la frase: *Juan es un buen chico*, para la cual tendremos $Cb = \emptyset$ puesto que se trata de la primera oración del discurso, y como $Cf = [Juan]$.
- ◆ Con la siguiente frase: *Él se encontró ayer con María*, tendremos $Cb = Juan$ y $Cf = [Juan > María]$.
- ◆ Consideremos a continuación las siguientes posibilidades:
 - ◇ En caso que viniese después la frase: *A él le gusta ella*, quedaría $Cb = Juan$ y $Cf = [Juan > María]$, por lo que entraríamos en el caso de la Figura 18 denominado *continuar*.
 - ◇ En caso que viniese: *A ella le gusta él*, tendríamos $Cb = Juan$ y $Cf = [María > Juan]$, entrando dentro del caso denominado *retener*.
 - ◇ En el caso de: *Ella estaba con Lucía*, tendríamos $Cb = María$ y $Cf = [María > Lucía]$, es decir, dentro del caso *cambio ligero*.
 - ◇ En caso de: *Lucía estaba con ella*, tendríamos $Cb = María$ y $Cf = [Lucía > María]$, dentro del *cambio radical*.

El algoritmo desarrollado por Brennan, Friedman y Pollard en [18] construye y actualiza la estructura discursiva al mismo tiempo que ayuda a localizar los antecedentes anafóricos. Asimismo añade algunas extensiones al trabajo de Grosz, Joshi y Weinstein [74] para tratar mayor número de casos, por ejemplo construye una nueva regla que hace posible el cambio de foco discursivo en más ocasiones. Llevan sus ideas a la práctica con el desarrollo de un sistema que sirve de interfaz a una base de datos. Para ello hay dos procesadores, uno semántico que construye representaciones de las oraciones anotadas con información sintáctica, información sobre concordancia e información sobre su función gramatical que pasa al otro procesador: el pragmático. Su algoritmo se puede resumir en tres fases:

- ◆ En la primera fase se localizan todas las expresiones anafóricas que se encuentren en la oración y se ordenan según su posición sintáctica con el sujeto de la oración en primer lugar. Con esta ordenación se establecen preferencias en la selección de candidatos. Además, se construyen todas las combinaciones posibles del centro focal que mira hacia atrás (Cb) y el que mira adelante (Cf).
- ◆ En la segunda fase se aplican una serie de filtros que eliminan como posibles antecedentes todas aquellas combinaciones de Cb y Cf que contradigan las reglas de indexación entre antecedente y expresión anafórica.
- ◆ Y finalmente en la tercera, se clasifica cada par (Cb , Cf) en función del tipo de transición que se da en el paso de una oración a otra. Así, por ejemplo, si el foco del discurso continúa siendo el mismo, los pronombres que aparezcan tendrán como antecedente este foco discursivo. Si por el contrario se detecta un cambio en el foco del discurso, tendrá preferencia como candidato a antecedente aquel par (Cb , Cf) que señale precisamente ese cambio y por tanto, las expresiones anafóricas que se hayan empleado considerarán el nuevo foco del discurso como antecedente preferido.

Por ejemplo, en el texto: *Brennan drives an Alfa Romero. She drives too fast. Friedman races her on weekends. She often beats her*, en la primera oración el *Cb* y *Cf* coinciden en *Brennan*. En la segunda oración el *Cb* y *Cf* siguen siendo los mismos por lo que el foco del discurso sigue siendo *Brennan*. Como no se ha producido ningún cambio en el foco, el pronombre *she* tiene como antecedente la entidad *Brennan*. En la tercera, el *Cb* es *Brennan* pero el *Cf* cambia a *Friedman*, esto es, hay un cambio de foco que influye en la selección del antecedente de *she* en la cuarta oración y hace que *Friedman* desplace a *Brennan*.

El inconveniente de este sistema es que no tiene en cuenta la incorporación de otras fuentes de conocimiento tales como la información semántica, el conocimiento del mundo o restricciones sintácticas que podrían rechazar el antecedente preferido por el foco discursivo.

En [28][27] Carter desarrolla el sistema *SPAR (Shallow Processing Anaphor Resolver)* con el deseo de comprobar la hipótesis de que no es necesario efectuar un tratamiento demasiado profundo del lenguaje para obtener resultados correctos en la resolución de ambigüedades. Carter afirma que incluso es posible limitar la cantidad de conocimiento del mundo necesaria para tratar computacionalmente el lenguaje, siempre que la información estrictamente lingüística esté suficientemente fundada. En este sistema los candidatos a antecedente son seleccionados tras la aplicación de las reglas contenidas en el módulo de gestión del discurso. Este sistema se compone de los siguientes módulos: sintáctico, semántico, módulo con reglas de gestión y actualización de la estructura del foco del discurso (módulo dominante) y un módulo que aplica reglas de inferencia sobre el mundo. El algoritmo utilizado se puede sintetizar en los siguientes pasos:

- ◆ Aplicación de los módulos sintáctico y semántico.
- ◆ Aplicación del módulo de gestión discursiva basado en reglas sobre la focalización de las entidades discursivas. Éste módulo está basado en las reglas de Sidner [153] pero añade un nuevo concepto: *la preferencia débil*. Este concepto se utilizará para determinar situaciones en las que no se tenga clara la preferencia, en cuyo caso el resto de módulos del SPAR tomarán la decisión final sobre qué candidato es el correcto. Un ejemplo sería la diferente selección del antecedente en función del verbo que se utilice en la segunda oración: *The monkey picked a banana. The elephant ate it.*, o para el verbo *attacked*: *The monkey_i picked a banana. The elephant attacked it_i*. De este modo, si se hubiese permitido una preferencia fuerte sobre uno de los dos candidatos, *monkey* o *banana*, siempre existiría la posibilidad de que el candidato propuesto fuera el erróneo.
- ◆ Aplicación de reglas de concordancia sintáctica y restricciones semánticas
- ◆ Aplicación de reglas de restricción sintáctica para la localización del antecedente de un pronombre.
- ◆ Reglas de inferencia que se aplicarán sólo cuando todavía existen relaciones anafóricas ambiguas tras la fase anterior.
- ◆ Aplicación de reglas heurísticas si todavía hay ambigüedad.

Los resultados que obtiene son satisfactorios ya que Carter obtiene un porcentaje del 93% en la identificación de los antecedentes correspondientes a 242 pronombres.

En [100] Luperfoy construye una representación del discurso en tres capas y la aplica a un interfaz de diálogos en lenguaje natural como parte del proyecto *Human Interface Tool Suite (HITS)* del *MCC Human Interface Laboratory*. Sus aplicaciones son un editor del conocimiento para la base de conocimiento *Cyc* [98], un editor de iconos para el diseño de pantallas para máquinas fotocopadoras y una herramienta de recuperación de información. En su sistema Luperfoy trata con sintagmas nominales definidos.

Las tres capas de las que habla son la capa lingüística, el modelo del discurso y la base de conocimiento. En la primera capa se introduce un objeto lingüístico por cada expresión anafórica y posible antecedente que aparezca en el texto. En la segunda se crea otro objeto por cada concepto que se desarrolle en el discurso. Y finalmente, en la tercera capa, la base de conocimiento, representa el sistema de conocimiento de una persona que participe en el diálogo. Luperfoy considera muy importante la separación entre la segunda y tercera capa para distinguir entre la comprensión del discurso y los conocimientos previos al propio discurso. Es decir, en algunas ocasiones aparece un objeto en el discurso que no es familiar al oyente, por lo que momentáneamente no se puede enlazar en la base de conocimiento. En estos casos se crearía un objeto del discurso sin especificar, quedando a la espera de que el discurso se desarrolle y nos ofrezca nueva información que nos permita clarificar su enlace en la base de conocimiento.

En [76] Hardt propone un interesante sistema basado en esta teoría del foco del discurso para tratar con las denominadas identificaciones descuidadas (*sloppy identity*) producidas en la elipsis verbal y para tratar con los pronombres del tipo “cheque” (*paycheck pronouns* conocidos así por el ejemplo de Karttunen: *The man who gave [his paycheck], to his wife was wiser than the one who gave it, to his mistress*). En estos casos la expresión anafórica y su antecedente no tienen idéntico significado. Por ejemplo, en: *Pedro quiere a su gato. Juan también*, al resolver la elipsis verbal nos quedaría en la segunda frase: *Juan también quiere a su gato*, con lo que la referencia producida por el posesivo *su* habría que resolverla nuevamente (¿Juan quiere a su gato, o al gato de Pedro?). Basándose en el sistema desarrollado en [123] por Muskens, Hardt se propone plantear referencias al foco del discurso de manera explícita, para que después de resolver la elipsis el objeto elidido siga haciendo referencia al nuevo foco del discurso de la oración. Es decir, sobre el anterior ejemplo, en la primera frase *Pedro* es el foco del discurso, pero en la segunda frase *Juan* pasa a ser el nuevo foco. De este modo, la entidad *su gato* en la primera frase referenciará a su foco (*Pedro*), y al recuperarse en la segunda frase pasará a referenciar al nuevo foco de la misma (*Juan*), con lo que no habría que repetir el proceso de resolución de la anáfora.

Una de las desventajas de la teoría del foco del discurso es la comentada en [126] por Manabu y Kouji, según la cual esta teoría no ha sido totalmente probada en textos reales no restringidos, sino únicamente en sucesivas oraciones simples que habitualmente contienen un sólo verbo. Manabu y Kouji proponen precisamente dos métodos de manejar oraciones complejas en las que haya más de un verbo: *Oración1 Conjunción Oración2*, donde *Oración1* y *Oración2* han de ser oraciones simples. El primer método de tratar estas oraciones sería el considerar la conjunción como si fuese un punto de separación entre frases (con Cb y Cf independientes entre sí). Y el segundo método consistiría en tratar toda la oración como un conjunto, es decir, no considerarlas como frases independientes (con un único Cb y Cf). Ellos comparan ambos métodos concluyendo que el segundo reporta ciertos problemas con lo que consideran preferible tomar las oraciones coordinadas como oraciones simples. Esta misma solución es seguida en otros trabajos como por ejemplo los de Di Eugenio [49] y Kameyama [86].

En [11] Azzam también intenta solucionar la anterior limitación de la teoría del foco del discurso sobre oraciones complejas, proponiendo un algoritmo para la resolución de la anáfora pronominal intrasentencial que ocurre en oraciones que contienen cláusulas subordinadas. Este algoritmo toma como base los trabajos de Sidner [154][152][153], intentando expandirlos para el tratamiento de este tipo de oraciones. Según Azzam, el algoritmo propuesto por Sidner debe ser mejorado en la anáfora intrasentencial, ya que éste sólo trabaja con los antecedentes de las oraciones anteriores, sin tener en cuenta los que aparecen en la oración actual (opinión también compartida por Carter en [28]). Otro problema que encuentra en el algoritmo de Sidner es cuando la primera oración del texto presenta una expresión anafórica (normalmente en una oración subordinada), con lo que el algoritmo que Sidner propone para seleccionar ese foco inicial no funciona correctamente. En la solución propuesta por Azzam, se considera que las oraciones que contienen cláusulas subordinadas incluyen varias situaciones o hechos elementales. Por ejemplo, la frase: *[Tres de las empresas más importantes del mundo], dijeron que ellas, están formando una plataforma común*, es dividida en dos hechos, por una parte, el hecho de que alguien está diciendo “algo”, y por otra parte lo que se está diciendo. Cada uno de estos hechos será el marco sobre el que se aplicará la teoría de foco del discurso propuesta por Azzam, igual que si se tratase de sucesivas oraciones simples. De este modo se evita el problema ocasionado por las expresiones anafóricas encontradas en una cláusula subordinada de la primera oración. Para los pronombres posesivos, recíprocos y reflexivos que aparezcan en el primer hecho del texto prevé un tratamiento especial. Por ejemplo, en: *Las empresas son por sí mismas...*, en primer lugar intenta resolver estas referencias antes de aplicar el algoritmo de adivinar el foco de esta primera oración, suponiendo que no habrá gran ambigüedad en esta etapa del discurso. El proceso de división de la frase en hechos se lleva a cabo por medio de un analizador conceptual implementado en [10] y [9] por el mismo Azzam.

En [155] y [156] Strube también extiende esta teoría del foco del discurso para la resolución de la anáfora intrasentencial y para el manejo de oraciones complejas en alemán. Aquí supone para las oraciones complejas un Cb y un Cf por cada cláusula, considerando inadecuada para el alemán la ordenación de la lista de entidades Cf basado en los criterios de roles temáticos, tal y como Suri y McCoy proponen en [159], donde además se plantea una estrategia mixta para el tratamiento de las oraciones complejas, según la cual en determinadas ocasiones se preferirán antecedentes intraoracionales y en otras ocasiones interoracionales. En lugar de ello, considerará más adecuada una ordenación en función de la información estructural de la oración.

Tal y como Di Eugenio cuenta en [49], otro tema no resuelto todavía en esta teoría del foco del discurso es el del tratamiento de los posesivos y el determinar cómo afectan éstos a los centros focales. Di Eugenio llega a la conclusión de que Cb no parece ser afectado por estos posesivos, mientras que Cf sí que necesita ser modificado. Según Di Eugenio, un sintagma nominal de tipo posesivo se refiere a dos entidades: el poseedor (P_{or}) y lo poseído (P_{do}). Lo poseído se corresponde con el sintagma nominal completo, por lo que su posición dentro de Cf es determinado por su función gramatical. Con respecto al poseedor, considera (tras pruebas heurísticas) que la posición que le debe corresponder es la inmediatamente anterior al objeto poseído si éste es inanimado, y como siguiente si es animado.

También en [51] Ersan y Akman muestran la ineficacia de esta teoría del foco del discurso ante los casos de anáfora que se suceden en estructuras paralelas. Por ejemplo, en el siguiente texto: *The green Whitierleaf is most commonly found near the wild rose. The wild violet is found near it too*, el pronombre *it* se refiere al sintagma nominal *the wild rose*

al utilizar la noción de paralelismo entre estructuras sintácticas, sin embargo, al aplicar la teoría del foco, ésta elegiría como antecedente a *Whitierleaf*.

Para finalizar este apartado, comentar el trabajo de Mitkov en [109][113] en el que intenta combinar los métodos lingüísticos clásicos con la teoría del foco del discurso. Para ello plantea la posibilidad de aplicar la información morfológica, lingüística y semántica del modo que hemos estudiado anteriormente, junto con esta teoría aplicada como preferencia en caso de ambigüedad a la hora de seleccionar un antecedente. Además, Mitkov realiza la aportación de aplicar un método probabilístico a la selección del foco del discurso en la primera frase del texto. Este método lo lleva a cabo en dos pasos: selección de una serie de reglas heurísticas obtenidas de un estudio previo del dominio del texto, las cuales nos indican modelos seguidos por el foco del discurso durante el texto analizado, y como segundo paso, el cálculo de la probabilidad de un determinado segmento a ser el foco en la primera frase en función de las anteriores reglas.

3.2.3 Sistemas alternativos

En la subsección anterior hemos revisado los sistemas integrados en los que no se utilizaba información obtenida a partir de corpus. A continuación vamos a introducir una serie de sistemas que emplean fundamentalmente información de este tipo obtenida del estudio del corpus sobre el que posteriormente se evaluará el sistema propuesto.

Uno de estos sistemas alternativos es el propuesto por Dagan e Itai en [45]. Ellos desarrollan una aproximación estadística para la desambiguación de pronombres. Para ello aplicaron su método sobre ocurrencias del pronombre *it* que se encuentran en un grupo de oraciones seleccionadas aleatoriamente. La información que utilizaban para la resolución era únicamente la de las plantillas obtenidas del análisis previo del corpus. Estas plantillas consisten en patrones asociados a cada tipo de expresión anafórica con la posición que ocupaba su antecedente. Después del análisis previo del corpus, estos patrones se aplican a las nuevas expresiones anafóricas que se encuentren, seleccionando el antecedente que se encuentre en la misma posición que muestra la plantilla. La evaluación del método obtuvo un 87% de éxito, eso sí, eliminando manualmente los casos de pronombres *it* no anafóricos.

Un ejemplo de la aplicación de este método podría ser el siguiente: *They know full well that the companies held tax money, aside for collection later on the basis that the government, said it, was going to collect it.*. Tenemos dos ocurrencias del pronombre *it*: una funcionando como sujeto y otra como objeto, ambas del verbo *collect*. Existen tres posibles antecedentes: *money*, *collection* y *government*, y después del análisis del corpus se dispone de las plantillas mostradas en la Figura 19 que nos miden el número de ocurrencias de cada antecedente funcionando como sujeto o como objeto del verbo *collect*. Según estas estadísticas concluyen que *government* es el antecedente del primer pronombre *it*, y que *money* es el antecedente del segundo.

<i>Sujeto</i>	<i>Verbo</i>	<i>Número de ocurrencias en el texto</i>
<i>collection</i>	<i>collect</i>	0
<i>money</i>	<i>collect</i>	5
<i>government</i>	<i>collect</i>	198
<i>Verbo</i>	<i>Objeto</i>	<i>Número de ocurrencias en el texto</i>
<i>collect</i>	<i>collection</i>	0
<i>collect</i>	<i>money</i>	149
<i>collect</i>	<i>government</i>	0

Figura 19. Plantillas obtenidas del análisis del corpus según Dagan e Itai.

En [125] Nasukawa describe una aproximación muy simple que utiliza información interoracional extraída de un texto fuente para mejorar la precisión de su sistema de resolución de la anáfora pronominal. El principal mérito del trabajo de Nasukawa es el que no necesita conocimiento del mundo exterior ni realiza un análisis sintáctico, morfológico o semántico. Él sugiere que las plantillas que definen la relación entre modificadores y modificados pueden resultar útiles para la detección del antecedente correcto de un pronombre. Por ejemplo, de la oración *He moved his residence* extrae la plantilla por la que *residence* puede ser objeto del verbo *move*. Del mismo modo también considera que la *reiteración* puede considerarse como otro indicador de la presencia del antecedente, expresada esta reiteración por la sucesión de sintagmas nominales con la misma palabra núcleo. Igualmente plantea la incorporación de otras reglas heurísticas dependientes del dominio en la resolución, como por ejemplo la preferencia por el sintagma nominal con la función de sujeto en lugar de los que ocupen la función de objeto.

Nasukawa probó su sistema sobre 1904 oraciones consecutivas de ocho capítulos de dos manuales diferentes de informática, que contenían 112 pronombres en tercera persona, obteniendo una precisión del 93,8%.

Connolly, Burger y Day describen en [41] una aproximación al tratamiento de la anáfora basada en el aprendizaje computacional. Ellos enfocan la anáfora como un problema de clasificación, es decir, dada una expresión anafórica y dos posibles antecedentes reducen el problema a la elección entre una de esas dos clases (cada uno de los dos antecedentes). Una vez elegida una de las clases se compara este antecedente con otro, repitiendo el proceso sucesivamente hasta que no quede más que una solución posible. Para elegir entre cada pareja de antecedentes se codifica la información de forma discreta como vectores de atributos (de modo similar al trabajo de Rico en [143]), donde cada atributo describe propiedades de la expresión anafórica, los antecedentes y la relación entre ellos tres.

En [146] Rocha describe una aproximación al tratamiento de la anáfora en diálogos, utilizando textos de conversaciones en inglés y portugués. Para ello utiliza la información probabilística obtenida de la anotación manual de un corpus de conversaciones de este estilo (sobre unos 3000 casos de anáfora para cada lenguaje). Esta información se almacena en un modelo basado en árboles probabilísticos. Según el estudio que realiza del corpus, se anotarán los casos de anáfora de acuerdo a cuatro propiedades. La anotación se realizará manualmente por analistas con el objetivo de marcar las entidades importantes en el discurso: tópico del discurso tanto global como local, expresiones anafóricas y sus antecedentes. Para cada tipo de expresión anafórica se obtendrá su árbol probabilístico.

Cada expresión anafórica se anota en el corpus de acuerdo a las siguientes cuatro propiedades:

- ◆ *Tipo de expresión anafórica*, distinguiendo a su vez entre las expresiones formadas por palabras (pronombres con función de sujeto, verbos anafóricos o adverbios), por sintagmas (sintagmas nominales definidos) y casos de anáfora de tipo "one".
- ◆ *Tipo de antecedente*, según sea éste explícito, implícito (no aparece en el texto) o no referencial (casos de pronombres no anafóricos como el pronombre *it* cuando actúa como sujeto de una oración impersonal, por ejemplo en la frase: *It's raining*). En el caso que el antecedente aparezca explícitamente en el texto, se le añade a esta propiedad el número de antecedente según una lista de antecedentes posibles aparecidos en el texto.
- ◆ *Grado de topicalidad del antecedente*. En caso que sea el tópico del segmento en el que aparece se le clasifica como *elemento temático*. Si además éste tiene relación con los participantes del diálogo o con algún elemento del tópico global del discurso, entonces se le clasifica como *elemento temático del discurso*.
- ◆ *Información psicolingüística*, en la que se intenta codificar observaciones encontradas en el análisis del corpus que demuestran que para expresiones anafóricas de la misma categoría se pueden necesitar diferentes estrategias para su resolución. Estas estrategias las codificarán en cuatro grupos: basadas en procesos léxicos (utilizando conocimiento semántico asociado a la expresión anafórica, conocimiento del mundo y reiteración léxica), basadas en el discurso, según la posición (nociones de paralelismo entre expresión anafórica y su antecedente) y según procesos sintácticos.

Para cada una de esas cuatro propiedades se obtendrán datos probabilísticos de la distribución de cada una de sus subcategorías en los casos de anáfora estudiados. Como ejemplo se muestra en la Figura 20, la distribución de los casos de anáfora en función del tipo de expresión anafórica. La combinación de los datos probabilísticos de las cuatro propiedades se realizará en árboles probabilísticos cuyas raíces serán el tipo de expresión anafórica.

	<i>Frecuencia</i>	<i>Porcentaje</i>
<i>Pronombres</i>	1.579	51,1
<i>Verbos y adverbios</i>	318	10,3
<i>Sintagmas nominales</i>	1.193	38,6
<i>Total</i>	3.090	100

Figura 20. Frecuencias de expresiones anafóricas encontradas en el corpus.

La aplicación de esta información estadística irá orientada principalmente a la obtención de esquemas de resolución de expresiones anafóricas según los criterios que hayan alcanzado una mayor probabilidad. Por ejemplo, en la Figura 21 se muestra la regla obtenida para la resolución del pronombre *it* cuando funciona como objeto. En primer lugar se aplica la estrategia que haya alcanzado la mayor probabilidad tras el estudio del corpus. En caso que no se encuentren antecedentes tras aplicarla, se considera que se trata de un

caso especial, por lo que se recurre a estrategias alternativas, de nuevo ordenadas según la probabilidad de cada una de ellas.

Obtener antecedentes según la estrategia de posiciones
Si no se obtiene ningún antecedente
 Seleccionar el primer antecedente posible
 Si no existe ese primer antecedente
 Aplicar procesos léxicos

Figura 21. Regla para la resolución del pronombre “it” cuando funciona como objeto.

Tal y como comenta Rocha, este sistema todavía no está finalizado por lo que no se dispone de medidas de su rendimiento. Además, sólo se ha realizado el estudio sobre el corpus en inglés, con lo que la consiguiente transformación y obtención de las reglas de resolución de los distintos tipos de anáfora para el portugués está todavía pendiente (por ejemplo en portugués no existe la anáfora de tipo “one”). Asimismo, Rocha destaca que actualmente se realiza una anotación totalmente manual, por lo que considera necesario el desarrollo del campo de la anotación automática de corpus para su empleo en aplicaciones reales.

También dentro de este tipo de sistemas alternativos merece la pena recordar los trabajos anteriormente comentados de Mitkov [118][116] en los que utiliza una serie de *indicadores de antecedente* que consisten en reglas heurísticas de preferencia obtenidas del estudio del corpus sobre el que se aplicarán posteriormente dichas reglas. En definitiva, se trata de información obtenida a partir del estudio del corpus, o sea, reglas dependientes del dominio sobre el que se trabajará posteriormente. Por el momento dichas reglas se obtienen de manera manual e indican comportamientos habituales que siguen las expresiones anafóricas para localizar sus antecedentes.

4. Formalismo Gramatical *SUG*

En este capítulo expondremos un nuevo formalismo gramatical: *gramática de unificación de huecos (SUG, Slot Unification Grammar)*, y un sistema modular de procesamiento del lenguaje natural al que hemos llamado *SUP (Slot Unification Parser)*. *SUG* es un nuevo formalismo desarrollado por nosotros como una extensión de las *gramáticas de cláusulas definidas (DCG, Definite Clause Grammar)*. Éste será el formalismo que utilizaremos para almacenar la información sintáctica necesaria para el algoritmo de resolución de la anáfora que proponemos en este trabajo. En cuanto a *SUP* constituye el sistema en el que integraremos nuestra propuesta de resolución de la anáfora, y que utiliza las *SUG* como formalismo gramatical.

Comenzaremos con una sección dedicada a los fundamentos teóricos de los formalismos gramaticales, continuando con la descripción de varios de ellos pertenecientes a la familia de formalismos basados en la unificación. Estas dos primeras secciones nos permitirán introducir las aportaciones que nos presenta el formalismo *SUG* respecto a los formalismos ya existentes actualmente. Y finalizaremos con la descripción del sistema *SUP*, en el que se integra el formalismo *SUG* y nuestro algoritmo de tratamiento de la anáfora.

4.1 Fundamentos teóricos

4.1.1 Concepto de “gramática”

Comencemos por definir el concepto de *gramática*. Shieber lo define en [150] como un lenguaje diseñado con el objetivo de describir a los propios lenguajes: el conjunto de oraciones que abarca, las propiedades estructurales de esas oraciones (su *sintaxis*) y el significado de esas oraciones (su *semántica*). De esta definición se desprende que con la

gramática se especifica por una parte la estructura sintáctica del lenguaje, y por otra parte la estructura semántica.

La mayoría de las representaciones sintácticas del lenguaje están basadas en la noción de *gramáticas incontextuales* (CFG, *Context Free Grammar*). Estas gramáticas están formadas por un conjunto finito de reglas con un único símbolo en su parte izquierda. Una CFG está formada por la cuádrupla $G=(NT, T, S, P)$, donde:

- ◆ NT es el conjunto finito de símbolos no terminales correspondientes a las categorías sintácticas.
- ◆ T es el conjunto finito de símbolos terminales correspondientes a las categorías léxicas.
- ◆ S es el símbolo inicial tal que $S \in NT$, a partir del cual mediante P pueden generarse todas las oraciones del lenguaje definido por la gramática: $L(G)$.
- ◆ P es el conjunto de *reglas de producción* de la forma $A \rightarrow W$, donde $A \in NT$ y $W \in (NT \cup T)^*$.

En las CFG la estructura de una oración se representa en términos de qué constituyentes son subpartes de otros constituyentes. Esta información se representa con frecuencia en forma de árbol tal y como se muestra en la Figura 22.

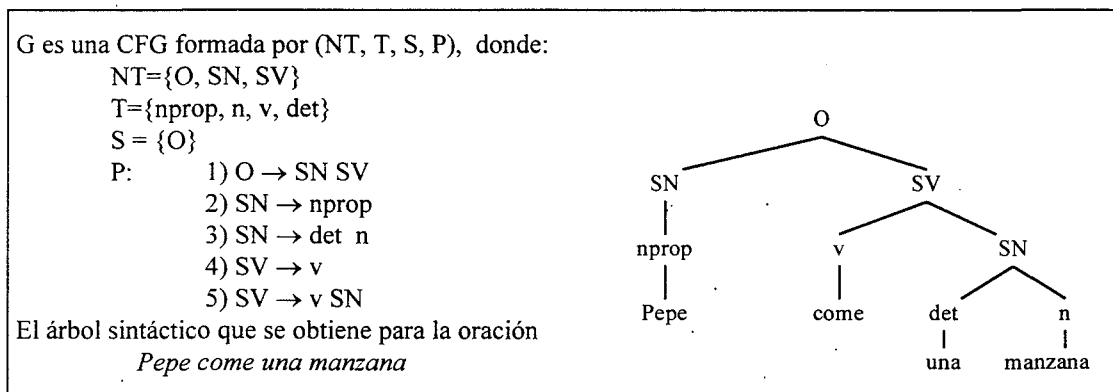


Figura 22. Ejemplo de una gramática incontextual.

Las CFG son una clase de gramáticas muy interesantes por dos razones: en primer lugar el formalismo es suficientemente poderoso para describir muchas de las estructuras del lenguaje natural y, en segundo lugar, son lo suficientemente restrictivas para que se puedan construir analizadores eficientes.

Aunque las CFG fueron utilizadas en los primeros sistemas de tratamiento del lenguaje natural, éstas son inadecuadas para modelizarlo. Se puede evidenciar dicha inadecuación citando los siguientes ejemplos no modelizables por medio de estas gramáticas:

- ◆ El tratamiento de la concordancia en género, número y persona entre diferentes constituyentes, como por ejemplo entre el sujeto y el predicado de una oración: *Los niños de la Escuela fueron a la feria.*

- ◆ Tratamiento de constituyentes discontinuos: *¿Qué películas hay de aventuras?*, donde *de aventuras*, se refiere a *películas*.
- ◆ Asignar la misma estructura a oraciones diferentes pero relacionadas como tratamiento de pasivas, interrogativas, paráfrasis, etc.: *Pedro hizo un pastel* o *El pastel fue hecho por Pedro*.
- ◆ Dar diferente estructura a oraciones cuya semántica es distinta aunque aparentemente parezcan similares: *Pedro sube la maleta* y *Pedro sube la escalera*.

Estas inadecuaciones fueron tratadas por lingüistas y en especial por Chomsky en [36][32] a través de la *gramática transformacional (TG, Transformational Grammar)*. Básicamente consistía en añadir otro tipo de regla a las CFG generando también un árbol de análisis pero añadiéndole ciertas etiquetas, como por ejemplo la información correspondiente a la concordancia en número, género o persona. El conjunto de transformaciones aplicadas al árbol de análisis venía a solucionar las evidencias anteriores con el fin de poder transmitir la pluralidad a todos los sectores implicados del árbol y poder así reforzar las concordancias necesarias. De este modo aparecía la distinción entre *transformaciones obligatorias* y *opcionales*. Las primeras se encargaban de reforzar la concordancia entre constituyentes, y las segundas se utilizaban para captar las relaciones como por ejemplo de las oraciones activas y pasivas.

Aunque las TG sentaron las bases de un gran desarrollo lingüístico, desde un punto de vista computacional fueron mucho peor que las CFG. Esto se debe a que se iniciaron como *modelo generativo*, es decir, un modelo para decir cómo se produce una oración a partir de un símbolo inicial de la gramática, pero al invertir el modelo para analizar una oración se producía un auténtico desastre computacional, debido fundamentalmente a que las transformaciones actúan sobre árboles y no sobre cadenas de palabras.

En la literatura actual aparecen una gran diversidad de gramáticas basadas en las CFG siendo conveniente distinguir entre *formalismos gramaticales* y *teorías lingüísticas o gramaticales*. Según estudia Shieber en [150] la diferencia fundamental entre ambos es el distinto objetivo con el que son diseñados, lo cual hace que presenten distintas características o herramientas. Los formalismos gramaticales incluyen mecanismos generales para incrementar el poder expresivo de la gramática, mientras que las teorías lingüísticas tienden a incorporar mecanismos diseñados para un particular tipo de análisis lingüístico. Como ejemplos más representativos de formalismos encontramos las *gramáticas de unificación funcional (FUG, Functional Unification Grammar)* [90][89], las *gramáticas de cláusulas definidas (DCG, Definite Clause Grammar)* [132] y las *gramáticas de huecos (SG, Slot Grammar)* [105][103][104], y como ejemplos de teorías lingüísticas tenemos las *gramáticas léxico-funcional (LFG, Lexical-Functional Grammar)* [19], las *gramáticas de estructura sintagmática generalizada (GPSG, Generalized Phrase Structure Grammar)* [67] y las *gramáticas de estructura sintagmática dirigidas por el núcleo (HPSG, Head-Driven Phrase Structure Grammar)* [135].

La elección entre una gramática que se encuadre dentro de los formalismos o de las teorías es un hecho fundamental para el estudio que se pretenda realizar, puesto que determinará la influencia de los siguientes parámetros:

- ◆ El grado de precisión con el que se pueden exponer la descripción de los fenómenos lingüísticos.
- ◆ El nivel de expresividad del lenguaje formal de la gramática.

- ◆ Y la eficacia computacional de las posibles implementaciones de la gramática si es que existen, y en caso cierto, qué limitaciones computacionales arrastran.

Como ejemplo de esta influencia, podemos observar que el mantener un alto nivel expresivo se suele considerar como una ventaja para un formalismo, mientras que no lo es tanto para una teoría lingüística. Para algunas de estas teorías, como las GPSG y LFG, el poder expresivo de las mismas es un asunto a ser restringido en lugar de ser fomentado. Del mismo modo tiene mayor importancia la eficacia computacional para el caso de hablar de un formalismo que para una teoría lingüística.

En nuestro caso ya que pretendemos abordar fenómenos lingüísticos desde el punto de vista computacional, utilizaremos un formalismo gramatical en lugar de una teoría para especificar la estructura sintáctica del lenguaje. Más concretamente utilizaremos el formalismo gramatical desarrollado por nosotros *gramática de unificación de huecos (SUG)* [53][128][60] que supone una extensión del formalismo DCG.

4.1.2 Análisis sintáctico

En la subsección anterior hemos visto que la gramática supone un mecanismo de especificación de la estructura sintáctica del lenguaje, pero aparte de esta especificación hay que determinar cómo utilizamos esta información para determinar la estructura de una determinada oración de acuerdo a la gramática, proceso que se denomina *análisis sintáctico*.

Como confirmación de la división existente entre gramática y análisis sintáctico, podemos citar la definición de gramática según Allen [3] como la especificación formal de las estructuras permitidas por el lenguaje, diferenciándola de la técnica de análisis sintáctico que será el método utilizado en una oración para determinar su estructura de acuerdo a la gramática. Se confirma esa diferenciación, al destacar que para realizar un análisis sintáctico hay que plantearse cómo representar la información sintáctica (gramática) y cuál será la técnica de análisis a utilizar.

El análisis sintáctico constituye el núcleo de cualquier sistema de procesamiento del lenguaje natural. Su función consiste en determinar si una frase es gramaticalmente correcta y en proporcionar una estructura asociada a la frase que refleje sus relaciones sintácticas para ser utilizada en fases de análisis posteriores.

Aparecen diferentes técnicas de análisis como por ejemplo las *descendentes (top-down)* y *ascendentes (bottom-up)*.

Existen diferentes estrategias o técnicas de análisis:

- ◆ En *paralelo* o *secuencial*.
- ◆ *Ascendente* o *descendente*.

Las *estrategias en paralelo* siguen todas las secuencias de derivación a la vez, mientras que las *secuenciales* siguen una sóla. Las necesidades de espacio de las estrategias en paralelo son mayores ya que deben almacenar todos los análisis parciales.

En las estrategias *ascendentes* o *bottom-up* se comienza por las palabras de la oración que se intenta analizar que se unen formando constituyentes más complejos según se van aplicando las reglas cuyas condiciones lo permitan. Además si el análisis es en paralelo, entonces devuelve todas las interpretaciones posibles simultáneamente. Es decir, se aplicarán todas las reglas cuyas condiciones se cumplan, pudiendo llegar a producirse varios análisis para una misma cadena si no existen restricciones que lo impidan. Sin embargo, una estrategia secuencial devolverá un análisis tan pronto como lo encuentre, dando por terminado el proceso.

En las *descendentes* o *top-down* la secuencia de derivación comienza por la parte superior o símbolo inicial de la gramática hasta la oración objeto del análisis. Estas estrategias también pueden utilizar *algoritmos de análisis con retroceso* o *backtracking* los cuales cuando no pueden continuar por una determinada derivación durante el análisis de una oración, vuelven a un punto anterior del análisis para intentar una derivación alternativa. Una ventaja clara del análisis ascendente es que un análisis parcial se construirá sólo una vez aunque se utilice posteriormente como un constituyente de muchos análisis parciales diferentes. En contraste, el análisis descendente puede acabar re-expandiendo muchas veces un símbolo dado comenzando en la misma palabra si ese símbolo aparece en contextos diferentes.

4.1.3 Análisis semántico

Según vimos en la definición de gramática según Shieber [150] además de especificar las propiedades estructurales de las oraciones (su *sintaxis*) también se especifica el significado de esas oraciones (su *semántica*). En esta subsección se describe el modo de especificar este significado de las oraciones y el proceso de obtener la estructura semántica de la oración denominado *análisis* o *interpretación semántica*.

La representación del significado de la oración independientemente del contexto se llama *fórmula* o *forma lógica*. El proceso de emparejar una oración con su forma lógica se llama *análisis* o *interpretación semántica*. O sea, el proceso de análisis semántico consiste en la obtención del significado independiente del contexto en que aparece la frase por medio de un lenguaje formal que llamaremos fórmula o forma lógica. Posteriormente a la interpretación semántica, tal y como Allen expresa en [3], aparece otro proceso en el que se le añadiría a esta fórmula lógica la información referente al contexto. Este proceso final se le conoce como *análisis* o *interpretación contextual* y consiste en emparejar una fórmula lógica al lenguaje de representación del conocimiento.

El lenguaje formal que se suele utilizar habitualmente para la construcción de la fórmula lógica es el *Cálculo de Predicados de Primer Orden (CPPO)*, ya que dentro de su sencillez nos permite alcanzar los objetivos del análisis semántico, como por ejemplo su combinación con mecanismos de inferencia poderosos y simples que permiten el razonamiento acerca del conocimiento de forma fácil.

A continuación pasamos a desarrollar este lenguaje formal basado en el CPPO. El sentido de las palabras del lenguaje servirán como *átomos* o *constantes* de la representación. Estas constantes serán clasificadas en función de los tipos de cosas que describen. Por ejemplo, las constantes que describen objetos en el mundo incluyendo objetos abstractos

tales como eventos o situaciones son llamados *términos*. Las constantes que describen relaciones y propiedades son llamados *predicados*. Una *proposición* en el lenguaje está formado de un predicado seguido por un número apropiado de términos con sus argumentos: predicados con un único argumento se llamarán *predicados unarios*, predicados con dos argumentos serán *predicados binarios*, y finalmente los predicados con n argumentos constituirán los *predicados n-arios*. Podemos observar un ejemplo de diferentes clases de palabras o constantes en la Figura 23.

<i>Términos</i>	<i>Predicados unarios</i>	<i>Predicados n-arios</i>
<i>Nombres propios:</i> <i>Juan, Luis, ...</i>	<i>Nombres comunes:</i> <i>perro(X), árbol(X), ...</i>	<i>Verbos (donde n depende de los argumentos del verbo):</i> <i>correr(X,Y), cantar(X,Y,Z), ...</i>

Figura 23. Ejemplos de palabras en el lenguaje de CPPO.

Las proposiciones complejas se construirán usando una nueva clase de constantes llamadas *operadores lógicos*. El CPPO contiene operadores como la *disyunción* (\vee) o la *conjunción* (\wedge ó también $\&$) frecuentemente llamada *implicación*. Las conectivas en el lenguaje natural frecuentemente expresan relaciones complejas entre oraciones. Por ejemplo, la conjunción *y* puede corresponderse con el operador lógico $\&$, aunque también puede significar una secuencia temporal: *Fui a casa y tome una copa*, en la cual *ir a casa* precede a *tomar una copa*. Los operadores lógicos se usarán además para conectar proposiciones no explícitamente unión de oraciones.

Con proposiciones simples sólo definimos subconjuntos limitados, normalmente oraciones de un sólo verbo. Para oraciones más complejas, deberemos definir construcciones semánticas adicionales, como por ejemplo los *cuantificadores*. En el CPPO hay dos cuantificadores: *Todo* (\forall) y *Existe* (\exists), pero en cualquier idioma es necesario añadir más cuantificadores como por ejemplo: *cada*, *existenVarios*, *muchos*, *algún*, etc. La estructura habitual de un cuantificador es la formada por la secuencia: *cuantificador (variable, restricción de proposición, cuerpo de proposición)*, como en los ejemplos mostrados en la Figura 24, donde para el sintagma *un perro ladra* se representa mediante el cuantificador *existe* que se compone de la variable X , la restricción de proposición *perro (X)* y como cuerpo de proposición *ladrar (X)*.

<i>Oración</i>	<i>Fórmula lógica</i>
<i>Cada hombre pasea</i>	<i>cada (X, hombre (X), pasear (X))</i>
<i>Los perros ladran</i>	<i>existenVarios (X, perro (X), ladrar (X))</i>
<i>Un perro ladra</i>	<i>existe (X, perro (X), ladrar (X))</i>
<i>Todos los perros ladran</i>	<i>todo (X, perro (X), ladrar (X))</i>

Figura 24. Ejemplos de cuantificadores.

Una de las formas más complejas de ambigüedad en las fórmulas lógicas surge del proceso de establecer el ámbito entre sucesivos cuantificadores. Por ejemplo, la siguiente frase: *Cada niño quiere a un perro* es ambigua dependiendo del ámbito del cuantificador, lo

que redundará en diferentes lecturas de la frase. En primer lugar, podríamos interpretarlo como que existe un grupo de niños, y cada uno de ellos quiere al mismo perro, lo cual vendría dado por la fórmula lógica: *todo* (X , niño (X), existe (Y , perro (Y), querer (X , Y))). O también podríamos interpretarlo como que cada niño quiere a un perro diferente, que se representaría como: *existe* (Y , perro (Y), *todo* (X , niño (X), querer (X , Y))). El problema es que no hay un método independiente del contexto para resolver tal tópico, así la ambigüedad debe ser representada en la forma lógica de la oración, dejando su resolución para la posterior fase de interpretación contextual, es decir, sin resolver la precedencia de uno u otro cuantificador. Por ejemplo, en [3] Allen la representaría como: (*QUERER* <CADA b1 (NIÑO b1)> <UN d1 (PERRO d1)>. En [42] Covington la representaría como $(\forall x)$ (niño(x) \rightarrow $(\exists y)$ (perro (y) \wedge querer (x, y))) o en su equivalente en Prolog *cada* (X , niño (X), existe (Y , perro (Y), querer (X, Y))). En la Figura 25 se muestran varios ejemplos de fórmulas lógicas propuestas por Covington. En nuestra especificación de lenguaje de fórmula lógica que utilizaremos en el resto del trabajo hemos establecido la misma hipótesis de precedencia del cuantificador que el enfoque seguido por Covington [42] y Moreno [120], de tal manera que sencillamente estableceremos un orden secuencial de cuantificadores, según el cual anidaremos sucesivamente todos los cuantificadores que encontremos, colocando en el nivel más externo el primer cuantificador que aparezca en la frase. Así devolveríamos como fórmula lógica: *cada* (X , niño (X), existe (Y , perro (Y), querer (X , Y))).

Oración	Fórmula lógica	Representación en Prolog
<i>Fido barked</i>	$barked(Fido)$	$barked(Fido)$
<i>A dog barked</i>	$(\exists x)(dog(x) \wedge barked(x))$	$some(X, dog(X), barked(X))$
<i>Every dog barked</i>	$(\forall x)(dog(x) \Rightarrow barked(x))$	$all(X, dog(X), barked(X))$
<i>Fido chased a cat</i>	$(\exists x)(cat(x) \wedge chased(fido, x))$	$some(X, cat(X), chased(fido, X))$
<i>Fido chased every cat</i>	$(\forall x)(cat(x) \Rightarrow chased(fido, x))$	$all(X, cat(X), chased(fido, X))$
<i>A dog chased a cat</i>	$(\exists x)(dog(x) \wedge (\exists y)(cat(y) \wedge chased(x, y)))$	$some(X, dog(X), some(Y, cat(Y), chased(X, Y)))$
<i>A dog chased every cat</i>	$(\exists x)(dog(x) \wedge (\forall y)(cat(y) \Rightarrow chased(x, y)))$	$some(X, dog(X), all(Y, cat(Y), chased(X, Y)))$
<i>Every dog chased a cat</i>	$(\forall x)(dog(x) \Rightarrow (\exists y)(cat(y) \wedge chased(x, y)))$	$all(X, dog(X), some(Y, cat(Y), chased(X, Y)))$
<i>Every dog chased every cat</i>	$(\forall x)(dog(x) \Rightarrow (\forall y)(cat(y) \Rightarrow chased(x, y)))$	$all(X, dog(X), all(Y, cat(Y), chased(X, Y)))$
<i>Max does not bark</i>		$no(_, true, barks(max))$
<i>All dogs do not bark</i>		$all(X, dog(X), no(_, true, barks(X)))$
<i>No dog barks</i>		$no(X, dog(X), barks(X))$

Figura 25. Lenguaje de fórmula lógica propuesto por Covington.

Otro ejemplo más de ambigüedad aparece en la lectura de sintagmas nominales en plural. Por ejemplo, en la siguiente frase: *Tres chicos compraron cinco rosas*, Covington [42] distingue tres diferentes lecturas:

- ◆ *Lectura distributiva*, donde se podría entender que hay tres chicos, de los cuales cada uno compró cinco rosas. Esta lectura en castellano se haría más explícita al añadir *cada uno*, es decir: *Tres chicos compraron cinco rosas cada uno*.
- ◆ *Lectura colectiva*, donde los tres chicos como un grupo compraron cinco rosas. Esta interpretación supone el añadir el concepto de *conjunto* o *colectivo* que supone un referente del discurso con varios atributos, como por ejemplo los elementos que componen ese conjunto (cada uno de los tres chicos), la cardinalidad o número de elementos (en este caso tres) y finalmente las propiedades que comparten todos los elementos (aquí sería $\forall x$ (*boy* (x))).
- ◆ Y finalmente la *lectura conjunta*, en la que se interpretaría que un total de tres chicos compraron rosas (con posibilidad de que cada uno comprase diferente número de rosas) siendo el número total de rosas compradas cinco.

En [3] Allen considera necesario añadir dos restricciones en los modificadores para distinguir entre las lecturas distributivas o colectivas de un conjunto de objetos. La primera mostraría las restricciones sobre el conjunto (*SR*), y la segunda las restricciones sobre cada elemento individual del conjunto (*IR*). En la Figura 26 se muestran dos ejemplos de estas restricciones. En el segundo ejemplo se trabaja sobre el sintagma *tres mujeres* en el que como variable del cuantificador se escoge a *W1*, como restricción sobre el conjunto se exige que su cardinalidad sea tres y como restricción sobre cada elemento individual del conjunto se exige que cada uno de los elementos sea una mujer.

" <i>algunos gatos negros</i> "	DE=C1, SR= $ C1 >1$, IR= $\{C1 \subseteq \{c/\text{Gato}(c)\&\text{negro}(c)\}$
" <i>tres mujeres</i> "	DE=W1, SR= $ W1 =3$, IR= $\{W1 \subseteq \{w/\text{Mujer}(w)\}$

Figura 26. Representación de sintagmas nominales plurales según Allen.

Este lenguaje de CPPO también se puede enriquecer en diversos aspectos para tratar de incluir más información dentro de la fórmula lógica, como por ejemplo la información referente al tiempo de la oración (presente, futuro, etc.). De este modo, se podrían representar los distintos significados de las frases: *Juan ve a Luis*, *Juan vio a Luis* y *Juan verá a Luis*, en sus diferentes fórmulas lógicas: *presente ver*(Juan, Luis), *pasado ver*(Juan, Luis) y *futuro ver*(Juan, Luis).

Con esto damos por concluida esta sección en la que hemos descrito el concepto de gramática, análisis sintáctico, análisis semántico y el lenguaje de la fórmula lógica basado en el CPPO que utilizaremos en este trabajo. En la próxima sección describiremos brevemente diversos formalismos gramaticales relacionados con las *gramáticas de unificación de huecos (SUG)*, formalismo que utilizaremos para representar la información necesaria para la resolución de la anáfora.

4.2 Gramáticas basadas en la unificación

4.2.1 Fundamentos teóricos

Una de las familias de gramáticas más indicadas para el procesamiento del lenguaje natural es la que se engloba bajo el nombre de *gramáticas de unificación* (*UG*, *Unification Grammar*). Las *UG* tal y como se definen en Covington [42] y en Shieber [150] son aquellas que están basadas en la operación denominada *unificación*²². Estas gramáticas codifican la información tanto sintáctica como semántica en pares (*atributo, valor*) asignando un determinado valor a un atributo únicamente a través de esta operación de unificación. A estas gramáticas también se las conoce como *unification-based* (por la operación de unificación en la que están basadas) o *complex-feature-based* (por las estructuras de pares (*atributo, valor*) en que codifican la información).

A estos conjuntos de pares (*atributo, valor*) se les ha llamado de diversas maneras. En este trabajo los denominaremos: *estructura de rasgos* o *feature structures*. En las *LFG*: *f-structures*; en las *GPSG*: *feature matrices* o *categories*; en las *FUG*: *functional structures*; y en las *DCG*: *terms*. La notación habitual de estas estructuras de rasgos es la mostrada en la Figura 27 para el caso de un sintagma nominal de nombre *SN* con tres atributos: *coordinacion*, *f* y *g*. Las características principales de estas estructuras de rasgos son dos: por un lado son por sí mismas *estructuradas* (en la Figura 27 podemos observar el ejemplo de la estructuración dada por la característica *coordinación* en dos nuevos atributos: *numero* y *persona*), y por otro lado pueden *compartir estructuras* (*reentrancy*) lo cual significa que dos estructuras de rasgos pueden compartir el mismo valor. Ese valor compartido se denota utilizando índices (por ejemplo el valor compartido entre los atributos *f* y *g* de la Figura 27).

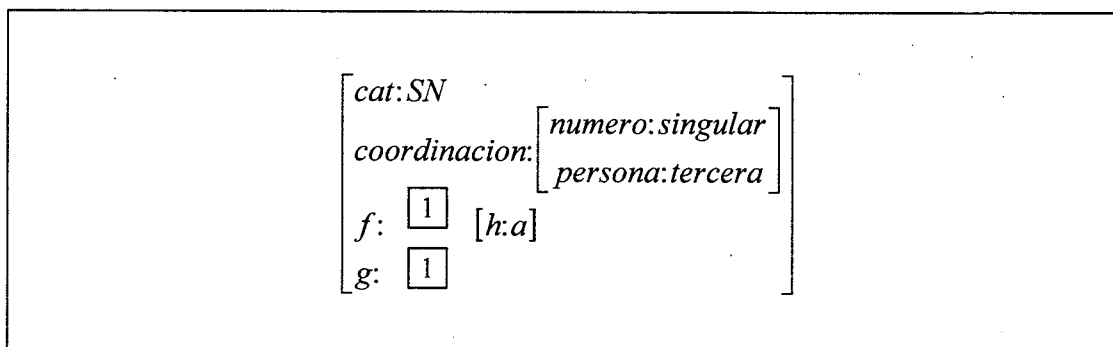


Figura 27. Ejemplo de una “estructura de rasgos”.

Cada uno de los atributos tiene asignado un determinado *tipo* que restringe los posibles valores que puede tomar, por ejemplo el atributo *número* sólo podrá tener los

²² Esta operación de unificación se define entre los pares (*atributo, valor*). Por ello definiremos formalmente esta operación después de la descripción de estos pares que realizamos a continuación.

valores *singular* o *plural*. El conjunto de *tipos* está parcialmente ordenado, existiendo una relación jerárquica. Como ejemplo, en la Figura 28 se muestra la relación jerárquica existente entre los atributos *signo*, *palabra* y *sintagma*. Los tipos que contienen más información están más bajos en la jerarquía, por ejemplo *palabra* o *sintagma*; y los tipos más generales estarán más altos en ella, *signo*. Además los subtipos herederán la información de sus supertipos.

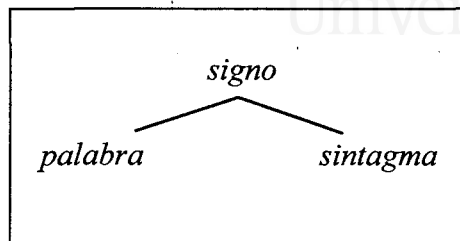


Figura 28. Ejemplos de relaciones jerárquicas entre atributos.

El hecho por el que se les ha considerado a estas gramáticas como un grupo es debido fundamentalmente a la dependencia que tienen de la operación de *unificación* que pasamos a definir a continuación. En primer lugar es necesario definir la relación de *subsunción* entre estructuras de rasgos. Ésta es un orden parcial que se establece entre las estructuras de rasgos mediante el cual pueden ser ordenadas parcialmente según su especificidad. De este modo una estructura más general *subsume* a las más específicas (siempre que no contengan información conflictiva). Podemos observar un ejemplo en la Figura 29, donde decimos que D_{SN3sg} tiene más información que (o es menos general que) D_{SN} , por lo que D_{SN} subsume a D_{SN3sg} , o de otro modo $D_{SN3sg} \subseteq D_{SN}$.

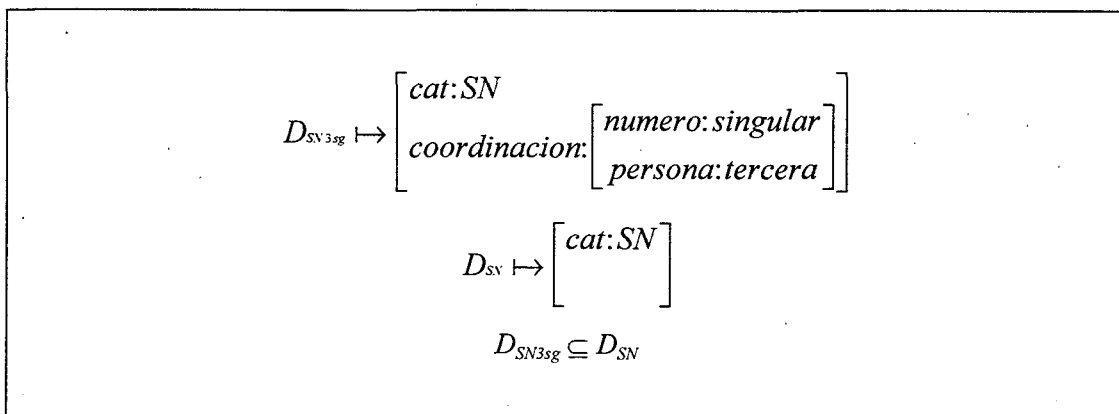


Figura 29. Ejemplo de "subsunción", en la que D_{SN} subsume a D_{SN3sg}

Es importante destacar que esta operación no relaciona a todo par de estructuras, ya que se puede dar el caso en que dos estructuras tengan información diferente pero compatible, caso de la Figura 30, o también caso que tengan información conflictiva, caso de la Figura 31.

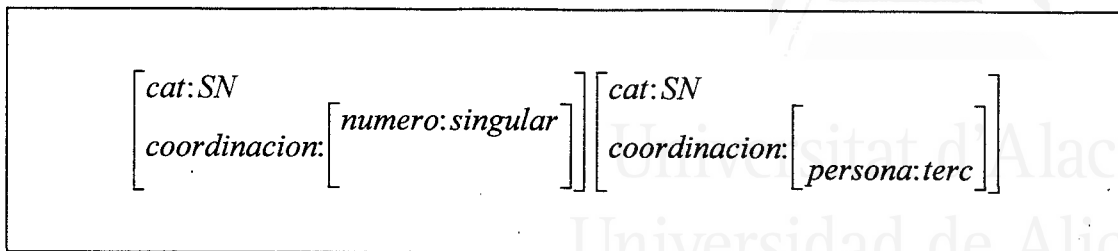


Figura 30. Ejemplo de dos estructuras con información diferente pero compatible.

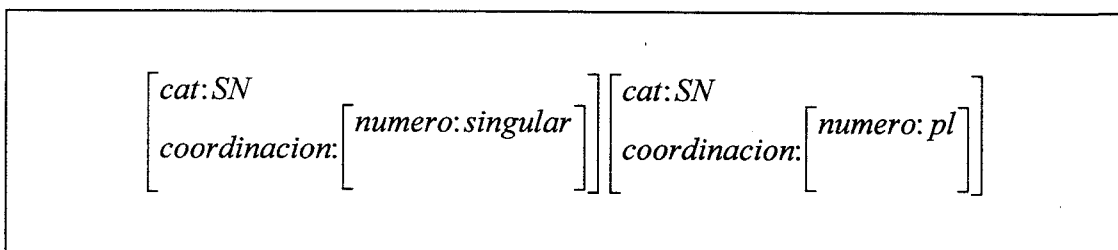


Figura 31. Ejemplo de dos estructuras con información diferente y conflictiva.

Y finalmente, siguiendo el trabajo de Shieber [150] se define la operación de *unificación* de dos estructuras D' y D'' , como la estructura más general D , tal que $D' \subseteq D$ y $D'' \subseteq D$. A esta operación la denotaremos como $D = D' \cup D''$, y de este modo el resultado de unificar dos estructuras consistirá en otra nueva estructura que contiene toda la información de las dos sin añadir nada más. Como resultado se obtiene la descripción más general que es compatible con las dos. Según esta definición es fácil adivinar que se pueden encontrar diferentes pares de estructuras imposibles de unificar, en cuyo caso hablaremos de *unificación fallida*, como por ejemplo las estructuras de la Figura 31.

Una vez detalladas las estructuras de rasgos y la operación que trabaja con ellas, la unificación, a continuación describiremos otro módulo importante dentro de las gramáticas de unificación: las *reglas inductivas* o *gramaticales*. Estas reglas definen por un lado la manera en que unas palabras se combinan con otras para formar unidades mayores del lenguaje llamadas *constituyentes*, y por otro lado también muestran cómo se relacionan las estructuras de rasgos asociadas. Estas reglas se suelen utilizar como una abstracción de las reglas de contexto libre. En la Figura 32, podemos observar un ejemplo de una regla de este tipo, sobre el formalismo PATR-II diseñado por Shieber y otros colaboradores en [147][151].

$X_0 \rightarrow X_1 X_2$ $\langle X_0 \text{ cat} \rangle = S$ $\langle X_1 \text{ cat} \rangle = SN$ $\langle X_2 \text{ cat} \rangle = SV$ $\langle X_0 \text{ head} \rangle = \langle X_2 \text{ head} \rangle$ $\langle X_0 \text{ head subject} \rangle = \langle X_1 \text{ head} \rangle$	De manera abreviada: $O \rightarrow SN SV$ $\langle S \text{ head} \rangle = \langle SV \text{ head} \rangle$ $\langle S \text{ head subject} \rangle = \langle NP \text{ head} \rangle$
--	--

Figura 32. Ejemplo de una regla gramatical de PATR-II.

También es importante destacar, que en determinadas variaciones de las UG se eliminan estas reglas expresándose esta información dentro de la propia estructura de rasgos, formando lo que se conoce como *sistemas lexicalistas*. En estos sistemas gran parte de la información necesaria para el análisis se encuentra en las estructuras especificadas en el diccionario. Ejemplos de gramáticas basadas en la unificación que se incluyan dentro de estos sistemas tenemos las *FUG* [90][89], y las *LFG* [19]. En la Figura 33, podemos observar cómo se representaría una estructura de rasgos que viene a sustituir a la regla $O \rightarrow SN SV$ dentro de la gramática *FUG*. En esta figura, se puede apreciar que en las *FUG* se añaden dos nuevos atributos respecto las estructuras de rasgos anteriores: *cset* y *pattern*. Estos nuevos atributos son a su vez un conjunto de estructuras de rasgos, el cual nos define qué subconstituyentes y en qué orden se asociarán en la regla.

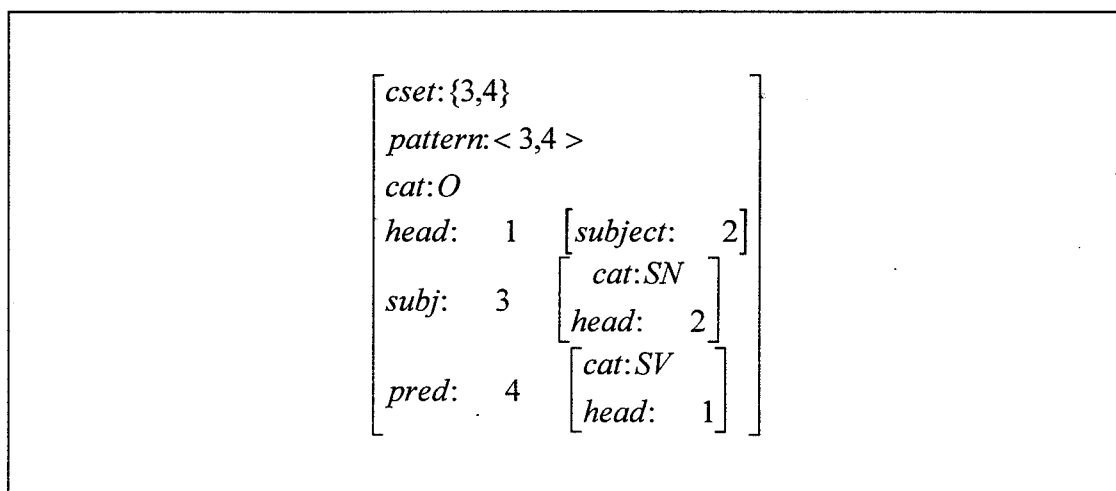


Figura 33. Estructura *FUG* equivalente a la regla mostrada en la Figura 32.

Y finalmente, otro módulo importante a destacar en toda gramática de unificación es el *diccionario* o *léxico*. Este diccionario, está formado por entradas léxicas que definen las estructuras de rasgos de las diferentes palabras del lenguaje. Aquí se almacena toda la información morfológica, fonológica, semántica y sintáctica de cada palabra, información que se utilizará para formar constituyentes de mayor tamaño.

A continuación pasaremos a mostrar una clasificación de las gramáticas basadas en la unificación, que nos servirá como apoyo en la elección final del formalismo a emplear en este trabajo.

4.2.2 Clasificación de las gramáticas de unificación

Estos formalismos, según Kevin Knight en [94], pueden clasificarse en cuatro grandes familias: *Norte Americana*, *categorial*, *GPSG* y *lógica*.

La primera de ellas, la familia *Norte Americana*, corresponde a los formalismos pertenecientes a trabajos en Lingüística Computacional desarrollados en Norte América a finales de 1970. En esta familia se enmarcan los siguientes formalismos: las *FUG* [90][89], y las *LFG* [19].

La segunda, la familia *categorial*, proviene de la investigación sobre la lingüística teórica en gramáticas categoriales, las cuales permitiendo la aplicación funcional usan la concatenación de cadenas para formar constituyentes. Los elementos informacionales son categorías complejas pudiendo verse como un rasgo functor evaluado categóricamente y un rasgo argumento. Variaciones sobre esta técnica se usan en la gramática *PATR-II* [147][151]. Entre los formalismos desarrollados en esta línea podríamos destacar: las *gramáticas categoriales de unificación (UCG, Unification Categorical Grammar)*, las *gramáticas combinatorio-categorial (CCG, Combinatory Categorical Grammar)*, y las *gramáticas meta-categorial (MCG, Meta-Categorical Grammar)*.

La tercera familia ha tomado el nombre de uno de sus formalismos: las *GPSG* [67]. Este formalismo se desarrolla en base a la idea de desarrollar un sistema restrictivo formal. Mientras que el fenómeno de manejar un amplio rango sintáctico y especialmente semántico se pensó no era posible en tales sistemas restringidos, este formalismo construyó un componente semántico dentro del mismo. Las *GPSG* describen un método para manejar las dependencias no ligadas sin recurrir a la transformación de estructuras. Pollard introdujo una variante de las *GPSG* añadiendo operaciones para combinar cadenas, reemplazando las metareglas de las *GPSG* por reglas léxicas y eliminando muchas de las restricciones subyacentes del dominio informacional, dando lugar a las denominadas *HPSG* [135].

El cuarto y último grupo, la familia *lógica*, consiste en formalismos gramaticales que trabajan en torno a la programación lógica. Aunque las gramáticas lógicas son independientes del lenguaje de programación utilizado en su implementación, el lenguaje *Prolog* es muy apropiado por ser la unificación una operación predefinida. Como gramáticas que se pueden incluir en este grupo aparecen las *gramáticas de metamorfosis (MG, Metamorfosis Grammar)* [40], las *DCG* [132], las *gramáticas de extraposición (EG, Extraposition Grammar)* [131], las *gramáticas de estructura modificada (MSG, Modified Structure Grammar)* [46], las *gramáticas lógico modulares (MLG, Modular Logic Grammar)* [102] y las *SG* [103][104].

La gramática que hemos elegido en esta Tesis para desarrollar el proceso de resolución de la anáfora, *SUG*, se incluye dentro del grupo de los formalismos gramaticales pertenecientes a la familia lógica, puesto que ha sido implementada en *Prolog* y además supone una extensión de las *DCG*. A continuación explicaremos en mayor detalle las gramáticas *DCG* y *SG* por ser las que mayor relación tienen con *SUG*.

4.2.2.1 GRAMÁTICAS DE CLÁUSULAS DEFINIDAS

Las *gramáticas de cláusulas definidas (DCG)* descritas por Pereira y Warren en [132], contienen todas las propiedades de las *gramáticas incontextuales (CFG)*, es decir permiten representar de manera clara y modular la estructura recursiva de las oraciones, pero superando las inadecuaciones de las *CFG* extendiéndolas en los siguientes aspectos:

- ◇ Permiten expresar las dependencias del contexto.
- ◇ Permiten la construcción de estructuras de árbol arbitrarias en el proceso de análisis sin verse restringidas por la estructura recursiva de la gramática.
- ◇ Permiten incluir condiciones extra en las reglas de la gramática.
- ◇ Extienden las *CFG* aumentando los símbolos no terminales con nuevos argumentos.

Formalmente se define una *DCG* como una cuaterna (T, NT, S, P) donde:

- ◆ T es el conjunto de símbolos terminales.
- ◆ NT es una extensión del conjunto de símbolos no terminales de una CFG, ya que a este conjunto pertenecen términos compuestos constituidos por predicado y argumentos.
- ◆ S es el símbolo inicial de la gramática, tal que $S \in NT$.
- ◆ P es el conjunto de reglas de producción. Las reglas DCG tienen la forma: $\alpha \rightarrow \beta$, donde $\alpha \in NT$, y $\beta \in (T \cup NT)^* \cup \{\text{llamadas a procedimientos}\}$.

En la Figura 34 podemos observar un ejemplo de gramática en notación DCG. La construcción de estructuras sintácticas o semánticas se realiza añadiendo argumentos extra a los predicados correspondientes a los símbolos no terminales y haciendo uso de la operación de unificación. Por ejemplo, siguiendo la gramática 2 de la Figura 34, la oración *El perro movía las orejas* se representaría por medio de la siguiente estructura: $o(sn(det(el), n(perro)), sv(v(movía), sn(det(las), n(orejas))))$.

Gramática 1: DCG sin argumentos

```

o → sn, sv
sn → det, n
sv → v, sn
det → [el]
det → [las]
det → [un]
n → [perro]
n → [hueso]
n → [orejas]
v → [comio]
v → [movía]

```

Gramática 2: DCG con argumentos

```

o(o(SN,SV)) → sn(SN), sv(SV).
sn(sn(D,N)) → det(D), n(N).
sv(sv(V,SN)) → v(V), sn(SN).
det(det(el)) → [el].
det(det(las)) → [las].
det(det(un)) → [un].
n(n(perro)) → [perro].
n(n(hueso)) → [hueso].
n(n(orejas)) → [orejas].
v(v(comio)) → [comio].
v(v(movía)) → [movía].

```

Figura 34. Ejemplos de gramáticas DCG.

Mediante las instanciaciones de los argumentos de los símbolos no terminales de una regla DCG pueden expresarse ciertas restricciones utilizando llamadas a procedimientos explícitos en el cuerpo de la regla. Por ejemplo, para expresar que el día de una fecha debe ser mayor que 0 y menor que 32 lo haríamos de la siguiente manera: $fecha(D,M) \rightarrow mes(M), [D], \{entero(D), 0 < D, D < 32\}$.

Los argumentos de los símbolos no terminales también pueden usarse para acarrear información contextual como la concordancia en género, número o persona. Por ejemplo, podemos expresar la concordancia en género y número entre el determinante y el nombre común de un sintagma nominal de la siguiente forma: $sn(Num, Gen) \rightarrow det(Num, Gen), n(Num, Gen)$.

Una DCG no es más que un conjunto de cláusulas definidas donde su significado es independiente de cualquier mecanismo de ejecución. Cada mecanismo de ejecución de las reglas de la gramática corresponde a una estrategia de análisis. Una de las ventajas de estas

gramáticas es que pueden ejecutarse casi²³ directamente en Prolog. La estrategia de análisis en Prolog será descendente y de izquierda a derecha con retroceso en un punto de fallo al inmediato anterior donde haya opciones a seguir distintas a la tomada anteriormente (algoritmo con retroceso o backtracking).

Las gramáticas DCG y todas las relacionadas con ella (SG, EG, MSG, etc.) utilizan una variación de las *estructuras de rasgos* descritas en la subsección 4.2.1, que son las denominadas *estructuras de términos*. Estos *términos* tienen la notación habitual de las estructuras en el campo de la Lógica y las Matemáticas: un símbolo predicado es seguido por un conjunto de nuevos predicados entre paréntesis. Los términos básicos son constantes y variables, los cuales se escriben en minúsculas y mayúsculas respectivamente siguiendo la convención de Prolog. En la Figura 35 podemos comprobar las diferencias entre las reglas gramaticales en la notación PATR-II, FUG y DCG.

<u>Gramática PATR-II:</u>	<u>Gramática FUG:</u>	<u>Gramática DCG:</u>
$O \rightarrow SN\ SV$ $\langle O\ head \rangle = \langle SV\ head \rangle$ $\langle O\ head\ subject \rangle = \langle SN\ head \rangle$	$cset: \{3,4\}$ $pattern: \langle 3,4 \rangle$ $cat: S$ $head: 1 \left[\begin{array}{l} subject: 2 \\ cat: NP \end{array} \right]$ $subj: 3 \left[\begin{array}{l} head: 2 \\ cat: VP \end{array} \right]$ $pred: 4 \left[\begin{array}{l} cat: VP \\ head: 1 \end{array} \right]$	$o(head(SubjHead,Form)) \rightarrow$ $sn(SubjHead), sv(head(SubjHead,$ $Form)).$

Figura 35. Equivalencia entre reglas de formalismos PATR-II, FUG y DCG.

Las estructuras de términos difieren de las estructuras de rasgos en dos importantes aspectos:

- ◆ En lugar de identificar los rasgos de manera explícita, los rasgos se identifican por medio del orden lineal fijo en que aparecen dentro del término.
- ◆ El número de argumentos de un término es importante en el proceso de unificación. A este número de argumentos se le denomina *aridad*.

Estos dos aspectos son muy importantes a tener en cuenta puesto que obligan a que las estructuras a unificar tengan la misma aridad y el mismo tipo de argumentos. Por ejemplo, las siguientes estructuras no serían unificables: *coordinacion (terc, sing)*, *coordinacion (sing, terc)* o *coordinacion (terc)*. Estos aspectos también muestran las propias desventajas de este formalismo, como es el hecho de tener que poner en cada

²³ La “casi” ejecución en Prolog se refiere a que necesitan de un traductor de cláusulas DCG a Prolog, aunque este tipo de traductores lo suelen incorporar todos los intérpretes y compiladores de Prolog.

término a unificar exactamente el mismo número y tipo de argumentos (aunque haya que dejarlos vacíos), lo que complica su manejo en sistemas de gran tamaño. Sin embargo, las gramáticas DCG nos ofrecen la ventaja de poderse ejecutar casi directamente en Prolog, lo cual nos permite su posterior integración con otros sistemas también implementados en este lenguaje.

Un sistema desarrollado basado en las DCG es ORBI [133], sistema experto en un entorno biofísico específico con un interfaz de lenguaje natural en portugués.

4.2.2.2 GRAMÁTICAS DE HUECOS

Las *gramáticas de huecos (SG)* fueron diseñadas por McCord en [103][105][104] y están basadas en la expresión sistemática de las reglas gramaticales y de la información lingüística en términos de *huecos (slots)* y *estructuras de huecos (slots frames)*. Este formalismo ha sido ampliamente estudiado y aplicado dentro del procesamiento del lenguaje natural como por ejemplo en Arrarte [7] y Palomar [129].

Los *huecos* representan relaciones gramaticales y son designados por nombres como *subj* (sujeto), *obj* (objeto directo) o *nadj* (adjetivo modificador del nombre). Hay huecos de dos tipos: *complementos* y *adjuntos*. Cada acepción de una palabra lleva asociada una lista de huecos complemento. Esta lista constituye la estructura de huecos de la acepción de la palabra. Además, para cada categoría léxica o parte de la oración se especifica en la gramática un conjunto de huecos adjunto. La información contenida en el léxico determina todos los posibles huecos para una acepción, de forma explícita para los complementos y de forma implícita para los adjuntos. La diferencia entre los huecos de tipo complemento y adjunto es que los de tipo complemento se usan cuando la relación gramatical es específica de determinadas palabras, y en cambio en los de tipo adjunto la relación gramatical es propia de todas las palabras de una determinada categoría léxica. Por ejemplo, el objeto directo es de tipo complemento puesto que es un rasgo propio de determinados verbos y debe ser incluido en la información léxica de cada uno de ellos. En cambio el hueco que corresponde a los adjetivos modificadores del nombre es de tipo adjunto puesto que está asociado a todas las palabras clasificadas como nombres en el léxico.

Cada hueco puede ser *opcional* u *obligatorio*. Un hueco puede ser siempre obligatorio (por ejemplo, el objeto de una preposición), o puede serlo para algunas palabras o acepciones. El proceso de análisis gramatical de una frase consiste básicamente en determinar qué relaciones del tipo *llenar el hueco S de* pueden establecerse entre los posibles análisis léxicos de sus palabras. El análisis resultante consta de una combinación de estructura superficial y forma lógica en una única estructura. La estructura superficial está expresada en forma de árboles de dependencia, en los que cada nodo lleva asociada una palabra núcleo P y cada uno de sus nodos hijos tiene por núcleo una palabra que llena un hueco de P. Además cada nodo va acompañado por la estructura de huecos de su núcleo, en el cual se especifican sus relaciones gramaticales, ya sean superficiales o profundas, inmediatas o remotas.

Este formalismo ha sido implementado en Prolog utilizando un analizador ascendente de tipo *chart*, tratándose de un *sistema lexicalista* ya que gran parte de la información necesaria para el análisis se encuentra en las estructuras de huecos especificadas en el diccionario.

Las reglas se dividen de manera modular en diferentes tipos que tratan diversos fenómenos relacionados con los huecos. Así tenemos reglas de llenado de huecos, reglas de orden sintácticas (indican cuándo un sintagma que llena un determinado hueco puede modificar a otro por la izquierda y cuándo por la derecha), de extraposición, etc.

Se hace uso de un sistema de evaluación numérica de los análisis basado en una estrategia heurística que compara los análisis obtenidos estableciendo un orden de preferencias. Este sistema de evaluación es utilizado no sólo para ordenar los análisis finales (en caso de haber más de uno), sino también para ordenar los análisis parciales obtenidos a lo largo del proceso, y de manera opcional descartar los de menor preferencia. Esto redundará en una mayor eficiencia de análisis y en la posibilidad de establecer restricciones gramaticales que evitan la proliferación de análisis alternativos, reduciendo de forma considerable el conjunto de análisis finales. Buena parte del sistema de evaluación se encuentra en el *shell* del sistema, aunque también en la gramática y en el léxico específicos de una lengua pueden definirse la información o reglas de evaluación convenientes.

4.3 Gramática de unificación de huecos (*SUG*)

El formalismo que utilizaremos en este trabajo es una extensión de las *DCG* al que hemos llamado *gramática de unificación de huecos (SUG, Slot Unification Grammar)* [53][128][60]. Esta sección la organizaremos en seis subsecciones. En la primera explicaremos los motivos que nos han llevado a introducir esta extensión de las gramáticas *DCG*, comentando brevemente las diferencias fundamentales entre ambas. A continuación definiremos formalmente las *SUG*. Seguidamente detallaremos las reglas de producción *SUG* centrándonos sobre todo en los operadores opcionales que definiremos posteriormente. Después detallaremos igualmente los denominados hechos *SUG*. Una vez definidas las *SUG* presentaremos un ejemplo completo de una gramática en esta notación. Y finalizaremos presentando el *analizador de reglas SUG* al que denominamos *SUP (Slot Unification Parser)*, analizador desarrollado en Prolog y sobre el que realizaremos las pruebas tanto de este formalismo *SUG* como del algoritmo de resolución de la anáfora que propondremos posteriormente.

4.3.1 *SUG* como una extensión de las *DCG*

Hemos realizado esta extensión de las *DCG* por los siguientes motivos:

- ◆ Facilitar la resolución de problemas lingüísticos de una forma modular. Algunos de estos problemas los hemos tratado ya por medio de las *SUG*: la coordinación en [53], la elipsis en [128], la extraposición en [61], y finalmente la anáfora en [59][54][57][55][56][52].
- ◆ Posibilitar el rehacer el orden de los componentes de una oración, como ocurriría en el caso del tratamiento de la extraposición o de la anáfora.
- ◆ Relajar la relación entre la gramática y el diccionario, alcanzando un mayor grado de independencia entre ambos. Este hecho se conseguirá al reducir el número de

accesos al diccionario por palabra a tan sólo uno durante toda la fase de análisis. Con ello se facilita el trabajo con diferentes diccionarios, y la aplicación de análisis sintácticos parciales o relajados sobre textos no restringidos en los que no tenemos la seguridad de obtener módulos de análisis sintáctico lo suficientemente robustos. También permitirá afrontar situaciones complicadas del tratamiento del léxico como el descomponer palabras combinadas en palabras sencillas, como por ejemplo palabras del tipo verbo más pronombre personal (*dime, cántame, darse*); o la situación inversa, comprobar si hay grupos de palabras (*multi-palabras*) que funcionan como palabras sencillas (*debajo de*).

- ◆ Mejorar la eficiencia de la implementación de las DCG en Prolog. Este objetivo se consigue al reducir el número de reglas gramaticales y el número de símbolos no terminales. También se alcanzará este objetivo ya que será capaz de recordar lo que ha sido analizado en una frase y lo que no, disminuyendo el número de intentos de análisis de cada constituyente de la gramática.

El motivo por el que se presenta una extensión de las gramáticas DCG es que el objetivo de esta Tesis está más próximo a la implementación computacional que a la teoría lingüística²⁴, por lo que partiremos de un formalismo gramatical de los cuales uno de los más extendidos son las DCG (por ejemplo Götz y Meurers en [68] diseñan un algoritmo que compila una teoría lingüística como son las gramáticas HPSG en gramáticas DCG).

Otro motivo por el que nos hemos inclinado por las DCG es que consideramos los modelos basados en reglas gramaticales más adecuados que los modelos lexicalistas (por ejemplo las *SG* [104]) para el tratamiento de textos no restringidos o para trabajar con diferentes diccionarios. Los consideramos más adecuados puesto que no presentan una dependencia del léxico tan importante que nos limite en el diseño de la gramática. Otro punto para el que también consideramos que son más viables los modelos basados en reglas gramaticales es para realizar análisis sintácticos parciales, o trabajando sobre la salida de un etiquetador tal y como lo hacen Kennedy y Boguraev en [93] (punto que expondremos con mayor detalle en posteriores secciones).

El sistema de resolución de la anáfora que proponemos en este trabajo utilizará este formalismo *SUG* para almacenar la información sintáctica. Este sistema optará por la modularidad con el objetivo de aumentar la independencia entre todos los componentes: gramática, diccionario, análisis sintáctico, análisis semántico y tratamiento de problemas lingüísticos. Con esta modularidad separamos el diseño de la gramática del módulo de tratamiento de problemas lingüísticos, de modo diferente a como Carvalho sugiere en [29]. Carvalho propone una gramática en la que se definen de modo explícito relaciones de coreferencia entre un pronombre y un sintagma nominal, es decir, extiende las DCG para que admitan reglas del tipo mostrado en la Figura 36. De este modo cuando se analiza la regla 1 se añade el sintagma nominal en una pila de sintagmas nominales que se utilizará

²⁴ Recordamos que la diferencia entre teoría lingüística y formalismo gramatical la vimos en la subsección 4.1.1. Esta diferencia consiste según Shieber [150] en el distinto objetivo con el que son diseñados, lo cual hace que presenten distintas características o herramientas. Los formalismos gramaticales incluyen mecanismos generales para incrementar el poder expresivo de la gramática, mientras que las teorías lingüísticas tienden a incorporar mecanismos diseñados para un particular tipo de análisis lingüístico

para la búsqueda de posibles antecedentes de los posteriores pronombres que se intenten resolver. Esta pila contendría una estructura formada por tres argumentos:

- ◆ El camino del árbol sintáctico en el que se encuentra el sintagma nominal en forma de cadena numérica: $N_1-N_2-N_3-\dots-N_n$ en la que cada N_i corresponde a la posición que ocupa el nodo en el árbol sintáctico. Esta cadena se utilizará para comprobar las restricciones c-dominio.
- ◆ La estructura $ref(X', P', G')$ en la que el primer argumento supone la variable de la fórmula lógica, el segundo su fórmula lógica y el tercero el género del sintagma nominal. Por ejemplo, para el sintagma nominal: *every man* se almacenaría el siguiente valor: $ref(X', man(X'), masc)$.
- ◆ Tipo de cuantificador (O'). Por ejemplo, para: *every man*, se almacenaría el valor: *univ*.

1	$nounph(_) / -O- / ref(X, P, G) \rightarrow determiner(O), noun(X, P, G).$
2	$nounph(Case) \rightarrow ref(_, _, G) \leftarrow \leftarrow pronoun(Case, G).$
3	$pronoun(Case, G) \rightarrow [W], \{pro(W, Case, G)\}.$
4	$noun(X, P, G) \rightarrow [W], \{noun(G, X, P, W)\}.$
5	$noun(masc, X, man(X), man).$
6	$determiner(univ) \rightarrow [every].$
7	$determiner(indef) \rightarrow [a].$
8	$pro(she, nom, fem).$

Figura 36. Gramática propuesta por Carvalho para definir relaciones de correferencia.

La desventaja de este tipo de resolución de la anáfora es precisamente la falta de modularidad, es decir están totalmente interrelacionados los módulos de análisis sintáctico y del diseño de la gramática, lo cual dificulta su posterior mantenimiento en sistemas reales de procesamiento del lenguaje natural tal y como también se declara en el sistema propuesto por Fischer en [62].

A continuación presentaremos las características del formalismo *SUG*, realizando las necesarias comparaciones con el formalismo gramatical *DCG* y mostrando una serie de ejemplos de gramáticas en notación *SUG*.

4.3.2 Definición formal

Las gramáticas de unificación de huecos (*SUG*, *Slot Unification Grammar*) se definen como una quintupla (T, NT, S, H, P) donde:

- ◆ T es un conjunto de símbolos terminales.
- ◆ NT es un conjunto finito de símbolos no terminales, tal que $T \cap NT = \emptyset$.
- ◆ S es el símbolo inicial de la gramática, tal que $S \in NT$.
- ◆ H es conjunto de los denominados hechos *SUG*. Estos hechos son *coordinated*, *juxtaposition*, *fusion*, *basicWord* o *esPalabra*.

♦ P es el conjunto de reglas de producción con la siguiente forma:

- $\alpha ++> \beta$
- β_i

donde $\alpha \in NT$, $\beta \in (T \cup NT)^* \cup \{\text{llamadas a procedimientos}\}$ y $\beta_i \in H$.

A continuación vamos a definir detalladamente las características de las reglas de producción y hechos SUG.

4.3.3 Reglas de producción

Como ya hemos comentado anteriormente las SUG son una extensión de las DCG, por lo que heredarán muchas de sus características. La principal diferencia entre ambas es que las reglas de producción SUG son de la forma $\alpha ++> \beta$ (en las DCG se utiliza la notación $\alpha \rightarrow \beta$), y suponiendo $\beta = \beta_1, \beta_2, \dots, \beta_n$, cada subconstituyente β_i puede omitirse en el constituyente α si se escribe entre el *operador opcional*: “<< β_i >>”. Este subconstituyente β_i se considerará *opcional* únicamente en esa regla gramatical.

Podemos encontrar un ejemplo de la aplicación del uso de este operador que indica la presencia de un constituyente opcional en las reglas gramaticales de los sintagmas nominales. Supongamos que en este ejemplo intentamos escribir las reglas gramaticales que reconozcan los siguientes tipos de sintagma nominal: *Juan, el libro, ese pobre hombre, ese hombre pobre, ese bonito libro gris, el libro de tu hermana* y *ese bonito libro gris de tu hermana*. Esto nos supondría el crear una regla gramatical para cada tipo de sintagma nominal tal y como se muestra en la Figura 37. En definitiva tendríamos que prever todas las combinaciones posibles entre los constituyentes: *determinante, adjetivo, nombre y sintagma preposicional*, de los cuales tan sólo el *nombre* aparece en todas ellas (tiene un carácter *obligatorio* al ejercer la función de núcleo del sintagma nominal).

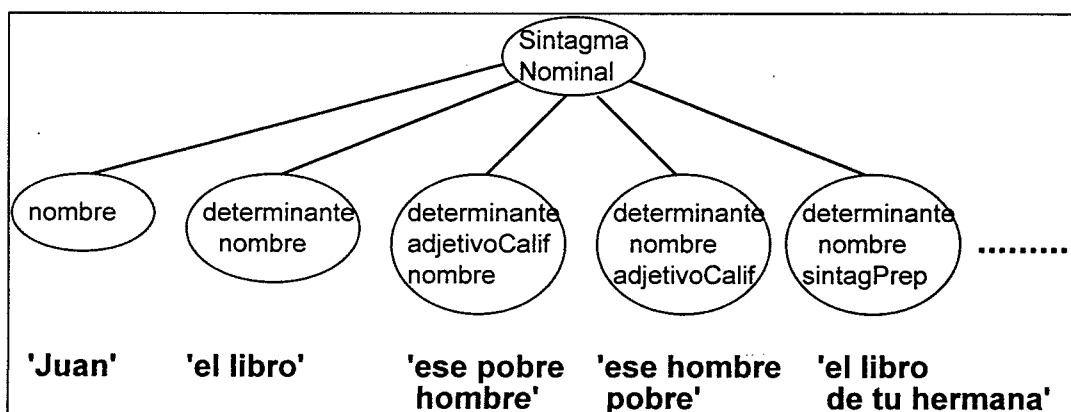


Figura 37. Ejemplos de sintagmas nominales.

En la Figura 38 se muestran todas estas reglas gramaticales en notación DCG. Para reducir el número de reglas gramaticales, existe la posibilidad de añadir la propiedad de opcionalidad en cada uno de los constituyentes de la gramática, o sea, que al escribir una determinada regla gramatical podamos incluir la propiedad por la que sus subconstituyentes

pueden aparecer o no en el mismo. Incluyendo esta propiedad nos aparece una variación de las gramáticas DCG en la que se definen componentes opcionales por medio de las dos siguientes reglas DCG:

constituyenteOpcional → *constituyenteObligatorio*.
constituyenteOpcional → [].

El resultado de escribir las anteriores reglas gramaticales en esta variación de las DCG se muestra en la Figura 38 bajo el epígrafe *gramática DCG con constituyentes opcionales*.

Con esta variación de las DCG se consigue reducir el número de reglas gramaticales, pero nos vemos obligados a tener que distinguir entre constituyentes opcionales y obligatorios, teniendo que añadir nuevos símbolos no terminales (por ejemplo, además del símbolo no terminal *adjetivo* tendremos que añadir el símbolo *opcAdjetivo* que tendrá carácter opcional).

Gramática DCG:	Gramática DCG con constituyentes opcionales:
sn → determinante, nombre. sn → determinante, adjetivo, nombre. sn → determinante, nombre, adjetivo. sn → determinante, nombre, sintagmaPreposicional. sn → determinante, adjetivo, nombre, adjetivo. sn → determinante, adjetivo, nombre, sintagmaPreposicional. sn → determinante, nombre, adjetivo, sintagmaPreposicional. sn → determinante, adjetivo, nombre, adjetivo, sintagmaPreposicional. sn → adjetivo, nombre. sn → adjetivo, nombre, adjetivo. sn → adjetivo, nombre, sintagmaPreposicional. sn → adjetivo, nombre, adjetivo, sintagmaPreposicional. sn → nombre. sn → nombre, adjetivo. sn → nombre, sintagmaPreposicional. sn → nombre, adjetivo, sintagmaPreposicional.	sn → opcDeterminante, opcAdjetivo, nombre, opcAdjetivo, opcSintPrep. opcDeterminante → determinante. opcDeterminante → []. opcAdjetivo → adjetivo. opcAdjetivo → []. opcSintPrep → sintagmaPreposicional. opcSintPrep → [].
	Gramática <i>SUG</i>:
	sn ++ <<determinante>>, <<adjetivo>>, nombre, <<adjetivo>>, <<sintPrepos>>.

Figura 38. Comparación entre DCG y *SUG*: constituyentes opcionales.

Sin embargo si utilizamos el operador opcional *SUG* superamos las anteriores notaciones DCG y DCG con constituyentes opcionales en los siguientes tres puntos:

- ◆ Con *SUG* no tenemos que añadir nuevos símbolos no terminales (tal y como ocurre con las DCG con constituyentes opcionales). Tan sólo hemos que colocar el constituyente entre los operadores << >> en la regla en que aparezca con carácter opcional. En la Figura 38 bajo el epígrafe *gramática *SUG** se expone la misma gramática en notación *SUG*. Como se puede apreciar, únicamente se necesita una regla gramatical en la que el determinante, el adjetivo (ya sea anterior o posterior al nombre) y el sintagma preposicional son constituyentes opcionales. En esta regla *SUG* sólo aparecerá un constituyente obligatorio: el nombre.
- ◆ Otra ventaja de este operador opcional es que nos permite comprobar si un determinado constituyente opcional ha sido analizado o no. Esto se llevará a cabo

añadiendo una etiqueta delante del constituyente opcional: << *Etiqueta* : *constituyente* >>, de tal modo que si el constituyente no fuese analizado el predicado *existe (Etiqueta)* sería falso. En caso contrario, o sea que el constituyente haya sido analizado, el predicado *existe (Etiqueta)* será cierto. Es conveniente remarcar que la comprobación por medio del predicado *existe(Etiqueta)* debe ir entre llaves al considerarse llamadas a procedimientos y no reglas gramaticales en sí mismas. En la Figura 39 se muestra un ejemplo del uso del operador opcional en combinación con estos predicados para la regla gramatical de las oraciones tanto personales como impersonales (en las últimas es obligatorio la ausencia de sujeto, por ejemplo en la oración *Llueve*). En esta figura aparece una llamada al predicado Prolog *ifthen (condicion, accion)*. Este predicado junto a *ifthenelse (condicion, accion, accionSino)*, son predicados específicos del intérprete Arity Prolog. Ambos predicados serán utilizados en diversos ejemplos de este trabajo. Para las versiones de Prolog en las que no están implementados estos predicados, son fáciles de añadir mediante las reglas Prolog expuestas en la Figura 40.

- ◆ Y una última ventaja de este operador es la de permitir una opcionalidad condicionada a que se cumpla una determinada restricción. Esto se consigue mediante la siguiente sintaxis: << *constituyente, restricción* >>. Por ejemplo, en la Figura 41 se ha escrito la regla gramatical del constituyente *adyacenteAdjetivo*, que estará formado por un adverbio opcional y un adjetivo calificativo. El adverbio que precede al adjetivo solamente sería opcional en caso que sea de cantidad (*los libros muy pesados*), de modo (*los libros frecuentemente pesados*) o de comparación (*los libros más pesados*), pero nunca podrá ser un adverbio de tiempo o lugar. Por ello esta opcionalidad del adverbio dentro del constituyente *adyacenteAdjetivo* se ve restringida a que sea un adverbio de tipo cantidad, modo o comparación.

```

oracSimple(Numero,Genero,Persona) ++>
<< SN:sn(Numero,Genero,Persona) >>,
sv(Numero,Genero,Persona,Tipo),
{ ifthen( (Tipo==impers ; Tipo==impersOD),
noExiste(SN) ) }.
% Las oraciones impersonales no tienen sujeto

```

Figura 39. Reglas gramaticales *SUG* para las oraciones personales e impersonales.

```

ifthen(Condicion,Entonces) :-
( call(Condicion),!, call(Entonces) ); true ).
ifthenelse(Condicion,Entonces,Sino) :-
( call(Condicion),!, call(Entonces) ); call(Sino) ).

```

Figura 40. Implementación en Prolog de los predicados “*ifthen*” e “*ifthenelse*”.

```

adyacenteAdjetivo (Numero, Genero) ++>
  << adverbio (TipoAdv), (TipoAdv == cantidad ; TipoAdv == modo ; TipoAdv == comparacion) >>,
  adjetivo (Numero, Genero, adjetivoCalificativo).

sn (Numero, Genero) ++> <<determinante (Numero, Genero) >>, nombre (Numero, Genero),
  <<adyacenteAdjetivo (Numero, Genero)>>.

```

Figura 41. Constituyentes opcionales con restricciones.

4.3.4 Hechos SUG

Otra de las novedades introducidas en SUG que las diferencian de las DCG es la presencia del conjunto de los denominados *hechos SUG*. Estos hechos son: *coordinated*, *juxtaposition*, *fusion*, *basicWord* o *esPalabra*, los cuales pasamos a explicar a continuación.

4.3.4.1 HECHO “COORDINATED”

Este hecho resuelve la coordinación entre constituyentes por medio de conjunciones. Tiene la siguiente sintaxis: *coordinated* (*A*, *ASimple*, *Restricción*), donde:

- ◊ *A* es el nombre del constituyente coordinado.
- ◊ *ASimple* es el nombre de los subconstituyentes que se coordinan.
- ◊ *Restricción* es la condición que se exige para que se puedan coordinar varios constituyentes *ASimple*.

Por ejemplo, el siguiente hecho analizará la coordinación de oraciones:

- ◆ *coordinated* (*oCoord*, *oSimple*, *_*), donde:
 - ◊ *oCoord* es el nombre del constituyente “oración coordinada”. Éste será el nombre del constituyente coordinado, es decir, cada vez que necesitemos un constituyente oración coordinada, lo denotaremos por *oCoord*.
Un ejemplo se daría en la definición de oraciones de relativo que están formadas por un pronombre relativo y una oración que puede aparecer coordinada: *El perro [que compró Luis y trajo Pedro]_{oRel} está enfermo*. Esta regla se escribiría del siguiente modo:
oRel ++> *pron* (relativo), *oCoord*.
 - ◊ *oSimple* es el nombre de las “oraciones simples” que se coordinan. Este constituyente se definirá a su vez mediante la siguiente regla de producción SUG:
oSimple ++> <<sn>>, *sv*.
la cual indica que las oraciones simples a coordinar se constituirán a su vez por un *sn* opcional y un *sv*.
 - ◊ *_* indica que no hay ninguna restricción para la coordinación. O sea, se permitirá la coordinación de cualquier tipo de oraciones.

Otro ejemplo se daría en la coordinación de adjetivos. Teniendo en cuenta que existen distintos tipos de adjetivos, sólo se permitirá la coordinación de adjetivos calificativos:

- ◆ *coordinated (adj (TipoAdj), adjSimple (TipoAdj), TipoAdj == calif)*, donde:
 - ◇ *adj* es el nombre del constituyente “adjetivo coordinado”. Éste será el nombre del constituyente coordinado, es decir, cada vez que necesitemos un constituyente adjetivo, lo denotaremos por *adj*.
Un ejemplo se daría en la definición de un sintagma nominal que tendrá como modificador un adjetivo demostrativo (que no se permitirá su coordinación, como ocurriría en: **[Este y aquel] libro no me gustan*) y un adjetivo calificativo (luego pueden aparecer varios adjetivos calificativos coordinados: *Este libro [azul y verde] es interesante*). Esta regla se escribiría del siguiente modo:

$$sn \text{ ++> } \langle\langle \text{adj (dem) } \rangle\rangle, \text{ nombre}, \langle\langle \text{adj (calif) } \rangle\rangle.$$
 - ◇ *adjSimple* es el nombre de los “adjetivos simples” que se coordinan. Estos constituyentes se dividen en diferentes clases en función de su argumento: *dem* o *calif*.
 - ◇ *TipoAdj == calif* indica que la coordinación se permitirá únicamente entre adjetivos calificativos, y no entre el resto de tipos de adjetivo (demostrativos, ordinales o numerales).

Este hecho *SUG* sería equivalente a las dos siguientes reglas DCG:

$$A \rightarrow A\text{Simple}, \text{ Restricción}, \text{ conjunción}, A.$$

$$A \rightarrow A\text{Simple}.$$

En la Figura 42 se muestra la comparación entre reglas DCG y *SUG* para llevar a cabo la coordinación de adjetivos calificativos, exigiendo que haya concordancia de número y género entre los adjetivos calificativos coordinados. Al aparecer como restricción para la coordinación que *Tipo == adjetivoCalificativo*, se rechaza la coordinación del resto de adjetivos, es decir, no se aceptaría la coordinación de adjetivos demostrativos en **[Este y ese] coche*, pero sí que se daría como válida la coordinación en *Este coche [azul y verde]*. También en esta figura se ha escrito la regla *SUG* para los sintagmas nominales, que estarán compuestos por un adjetivo calificativo opcional coordinado.

Reglas DCG	Reglas <i>SUG</i>
$\text{adjetivo (Numero, Genero, Tipo)} \rightarrow$ $\text{adjetivoSimple (Numero, Genero, Tipo),}$ $\{\text{Tipo} == \text{adjetivoCalificativo}\},$ $\text{conjuncion},$ $\text{adjetivo (Numero, Genero, Tipo)}.$	coordinated ($\text{adjetivo (Numero, Genero, Tipo),}$ $\text{adjetivoSimple (Numero, Genero, Tipo),}$ $\text{Tipo} == \text{adjetivoCalificativo}$ $\text{)}.$
$\text{adjetivo (Numero, Genero, Tipo)} \rightarrow$ $\text{adjetivoSimple (Numero, Genero, Tipo)}.$	$sn(\text{Numero, Genero}) \text{ ++> } \langle\langle \text{determ(Numero,}$ $\text{Genero) } \rangle\rangle, \text{ nombre(Numero, Genero),}$ $\langle\langle \text{adjetivo (Numero, Genero, adjetivoCalificativo) } \rangle\rangle.$

Figura 42. Ejemplo de coordinación con restricción: adjetivos calificativos.

En la Figura 43 se puede observar las reglas DCG y *SUG* equivalentes que resuelven la coordinación de oraciones, en la que no existe ninguna restricción para la coordinación,

por lo que aparece el tercer argumento del hecho *coordinated* con el valor “_”. Estas reglas aceptan la oración “Juan comió y bebió”.

Reglas DCG	Reglas SUG
<i>oracion</i> → <i>oracionSimple</i> , <i>conjuncion</i> , <i>oracion</i> . <i>oracion</i> → <i>oracionSimple</i> .	<i>coordinated</i> (<i>oracion</i> , <i>oracionSimple</i> , _).
<i>oracionSimple</i> → <i>sintagmaNominal</i> , <i>sintagmaVerbal</i> . <i>oracionSimple</i> → <i>sintagmaVerbal</i> .	<i>oracionSimple</i> ++> << <i>sintagmaNominal</i> >>, <i>sintagmaVerbal</i> .

Figura 43. Coordinación de oraciones.

En la Figura 44 se muestran las reglas gramaticales para la coordinación de sintagmas nominales en notación SUG. Estas reglas nos permitirán coordinar constituyentes de diferentes formas: sintagmas nominales formados por un nombre (ya sea propio o común), o un pronombre. De este modo, la oración *Ella, Pedro y el hijo de Juan compraron un libro* es aceptada por la gramática.

Reglas DCG	Reglas SUG
<i>sintagmaNominal</i> → <i>sintagmaNominalSimple</i> , <i>conjuncion</i> , <i>sintagmaNominal</i> . <i>sintagmaNominal</i> → <i>sintagmaNominalSimple</i> .	<i>coordinated</i> (<i>sintagmaNominal</i> , <i>sintagmaNominalSimple</i>).
<i>sintagmaNominalSimple</i> → <i>pronombre</i> .	<i>sintagmaNominalSimple</i> ++> <i>pronombre</i> .
<i>sintagmaNominalSimple</i> → <i>determinante</i> , <i>nombre</i> . <i>sintagmaNominalSimple</i> → <i>determinante</i> , <i>adjetivo</i> <i>nombre</i> .	<i>sintagmaNominalSimple</i> ++> << <i>determinante</i> >>, << <i>adjetivo</i> >>, <i>nombre</i> , << <i>adjetivo</i> >>, << <i>sintagmaPreposicional</i> >>.
<i>sintagmaNominalSimple</i> → <i>determinante</i> , <i>nombre</i> , <i>adjetivo</i> .	
<i>sintagmaNominalSimple</i> → <i>determinante</i> , <i>nombre</i> , <i>sintagmaPreposicional</i> .	
<i>sintagmaNominalSimple</i> → <i>determinante</i> , <i>adjetivo</i> <i>nombre</i> , <i>adjetivo</i> .	
...	
<i>oracion</i> → <i>sintagmaNominal</i> , <i>sintagmaVerbal</i> . <i>oracion</i> → <i>sintagmaVerbal</i> .	<i>oracion</i> ++> << <i>sintagmaNominal</i> >>, <i>sintagmaVerbal</i> .

Figura 44. Coordinación de sintagmas nominales de distinto tipo.

Una ventaja fácilmente apreciable de este hecho SUG sobre las reglas equivalentes en DCG es la reducción de reglas gramaticales. Otra ventaja es que para analizar un sintagma nominal no coordinado (siguiendo el ejemplo de la Figura 44) el constituyente *sintagmaNominalSimple* sólo será analizado una vez. Por el contrario, si utilizásemos notación DCG *sintagmaNominalSimple* sería analizado dos veces. En la Figura 45 se

muestra este proceso gráficamente. El motivo por el que en *SUG* se analizaría este componente tan sólo una vez es porque internamente el hecho *coordinated* se interpreta como la siguiente regla de producción *SUG*:

$$\text{sintagmaNominal}++>\text{sintagmaNominalSimple}, \langle\langle \text{CONJ:conjuncion} \rangle\rangle, \\ \{ ((\text{existe} (\text{CONJ}), \text{sintagmaNominal}) ; \text{true}) \}.$$

de este modo, en caso que no se encuentre una conjunción después del sintagma nominal simple se continuaría con el resto del análisis.

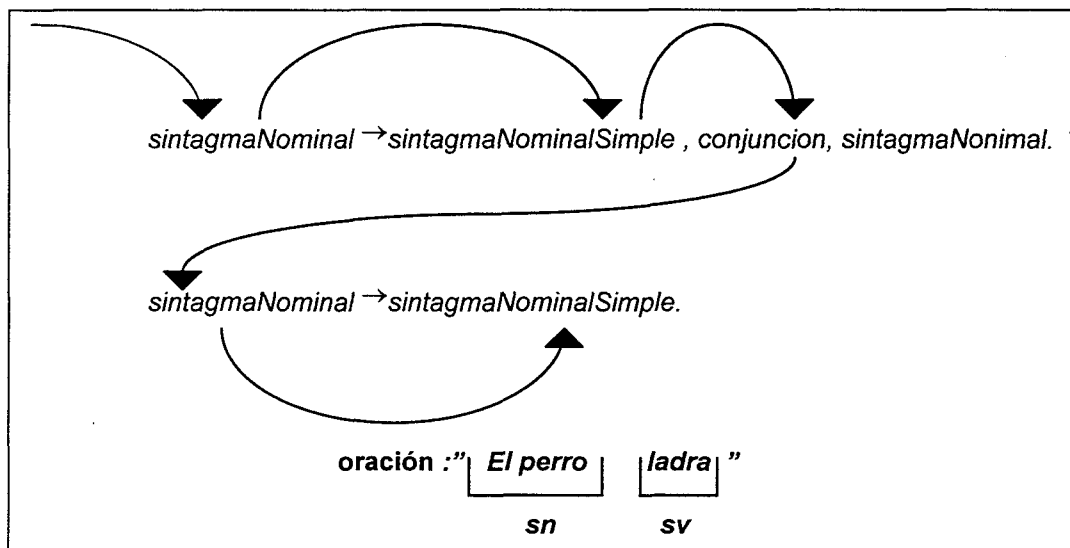


Figura 45. Análisis de la coordinación mediante notación DCG.

4.3.4.2 HECHO "JUXTAPOSITION"

En el lenguaje existe un tipo especial de coordinación que ocurre sin conjunciones al que se denomina *yuxtaposición*. Para llevar a cabo la *yuxtaposición* *SUG* prevé el hecho *juxtaposition* que tiene la siguiente sintaxis: *juxtaposition (A, ASimple, Restricción)*, donde:

- ◊ *A* es el nombre del constituyente yuxtapuesto.
- ◊ *ASimple* es el nombre de los subconstituyentes que se yuxtaponen.
- ◊ *Restricción* es la condición que se exige para que se puedan realizar la *yuxtaposición* de constituyentes *ASimple*.

Por ejemplo, el siguiente hecho analizará la *yuxtaposición* de adjetivos calificativos:

- ◆ *juxtaposition (adj (TipoAdj), adjSimple (TipoAdj), TipoAdj == calif)*, donde:
 - ◊ *adj* es el nombre del constituyente "adjetivo yuxtapuesto". Éste será el nombre del constituyente yuxtapuesto, es decir, cada vez que necesitemos un constituyente adjetivo, lo denotaremos por *adj*.

Un ejemplo se daría en la definición de un sintagma nominal que tendrá como modificador un adjetivo demostrativo (que no se permitirá su coordinación, como ocurriría en: **[Este y aquel] libro no me gustan*) y un adjetivo calificativo yuxtapuesto (luego pueden aparecer varios adjetivos calificativos

yuxtapuestos: *Este libro [azul verdoso oscuro] es interesante*). Esta regla se escribiría del siguiente modo:

$sn \text{ ++} > << \text{adj} (\text{dem}) >>, \text{nombre}, << \text{adj} (\text{calif}) >>.$

- ◇ *adjSimple* es el nombre de los “adjetivos simples” que se yuxtaponen. Estos constituyentes se dividen en diferentes clases en función de su argumento: *dem* o *calif*.
- ◇ *TipoAdj == calif* indica que la yuxtaposición se permitirá únicamente entre adjetivos calificativos, y no entre el resto de tipos de adjetivo (demostrativos, ordinales o numerales).

Este hecho sería equivalente a las siguientes reglas DCG:

$A \rightarrow ASimple, Restricción, A$

$A \rightarrow ASimple.$

Como hemos comentado en anteriores ejemplos los adjetivos calificativos pueden aparecer yuxtapuestos y coordinados, lo cual puede ocurrir de modo simultáneo como por ejemplo en: *Pedro tiene el pelo [[castaño oscuro], corto y rizado]*. Esto lo conseguimos en SUG mediante las reglas de la Figura 46, donde el nombre del constituyente adjetivo será *adjetivo*, y en el caso que se trate de adjetivos calificativos, éstos podrán aparecer coordinados y yuxtapuestos simultáneamente.

```

coordinated (
  adjetivo (Numero, Genero, Tipo),
  adjetivoYuxtapuesto (Numero, Genero, Tipo),
  (Tipo == adjetivoCalificativo)
).
juxtaposition (
  adjetivoYuxtapuesto (Numero, Genero, Tipo),
  adjetivoSimple (Numero, Genero, Tipo),
  (Tipo == adjetivoCalificativo)
).

```

Figura 46. Yuxtaposición y coordinación de adjetivos calificativos.

Las ventajas que plantea este hecho respecto a la solución equivalente obtenida con las DCG son prácticamente las mismas que habíamos visto para el hecho *coordinated*, es decir, disminución de reglas gramaticales y eficiencia en el proceso de análisis.

4.3.4.3 HECHO “FUSION”

En la introducción del formalismo SUG dijimos que uno de los objetivos que nos planteábamos con esta extensión de las gramáticas DCG es el mejorar la eficiencia de la implementación de las DCG en Prolog reduciendo el número de reglas gramaticales y el número de símbolos no terminales. Anteriormente hemos observado cómo se reduce el número de símbolos no terminales por medio del operador opcional y el número de reglas por medio de los hechos *coordinated* y *juxtaposition*. En este apartado seguiremos con la

misma línea de reducir el número de reglas gramaticales mediante el hecho *fusion* que nos permite “fusionar” varias reglas con subconstituyentes comunes en una sola regla.

Este hecho *SUG* tiene la siguiente sintaxis: *fusion (regla₁, regla₂, ..., regla_n)*, donde:

- ◊ Cada *regla_i* es una regla gramatical *SUG* que se fusionarán en una sola regla.
- ◊ Estas *regla_i* han de cumplir una serie de características:
 - Han de tener la misma cabeza de regla. Es decir, cada *regla_i* ha de tener la siguiente forma: $\alpha \text{ ++> } \beta_1, \alpha \text{ ++> } \beta_2, \dots, \alpha \text{ ++> } \beta_n$.
 - Los cuerpos de las reglas β_i presentarán una estructura similar: un mismo número de constituyentes entre los cuales debe haber ciertos constituyentes comunes.

Observemos el ejemplo de la Figura 47, en el que supondremos que existe un sintagma nominal de tipo verbal formado o bien por un verbo en infinitivo, o bien por este verbo en infinitivo pero seguido por un sintagma nominal tipo sustantivo o un sintagma preposicional. Este tipo de sintagma nominal se podría expresar por medio de dos reglas *SUG*:

sn(tipoVerbal) ++> <<art>>, verbo(Infinitivo), <<sn(tipoSustantivo)>>.
sn(tipoVerbal) ++> <<art>>, verbo(Infinitivo), <<sp>>.

Estas reglas como se puede observar tienen la misma cabeza de regla: *sn (tipoVerbal)* y tienen el mismo número de constituyentes en el cuerpo de la regla: *tres* con dos constituyentes comunes: *<<art>>, verbo (Infinitivo)*. Por ello podremos fusionarlas en tan sólo una regla. El resultado de la fusión se muestra en esta figura y consiste en una representación simbólica del proceso que se realizaría al analizar un sintagma nominal de tipo verbal:

Se analizará un artículo opcional y un verbo en infinitivo obligatorio sólomente una vez, pasando después a analizar un sintagma nominal tipo sustantivo o un sintagma preposicional.

fusion (
(sn(tipoVerbal) ++> << art >>, verbo(Infinitivo), << sn(tipoSustantivo) >>),
(sn(tipoVerbal) ++> << art >>, verbo(Infinitivo), << sp >>)
).

Resultado simbólico de la fusión:

sn(tipoVerbal) ++> <<art>>, verbo(Infinitivo), { << sn(tipoSustantivo.)>> }
<<sp>>

Figura 47. Fusión de sintagmas nominales de tipo verbal.

Podemos observar otro ejemplo de fusión con las reglas gramaticales del constituyente *aposición* que puede aparecer como modificador del sustantivo en un sintagma nominal, por ejemplo: *Ese niño, el hijo de tu hermana, rompió la ventana*. Este constituyente puede estar formado por un sintagma nominal delimitado entre comas, o en el caso que este sintagma nominal aparezca como último constituyente de la frase, podría estar

delimitado al final por un punto, como sería el caso de *La ventana fue rota por ese niño, el hijo de tu hermana.*

```

fusion (
  (aposicion (Numero) ++> [';'], sn (Numero), [';']),
  (aposicion (Numero) ++> [';'], sn (Numero), ['.'])
).

Resultado simbólico de la fusión:

aposicion (Numero) ++> [;], sn (Numero), {
  [.]
  [;]
}

sn (Numero) ++> <<determinante (Numero) >>, sustantivo(Numero),
<<aposicion (Numero)>>, <<sp>>.

```

Figura 48. Fusión en las reglas gramaticales de la aposición.

Como ventajas de este hecho SUG es de resaltar que esta fusión se realizará a nivel interno²⁵, con lo que ganamos por un lado la claridad de ver todas las reglas a fusionar de una manera independiente, y por otro lado ganamos eficiencia de ejecución al conseguir reducir el número de reglas gramaticales. También se mejorará la eficiencia de la gramática, ya que de este modo se analizan los componentes de las reglas a fusionar una sola vez, además con la ventaja que en el resultado final de la implementación que presentaremos posteriormente en este trabajo de este formalismo SUG en Prolog, seremos capaces de conocer qué componentes han sido analizados y qué no. Por ejemplo, en la Figura 47 proponíamos la fusión de dos reglas SUG para el sintagma nominal de tipo verbal formado o bien por un verbo en infinitivo, o bien por este verbo en infinitivo pero seguido por un sintagma nominal tipo sustantivo o un sintagma preposicional. En el caso de analizar el siguiente sintagma nominal: *El jugar al fútbol* sabremos que ha sido analizado un artículo, el verbo infinitivo y un sintagma preposicional. O en el caso de analizar: *Fumar* sabremos que no ha aparecido ni el artículo, ni el sintagma nominal ni el preposicional.

4.3.4.4 HECHO "BASICWORD"

El hecho SUG *basicWord* permite definir los símbolos terminales de la gramática, como podrían ser los adjetivos, pronombres, nombres, verbos, conjunciones o preposiciones.

Este hecho presenta la siguiente sintaxis: *basicWord (C)*, donde:

- ◊ *C* será un símbolo terminal de la gramática con la misma estructura que los símbolos terminales de las DCG:
 - Un predicado con el nombre del constituyente,

²⁵ Por nivel interno nos referimos a la traducción que veremos posteriormente de estas reglas SUG a Prolog.

- y tantos argumentos como se considere necesario para acarrear la información contextual como el número, género, tipo semántico, etc.
- Por ejemplo: *nombre (Numero, Genero, Tipo)*.

En la Figura 49 se muestran varios ejemplos de hechos *basicWord* que definen diversos símbolos terminales de una gramática *SUG* como son las preposiciones, determinantes, conjunciones, pronombres, nombres, verbos, adjetivos y adverbios.

```

basicWord( preposicion ).
basicWord( determinante(Numero, Genero) ).
basicWord( conjuncion ).
basicWord( pronombre(Numero, Genero, Tipo) ).
basicWord( nombre(Numero, Genero, Tipo) ).
basicWord( verbo(Numero, Persona, Tiempo) ).
basicWord( adjetivo(Numero, Genero, Tipo) ).
basicWord( adverbio(Tipo) ).

```

Figura 49. Ejemplos de hechos *SUG* “*basicWord*”.

En los traductores de DCG a Prolog esta definición de los símbolos terminales de la gramática se realiza del siguiente modo:

```

adv(Tipo) → [ADV], {esAdv(ADV, Tipo)}.
nombre(Numero, Genero) → [N], {esNombre(N, Numero, Genero)}.

```

en la que los predicados *esAdv* y *esNombre* suponen el modo de implementar el diccionario o léxico de palabras del lenguaje sobre el que va a trabajar la gramática:

```

esAdv(acaloradamente, mod).
esAdv(ahi, sit).
esNombre(amigo, sing, masc).
esNombre(hijo, sing, masc).

```

O sea, por una parte presenta una sintaxis más compleja que la propuesta en *SUG*, y por otra parte esta representación tiene la desventaja de complicar su trabajo con diferentes diccionarios, ya que en la propia definición de los símbolos terminales se exige que por ejemplo los adverbios estén almacenados como hechos con nombre *esAdv*. Sin embargo tal y como mostraremos en el próximo apartado, en nuestro formalismo *SUG* separamos la declaración de símbolos terminales (hecho *basicWord*) de la definición del diccionario de palabras (que se realizará por medio del hecho *esPalabra*), lo cual permite aumentar la modularidad del sistema y permitir el trabajo con diferentes diccionarios o como veremos posteriormente permitir el trabajar sobre la salida de un etiquetador.

4.3.4.5 HECHO “*ESPALABRA*”

Por último, el hecho *SUG* *esPalabra* se emplea para especificar el diccionario de entradas léxicas o palabras del lenguaje. En las entradas léxicas de este diccionario se

definen las palabras del lenguaje junto con toda su información de tipo sintáctico, morfológico o semántico.

Este hecho presenta la siguiente sintaxis: *esPalabra (P, L)*, donde:

- ◊ *P* será la palabra.
- ◊ *L* será la lista de entradas léxicas de la palabra *P*.

Por ejemplo, la palabra *para* tendrá dos entradas léxicas: una en la que funciona como una preposición, y otra en la que funciona como verbo (forma imperativa del verbo *parar*). Otro ejemplo similar ocurriría con la palabra *abandono* que puede funcionar como sustantivo o como verbo. Ambas palabras se representarían mediante el hecho *esPalabra* del siguiente modo:

esPalabra(para, [preposicion, verbo(singular)]).
esPalabra(abandono, [nombre(singular), verbo(singular)]).

En la Figura 50 se muestra un fragmento de un posible diccionario *SUG* implementado por medio de este hecho *esPalabra*.

```

esPalabra( el, [determinante(singular)] ).
esPalabra( verde, [adjetivoSimple] ).
esPalabra( ratón, [nombre( singular, masculino, nohumano )]).
esPalabra( coche, [nombre( singular, masculino, inorganico)]).
esPalabra( comió, [verbo( singular, masculino, animal, comestible)]).
esPalabra( queso, [sustantivo( singular, masculino, comestible)]).
esPalabra( no, [adverbio(negacion)] ).
esPalabra( para, [preposicion, verbo(singular)] ).

```

Figura 50. Diccionario *SUG* implementado mediante hechos “*esPalabra*”.

Las ventajas de especificar separadamente el diccionario y los símbolos terminales ya las enumeramos en el anterior apartado. Además al almacenar cada entrada léxica de una misma palabra en un sólo hecho (en una lista) nos facilitaría su posterior implementación por medio de un *árbol de letras* o *trie* con el objetivo de mejorar la eficiencia de las operaciones de acceso al diccionario, tal y como explica Covington en [42] y en la implementación de este árbol de letras en Prolog que nosotros presentamos en [58].

4.3.5 Ejemplo de una gramática en notación *SUG*

En este apartado mostraremos una gramática completa en notación *SUG*, sobre la que se aplican todas las características explicadas anteriormente. Esta gramática tiene como

símbolo inicial *oracion*, que a su vez se constituye de la coordinación de cualquier tipo²⁶ de oraciones simples denotadas por el símbolo no terminal *oracionSimple*. Cada oración simple se compondrá de un sintagma nominal opcional y un sintagma verbal obligatorio, exigiéndose que ambos concuerden en número.

El sintagma nominal podrá ser de tipo sustantivo (*tipoSustantivo*) o de tipo pronombre (*tipoPronombre*), permitiéndose la coordinación de sintagmas nominales de cualquier tipo y número (no se exige que concuerden en número los sintagmas nominales coordinados), por ejemplo: *Pedro y yo*, o en *Los niños y ella*.

También se permitirá la coordinación, cuyo nombre será *sintagmaPreposicional*, de sintagmas preposicionales simples (*sintagmaPreposicionalSimple*), los cuales a su vez estarán formados por una preposición y un sintagma nominal.

<i>coordinated (oracion, oracionSimple, _)</i> .
<i>oracionSimple ++> <<sintagmaNominal (Numero, _)>>, sintagmaVerbal (Numero)</i>).

<i>coordinated (sintagmaNominal (_, _), sintagmaNominalSimple (_, _), _)</i> .
<i>sintagmaNominalSimple (Numero, tipoSustantivo) ++> <<determinante (Numero)>>, nombre (Numero), <<adjetivo (Numero, calificativo)>>, <<sintagmaPreposicional>></i> .
<i>sintagmaNominalSimple (Numero, tipoPronombre) ++> pronombre (Numero)</i> .

<i>coordinated (sintagmaPreposicional, sintagmaPreposicionalSimple, _)</i> .
<i>sintagmaPreposicionalSimple ++> preposicion, sintagmaNominal (_, _)</i> .

<i>sintagmaVerbal (Numero) ++> verbo (Numero), <<sintagmaNominal (_, _)>>, <<sintagmaPreposicional>></i> .

<i>juxtaposition (adjetivo (Numero, Tipo), adjetivoSimple (Numero, Tipo), Tipo == calificativo)</i> .

<i>basicWord (preposicion)</i> .
<i>basicWord (determinante (Numero))</i> .
<i>basicWord (conjuncion)</i> .
<i>basicWord (pronombre (Numero))</i> .
<i>basicWord (nombre (Numero))</i> .
<i>basicWord (verbo (Numero))</i> .
<i>basicWord (adjetivoSimple (Numero, Tipo))</i> .
<i>basicWord (adverbio (Tipo))</i> .

<i>esPalabra (el, [determinante (singular)])</i> .
<i>esPalabra (niño, [nombre (singular)])</i> .
<i>esPalabra (compró, [verbo (singular)])</i> .
<i>esPalabra (la, [determinante (singular),pronombre (singular)])</i> .
<i>esPalabra (pelota, [nombre (singular)])</i> .
<i>esPalabra (y, [conjuncion])</i> .
<i>esPalabra (jugó, [verbo (singular)])</i> .
<i>esPalabra (con, [preposicion])</i> .
<i>esPalabra (ella, [pronombre (singular)])</i> .

Figura 51. Ejemplo de gramática *SUG* con “*oracion*” como símbolo inicial.

²⁶ Recordamos que el tercer argumento del hecho *SUG* *coordinated* especifica una posible restricción a la coordinación, y en este caso la coordinación de oraciones no presenta ninguna restricción.

Los adjetivos calificativos, de nombre *adjetivo*, permitirán la yuxtaposición de adjetivos calificativos simples (*adjetivoSimple*). En el caso que el adjetivo sea de otro tipo, por ejemplo ordinal o numeral, no se permitirá la yuxtaposición.

Por simplicidad, sólo se ha tenido en cuenta la información referente al número, aunque según ya hemos comentado anteriormente es posible añadir más información como podría ser el género o información semántica. En la Figura 52 se muestra un ejemplo del análisis que se obtendría con la frase: *El niño compró la pelota y jugó con ella.*

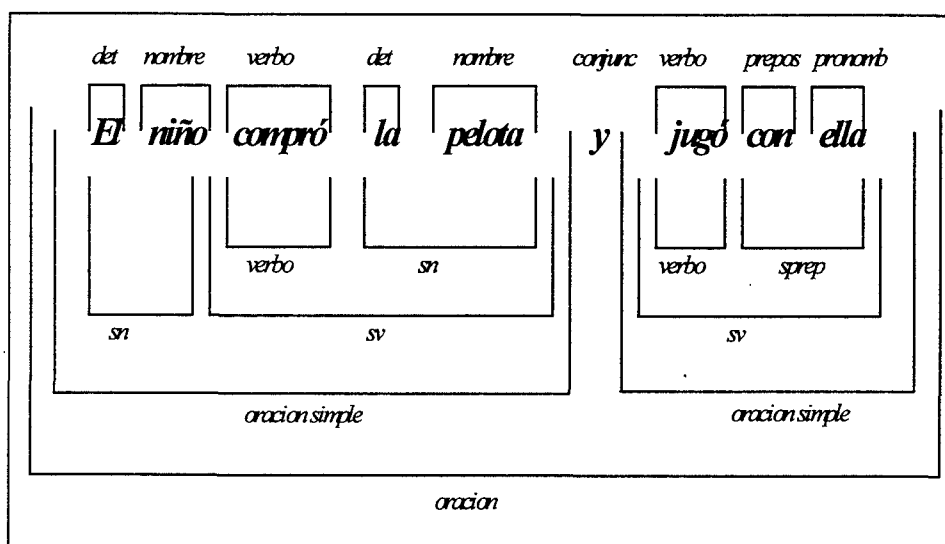


Figura 52. Ejemplo de frase aceptada por la gramática de la Figura 51.

4.4 Sistema SUP

En esta sección vamos a presentar el *analizador de reglas SUG* desarrollado por nosotros al que denominamos sistema *SUP* (*Slot Unification Parser*). Este analizador está desarrollado en Prolog, y sobre él pondremos en práctica el método de resolución de la anáfora que mostraremos posteriormente.

Este analizador SUP lo explicaremos en cuatro apartados. En el primero mostraremos el esquema de trabajo de SUP, en el que se podrá apreciar que se trata de un sistema modular en el que hay una serie de fases claramente diferenciadas que se verán en los siguientes tres apartados: traducción de reglas SUG a Prolog, la fase de análisis sintáctico y la de análisis semántico.

4.4.1 Esquema de trabajo del sistema SUP

El esquema de trabajo seguido por el sistema SUP se ha resumido en la Figura 53, y consiste en los siguientes puntos que detallaremos en los próximos apartados:

- ◆ Toda la información sintáctica de la frase se almacenará en una gramática en notación SUP.
- ◆ Esta gramática se traducirá a Prolog por medio del traductor que veremos en el próximo apartado, con lo que obtendremos un programa Prolog que aceptará como entrada la oración a analizar.
- ◆ El proceso de análisis sintáctico de esta oración se realizará en este programa Prolog. Como resultado de este análisis sintáctico el sistema devolverá una estructura de datos que almacenará toda la información (morfológica, sintáctica y semántica) necesaria para la resolución de problemas lingüísticos. A esta estructura de datos la llamamos *estructura de huecos*.
- ◆ Esta estructura de huecos la recibirá como entrada el módulo de resolución de problemas lingüísticos, en el que encuadraremos el algoritmo de resolución de la anáfora que aquí proponemos. Como salida de este módulo se devolverá una estructura de huecos modificada en la que se habrán eliminado estos problemas lingüísticos.
- ◆ El módulo de análisis semántico recibirá como entrada esta estructura de huecos modificada para obtener la interpretación semántica independiente del contexto.

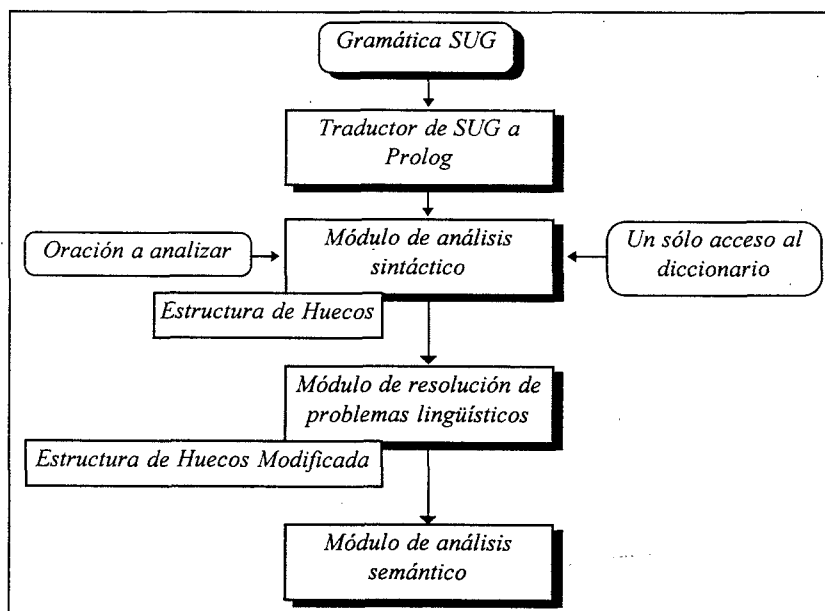


Figura 53. Esquema de trabajo del sistema SUP.

Es importante resaltar que este sistema SUP permite obtener sistemas de procesamiento del lenguaje natural modulares, en los que cada módulo alcanza un alto grado de independencia:

- ◇ La información sintáctica almacenada en la gramática *SUG* es independiente de la interpretación semántica que se realizará en el último módulo de análisis semántico, por lo que podríamos escoger el lenguaje de la fórmula lógica que prefiramos sin que esto afecte a la gramática *SUG*.
- ◇ Del mismo modo, también hay una gran independencia con el módulo del diccionario de entradas léxicas, al producirse tan sólo un acceso al diccionario antes de empezar la fase de análisis sintáctico. Esto nos permitirá trabajar con diferentes diccionarios o sobre la salida de un etiquetador sin que nos afecte al resto de módulos tal y como mostraremos posteriormente.
- ◇ El módulo de análisis sintáctico presenta también una alta independencia del resto de módulos. Actualmente *SUG* trabaja con una técnica de análisis descendente de izquierda a derecha con retroceso, pero no sería complejo el cambiar a una técnica ascendente, ya que el único objetivo es crear la correspondiente estructura de huecos.
- ◇ El módulo de resolución de problemas lingüísticos está estructurado de tal forma que permite el tratamiento modular de distintos fenómenos como podrían ser la extraposición de constituyentes, la elipsis o la anáfora, permitiéndose la interrelación entre ellos en casos en los que resulte beneficioso.

En los próximos apartados detallaremos cada uno de estos módulos.

4.4.2 Traductor de *SUG* a Prolog

Los motivos por los que se ha elegido Prolog como lenguaje de programación concuerdan con los esgrimidos por Covington en [42], por los que Prolog se puede considerar como un lenguaje adaptado para el procesamiento del lenguaje natural por los motivos que exponemos seguidamente. Prolog permite la construcción de estructuras de datos grandes y complejas. Esto facilita la representación de estructuras sintácticas y semánticas, entradas léxicas y similares. Además Prolog está diseñado para poder realizar representaciones del conocimiento sin dificultad y está construido a partir de un subconjunto de la lógica de primer orden, siendo relativamente sencillo el poder construir extensiones a esta lógica. También Prolog trae consigo un algoritmo de búsqueda: “*buscar primero en profundidad*”, que se puede usar en cualquier analizador, y asimismo incorpora la operación de *unificación*, la cual permite construir estructuras de datos paso a paso de tal modo que no importe el orden de esos pasos.

El traductor desarrollado trabaja de una forma similar a los traductores que convierten las reglas DCG en cláusulas Prolog. Estos traductores de DCG a Prolog existen en la mayoría de intérpretes o compiladores Prolog, y realizan esa transformación añadiendo dos argumentos extras a cada símbolo que no está entre llaves o corchetes. Por ejemplo, la regla DCG: $o \rightarrow sn, sv$, es traducida a la siguiente cláusula Prolog: $o(L1, L3) :- sn(L1, L2), sv(L2, L3)$, donde $L1$, $L2$ y $L3$ son las listas de palabras de la frase a analizar (por ejemplo al analizar la oración *Pedro lee un libro*, estas listas se instanciarían a los siguientes valores: $L1 = [Pedro, lee, un, libro]$, $L2 = [lee, un, libro]$ y $L3 = []$).

Nuestro traductor de reglas *SUG* a cláusulas Prolog funciona de un modo similar al de las DCG, y se ha probado en SICStus Prolog 2.1 y Arity Prolog 5.1. Su código Prolog se

muestra en el apéndice G. La diferencia radica en que se le ha añadido a cada subconstituyente un argumento extra (además de las listas), con el fin de formar lo que llamamos *estructura de huecos* o *slot structure* y que explicaremos en el próximo apartado. Dada por ejemplo la regla *SUG*: $o(\text{Numero}) \rightarrow sn(\text{Numero}), sv(\text{Numero})$, se verá traducida a Prolog del siguiente modo: $o(\text{Numero}, EHo, L1, L3) :- sn(\text{Numero}, EHsn, L1, L2), sv(\text{Numero}, EHsv, L2, L3)$.

4.4.3 Análisis sintáctico

Durante la fase de análisis sintáctico se utiliza el programa Prolog resultado de la traducción de las reglas gramaticales en notación *SUG*. Este programa utilizará una técnica de análisis descendente de izquierda a derecha con retroceso en un punto de fallo al inmediato anterior donde haya opciones a seguir distintas a la tomada anteriormente (algoritmo con retroceso o backtracking), es decir, el propio algoritmo de análisis de programas Prolog, pero mejorado en unos cuantos aspectos:

- ◆ El módulo de análisis sintáctico de SUP devolverá como salida una estructura de datos que aglutinará toda la información necesaria para la resolución de problemas lingüísticos. A esta estructura la llamaremos *estructura de huecos* y la explicaremos en el próximo apartado.
- ◆ Se realizará un sólo acceso al diccionario durante todo el proceso de análisis, hecho que explicaremos en el apartado 4.4.3.2, y que nos permitirá aplicar eficientemente algoritmos de poda en el análisis.

El motivo de utilizar este algoritmo descendente en lugar de otros, como pueden ser las técnicas de análisis sintáctico *ascendente* (la empleada en el trabajo de Palomar [127] con la implementación de las gramáticas Datalog Extendidas), ha sido únicamente por la sencillez de este algoritmo que nos permite la ejecución de una gramática *SUG* como cualquier programa Prolog, lo cual nos facilita su posterior integración con otros programas Prolog como pueden ser un sistema experto o una base de datos. La principal crítica que ha recibido este tipo de análisis descendente es el de ineficiencia, ofreciendo en algunos casos una complejidad exponencial. Tal y como mostraremos en el capítulo 6 hemos conseguido buenos resultados al utilizar las características propias de *SUG* (operadores opcionales y reducción del número de reglas) y algoritmos de poda que explicaremos en el apartado 4.4.3.2. Con ello conseguiremos complejidades inferiores a n^3 , siendo n el número de palabras de la frase.

4.4.3.1 ESTRUCTURA DE HUECOS DEVUELTA POR EL ANALIZADOR

Según hemos comentado anteriormente esta estructura de huecos aglutinará toda la información necesaria para la resolución de problemas lingüísticos. Esta información la obtiene automáticamente de cada regla de producción al incluir el resultado de la unificación de cada uno de los argumentos de cada constituyente: su información sintáctica, morfológica y semántica.

Una estructura de huecos es en realidad:

- ◆ Una estructura con functor²⁷ correspondiente al nombre del constituyente al que representa (*sn*, *sp*, *determinante*, etc.).
- ◆ Su primer argumento corresponde a otra estructura con nombre *conc* la cual incluye todos los argumentos del constituyente en la regla gramatical en que aparece (información morfológica y semántica: *Numero*, *Genero*, *TipoSemantico*, etc.).
- ◆ El segundo corresponde a la variable de la fórmula lógica final del constituyente, la cual se utilizará para ligar la variable en el caso que se resuelva un problema de elipsis o anáfora.
- ◆ Y los restantes argumentos corresponden a las estructuras de huecos de sus subconstituyentes.
- ◆ La información sintáctica depende de la posición que esta estructura de huecos ocupe en otras de nivel superior. Por ejemplo, la estructura de huecos de una oración declarativa: *oracion (conc(...), X, sn (...), sv (...))*, estará formada por una primera estructura de huecos que corresponderá a un sintagma nominal con función sintáctica de sujeto: *sn (...)*, y como segunda estructura de huecos la de un sintagma verbal con función de predicado: *sv (...)*.
- ◆ En esta estructura, el traductor dejará como variables Prolog sin instanciar (“_”) aquellos huecos que correspondan a componentes opcionales que no han sido analizados. De este modo, sabremos qué ha sido analizado en la oración y qué no (información de especial interés que hemos utilizado a la hora de resolver la elipsis [128] o la extraposición de constituyentes [61]). Otra ventaja de estas variables sin instanciar, es la facilidad con que nos permiten recuperar los elementos extrapuestos o elididos de un constituyente, puesto que la solución consistirá en una simple operación de unificación entre variables.

En la Figura 54 se muestra un ejemplo de la estructura de huecos del sintagma nominal *un perro*, en el que se puede observar cómo se incluye tanto la información morfológica (número y género) como la semántica (tipo semántico). También es de reseñar que la variable *X* almacenada en la estructura de huecos se utilizará en la fórmula lógica que se devuelve de la fase de análisis semántico, por ejemplo si el anterior sintagma nominal apareciese como sujeto en la siguiente frase: *Un perro ladró*, la fórmula lógica se correspondería con: *existe (X, perro (X), ladrar (X))*.

²⁷ Por *functor* entendemos el nombre de un predicado. Por ejemplo el predicado de aridad dos: *sn (N, G)* tendrá como functor a *sn* y dos argumentos: *N* y *G*.

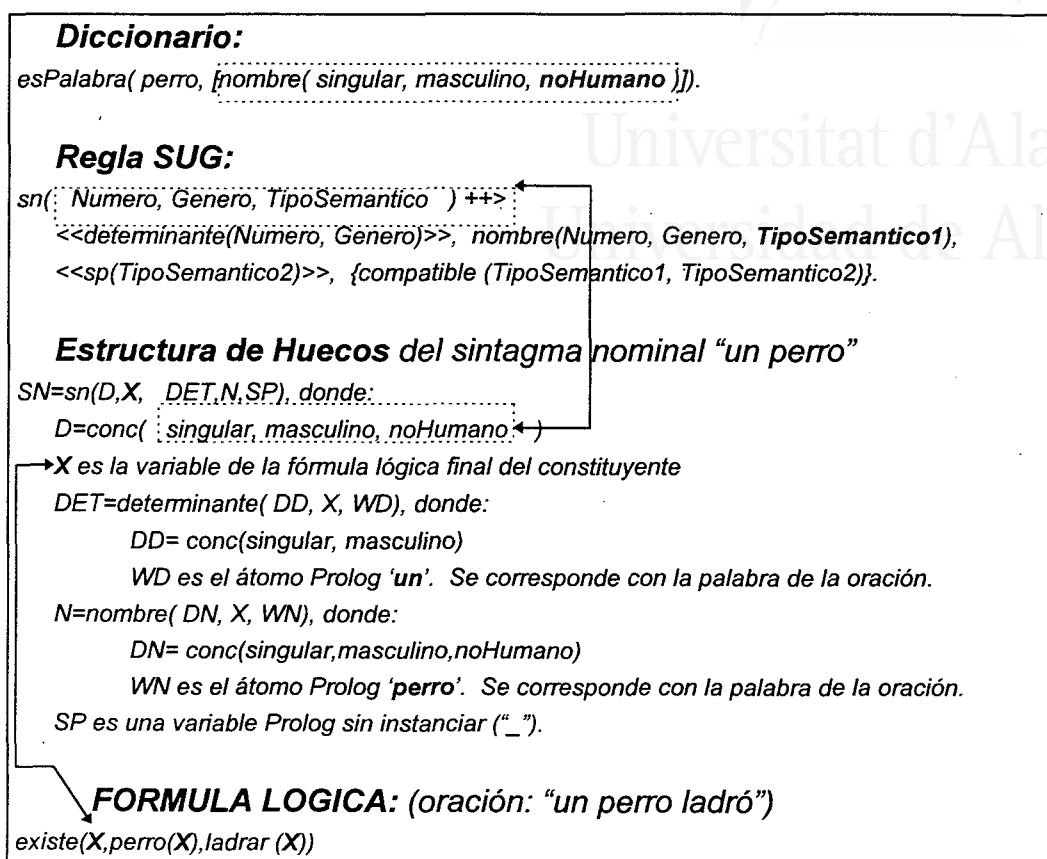


Figura 54. Estructura de huecos del sintagma nominal "un perro".

Con la intención de simplificar los restantes ejemplos de estructuras de huecos que mostremos en próximos ejemplos, no se incluirán los campos correspondientes a la estructura *conc* ni la variable *X* a menos que sea estrictamente necesario.

Respecto a la información semántica asociada a las estructuras de huecos, se corresponde a restricciones semánticas que se han incluido en el análisis sintáctico siguiendo el método IRSAS desarrollado por Moreno en [122][121]. Este método consistirá en añadir rasgos semánticos específicos para cada objeto que se registrarán en la entrada léxica de cada palabra usándose en todas las reglas de la gramática como un argumento más (al igual que la información morfológica). Los rasgos denotan las partes del universo a las cuales pertenecen los objetos, y constituirán una jerarquía de rasgos (en la Figura 75 de la página 161 se puede observar una jerarquía completa de rasgos). Todos estos rasgos se registran en la entrada léxica de cada palabra y se usarán en todas las reglas de la gramática como un argumento más (al igual que la información morfológica). De este modo se almacenarán automáticamente en la estructura de huecos de cada constituyente, pudiéndose utilizar tal y como se muestra en la Figura 54 para comprobar la compatibilidad entre dos determinados rasgos semánticos mediante el predicado Prolog: {compatible (TipoSemantico1, TipoSemantico2)}.

Para finalizar este apartado, en la Figura 55 se muestra la estructura de huecos simplificada obtenida en la coordinación de sintagmas preposicionales. Esta estructura se utilizará posteriormente en la resolución de problemas lingüísticos, concretamente para la

resolución de la denominada por Hirst en [80] *anáfora superficial numérica*, o sea, la anáfora producida por adjetivos ordinales o numerales (*Pedro dudó entre el libro y el disco, pero finalmente eligió [el segundo]*). Este tipo de anáfora se resuelve con esta estructura de huecos puesto que nos permite acceder a cualquier constituyente coordinado en el orden deseado. La condición que nos indicará que no hay más componentes coordinados es la de encontrar en los dos últimos argumentos de la estructura dos variables Prolog sin instanciar.

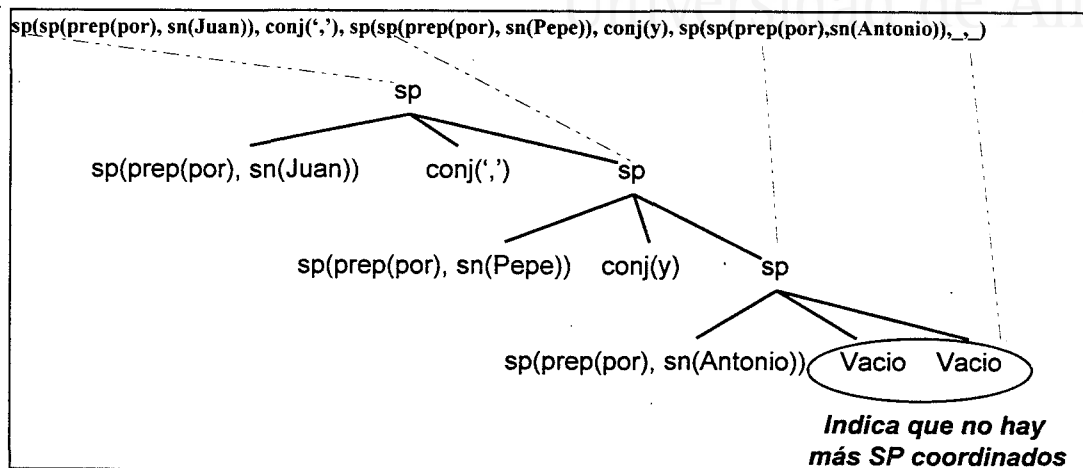


Figura 55. Ejemplo de la estructura de huecos obtenida en la coordinación.

4.4.3.2 REDUCIENDO EL NÚMERO DE ACCESOS AL DICCIONARIO

Es obvio que en una aplicación real de procesamiento del lenguaje natural se ha de trabajar con un diccionario de gran tamaño y de gran complejidad ante la gran cantidad de información que se ha de almacenar. Esta situación implica que se produzcan cuellos de botella al realizarse accesos repetidos al diccionario para cada uno de los análisis de una palabra. En SUP hemos intentado afrontar ese problema reduciendo el número de accesos al diccionario a tan sólo uno en toda la fase de análisis. Este acceso se producirá antes de empezar el análisis sintáctico de la oración, momento en el que se creará una lista de estructuras con functor *w* y aridad dos descrita en la Figura 56. En estas estructuras se almacena como primer argumento el símbolo terminal como un átomo Prolog, y como segundo argumento otra nueva lista de estructuras. En esta segunda lista cada uno de sus elementos consistirá en una estructura que determina una entrada léxica de la palabra. La cardinalidad de esta segunda lista determinará el número de diferentes entradas léxicas que posee la palabra, por ejemplo la palabra *abandono* podría tener dos acepciones: la primera como sustantivo y la segunda como verbo.

De este modo, cada vez que durante la fase de análisis se necesite información referente a una determinada palabra, en lugar de acceder al diccionario consultaremos esta lista de entradas léxicas de la palabra, es decir, nunca más accederemos al diccionario después de crear esta lista.

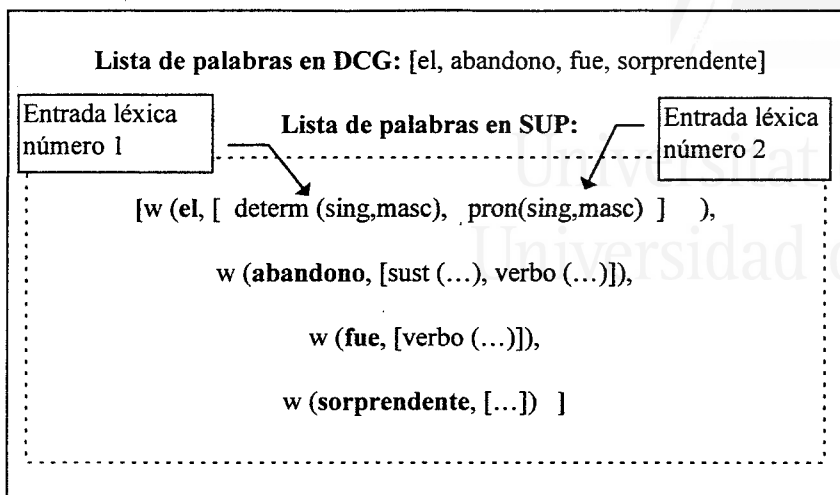


Figura 56. Comparación entre la lista de palabras empleada en SUP y la utilizada en DCG.

Así, el funcionamiento del analizador SUP respecto al de las DCG es bastante diferente, puesto que en las DCG se crea una simple lista de palabras, tal y como se muestra en la Figura 56, con lo que cada vez que se necesita información acerca de una palabra se produce un nuevo acceso al diccionario.

Además, al mantener en memoria esta lista de entradas léxicas nos facilita la implementación de un algoritmo de poda con el que mejorar la eficiencia del análisis sintáctico de la frase. Esta poda consistirá en estudiar las siguientes palabras de la frase a analizar en aquellos puntos donde hay varias alternativas posibles en lugar de realizar una búsqueda secuencial por cada una de las reglas gramaticales. En función del tipo morfológico de las restantes palabras a analizar se decidirá el camino a seguir rechazando las demás alternativas.

Podemos encontrar un ejemplo de la aplicación de esta poda en las oraciones impersonales (oraciones que no tienen sujeto), en las que la primera palabra de la oración siempre es un verbo. En la gramática *SUG* de la Figura 51 el constituyente oración tiene un sintagma nominal opcional como sujeto para analizar tanto oraciones personales como impersonales:

oracionSimple ++> <<*sintagmaNominal* >>, *sintagmaVerbal*.

En caso que la primera palabra de la oración sea un verbo (que no está ni en infinitivo, ni en participio ni en gerundio) podríamos concluir que no se va a encontrar ningún sintagma nominal. Tal y como está la gramática en esta figura, el analizador tendría que revisar todas las reglas gramaticales correspondientes al sintagma nominal con la ineficiencia correspondiente. Pero aplicando el algoritmo de poda podremos evitar este recorrido tal y como se muestra en la Figura 57. La modificación realizada en las reglas gramaticales de los sintagmas nominales también se expone en esta figura. Se puede observar que se ha distinguido entre *sintagma nominal simple* y *sintagma nominal simple podado*, incluyéndose en el primero el algoritmo de poda por medio del predicado del mismo nombre: *poda (TipoDeSN)*. Este predicado de poda estudiará la lista de palabras que quedan por analizar, fracasando en caso que se pueda concluir que a continuación no venga

un sintagma nominal (por ejemplo que venga una preposición o el final de la frase), y en caso contrario devolverá el tipo de sintagma nominal (tipo sustantivo o pronombre) en el argumento *TipoDeSN*. En la Figura 84 de la página 180, se muestra el contenido del predicado *poda*, en la implementación realizada de una gramática *SUG* de oraciones declarativas.

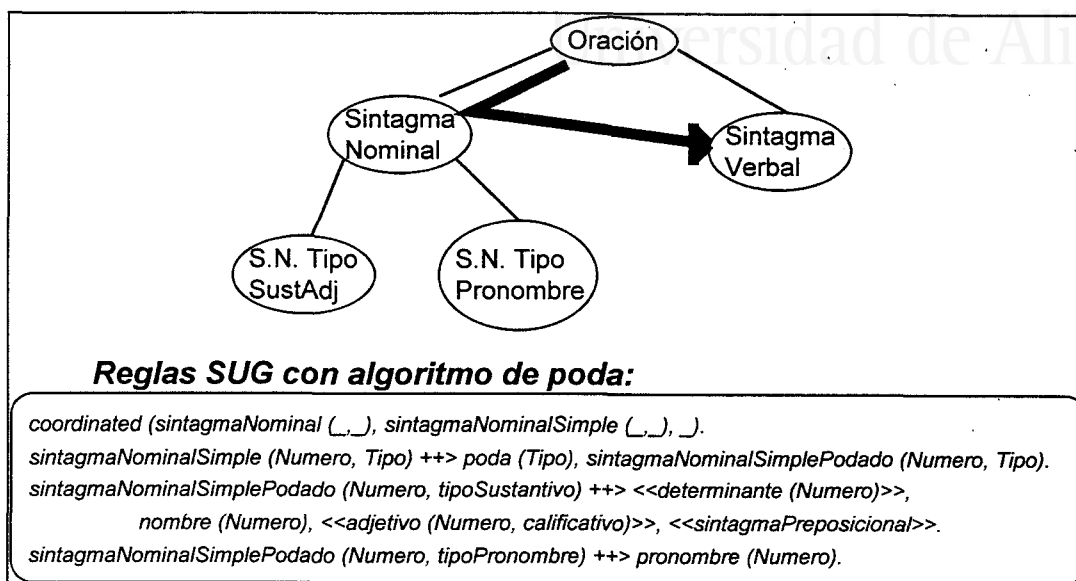


Figura 57. Proceso de poda con la oración impersonal: "Llueve mucho".

En la Figura 58 se muestra otro ejemplo de la poda realizada en los sintagmas nominales en la oración *Ellos lo rompieron*. En el sintagma nominal con función de sujeto, sabremos por anticipado que se trata de un sintagma nominal tipo pronombre, ya que la palabra que viene a continuación es un pronombre personal con función de sujeto, con lo que podríamos evitar la comprobación del sintagma nominal tipo sustantivo.

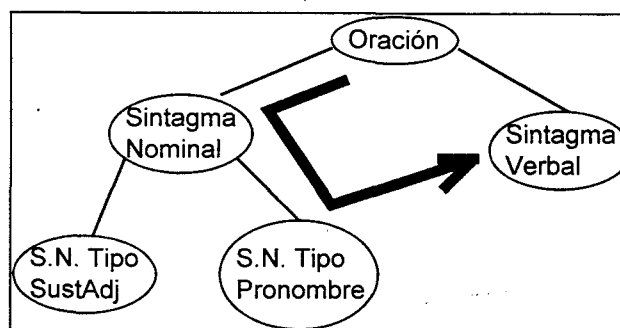


Figura 58. Poda realizada en la oración: "Ellos lo rompieron".

Este algoritmo de poda se implementa de manera muy eficiente en el analizador SUP, gracias a que mantenemos en memoria la lista de entradas léxicas de la palabra que nos permite conocer los tipos de palabras a analizar sin realizar sucesivos accesos al diccionario.

Otra de las ventajas de mantener esta lista de entradas léxicas de cada palabra es la posibilidad de resolver determinados problemas léxicos antes de comenzar la fase de análisis sintáctico, como el descomponer palabras, por ejemplo las de tipo verbo más pronombre personal (*dime, cántame*, etc.). Otro problema factible a tratar es el caso inverso en el que se podrían agrupar varias palabras que actúan como una sola (por ejemplo: *debajo de*). Y finalmente también se podría tomar una determinación con las palabras que no están almacenadas en el diccionario, las cuales se pueden rechazar o asignar una determinada categoría por defecto.

Otra ventaja adicional de realizar un sólo acceso al diccionario antes de entrar en la fase de análisis sintáctico, es el de facilitar el trabajo con diferentes tipos de diccionario. Incluso cada diccionario podría tener diferente modo de almacenar la información de cada palabra, tal y como se expondrá en el apartado 5.2, en el que trabajaremos sobre la salida de un etiquetador. Así conseguimos un alto grado de independencia entre el diccionario y las reglas gramaticales, ya que en el proceso de creación de la lista de entradas léxicas se creará un interfaz adecuado entre ambos módulos, el cual se ejecutará tan sólo una vez durante todo el análisis de cada frase.

4.4.4 Análisis semántico

En este apartado vamos a describir el proceso de análisis o interpretación semántica de la frase dentro del sistema SUP. Este proceso consiste en la obtención del significado independiente del contexto de la frase por medio de un lenguaje formal que llamaremos *fórmula lógica*. El lenguaje formal que vamos a utilizar para la construcción de la fórmula lógica es el del Cálculo de Predicados de Primer Orden (CPPO), y concretamente nos basaremos en el especificado por Covington en [42] y Moreno en [120] ya descrito en la subsección 4.1.3, realizando tan sólo las extensiones que consideremos necesarias.

4.4.4.1 PROCESO DE ANÁLISIS SEMÁNTICO EN SUP

Según comentamos en el apartado 4.4.1 en el que mostrábamos el esquema de trabajo de nuestro sistema SUP, hemos separado totalmente la especificación de la información sintáctica de la semántica, e igualmente las fases de análisis sintáctico de la de análisis semántico. Al realizar esta separación conseguimos la ventaja de aumentar la modularidad del sistema permitiendo una total independencia entre ambos módulos, lo que nos permitirá utilizar el lenguaje de fórmula lógica que se considere adecuado sin que esto influya en el módulo de análisis sintáctico.

Esta separación entre ambos módulos no existe en las DCG ya que se codifica en la misma gramática la fórmula lógica de cada constituyente como un argumento más tal y como se muestra en la Figura 59. Además se hace necesario el añadir esta información también en el léxico, con lo que aumentamos la dependencia con otro módulo más (el diccionario).

Gramática DCG con información de la fórmula lógica:	
$oracion(Sem)$	$\rightarrow sn(X, Scope, Sem), sv(X, Scope).$
$sn(X, Scope, Sem)$	$\rightarrow d(X, Rest, Scope, Sem), n(X, Rest).$
$sn(X, Scope, Sem)$	$\rightarrow d(X, Rest\&Rest1, Scope, Sem), n(X, Rest), a(X, Rest1).$
$sn(X, Sem, Sem)$	$\rightarrow prop(X).$
$sv(X, Sem)$	$\rightarrow v(X, Y, S0), sp(Y, S0, Sem).$
$sv(X, Sem)$	$\rightarrow v(X, Y, S0), sn(Y, S0, Sem).$
$sv(X, Sem)$	$\rightarrow v(X, Y, S0), sn(Y, Se1, Sem), sp(Z, S0, Se1).$
$sp(X, Y, Sem)$	$\rightarrow prep(X, Y, S0), sn(X, S0, Sem).$
Léxico con información de la fórmula lógica:	
$n(n(X, manzana(X)))$	$\rightarrow [manzana]$
$n(n(X, pera(X)))$	$\rightarrow [pera]$
$d(d(X, Y, Z, existe(X, Y, Z)))$	$\rightarrow [una]$
$d(d(X, Y, Z, cada(X, Y, Z)))$	$\rightarrow [cada]$
$v(v(X, Y, comer(X, Y)))$	$\rightarrow [come]$
$v(v(X, Y, Z, dar(X, Y, Z)))$	$\rightarrow [da]$
$v(v(X, pasear(X)))$	$\rightarrow [pasea]$
$v(v(X, Y, pasear(X, Y)))$	$\rightarrow [pasea]$
$nprop(nprop(juan))$	$\rightarrow [juan]$
$nprop(nprop(maria))$	$\rightarrow [maria]$
$a(a(X, roja(X)))$	$\rightarrow [roja]$
$a(a(X, grande(X)))$	$\rightarrow [grande]$

Figura 59. Especificación de la fórmula lógica en una DCG.

En SUP esta fórmula lógica se consigue directamente a partir de la estructura de huecos generada por el módulo de análisis sintáctico, después de haberse resuelto los problemas lingüísticos (elipsis, anáfora, etc.) en el módulo de resolución de estos problemas. De modo resumido el algoritmo que se sigue para obtener estas fórmulas lógicas consiste en recorrer la estructura de huecos obteniendo sus constituyentes principales (sintagmas nominales, preposicionales y verbales) para calcular su fórmula lógica, y en una posterior fase se unen cada una de estas fórmulas lógicas parciales en una fórmula lógica final conjunta.

En la Figura 60 se muestra el algoritmo seguido para calcular la fórmula lógica de una oración declarativa coordinada. El predicado que calcula esta fórmula se llama *fOracion* el cual recibe como entrada la estructura de huecos *EHO* y devuelve la fórmula lógica en *FL*. En este predicado, en primer lugar se extraen²⁸ de *EHO* mediante el predicado *componentesOracionCoordinada* las estructuras de huecos de la primera oración simple *EHOOracionSimple*, la conjunción *EHConjuncion* y la del resto de oraciones coordinadas *EHRestoDeOraciones*. Calculará la fórmula lógica de esa primera oración simple en *FLOS*

²⁸ En el sistema SUP habrá un predicado Prolog para extraer los componentes de cada constituyente de la gramática. Cada uno de estos predicados recibirá como parámetro la estructura de huecos del constituyente devolviendo cada uno de sus constituyentes. El nombre de estos predicados estará formado por la palabra *componentes* seguida del nombre del constituyente. Por ejemplo para el sintagma verbal tendremos el predicado *componentesSintagmaVerbal*.

llamando al predicado *flOracionSimple*²⁹ y en función del tipo de conjunción *EHConjuncion* la conectará de modo adecuado con el resto de oraciones coordinadas.

Siguiendo en la misma figura, la fórmula lógica de una oración simple se obtendrá mediante el predicado *flOracionSimple* que recibirá la estructura de huecos en *EHOS* y devolverá su fórmula lógica en *FLOS*. En primer lugar, mediante el predicado *componentesOracionSimple* se extraen las estructuras de huecos del sujeto *EHSujeto* y del sintagma verbal *EHSintagmaVerbal*. A continuación se calculan las fórmulas lógicas de ambos constituyentes mediante el predicado *flSujeto* en *FLSujeto* junto con sus cuantificadores en *CuantificadorSujeto*, y mediante *flSintagmaVerbal* en *FLVerbo*, *FLOD*, *FLOI*, *FLCC* y *FLNegacion* junto con sus cuantificadores en *CuantificadorOD*, *CuantificadorOI* y *CuantificadorCC*. La fórmula lógica final que se devuelve tendrá la siguiente estructura:

$$\text{CuantificadorSujeto (CuantificadorOD (CuantificadorOI (CuantificadorCC (FLCC (FLNegacion (FLVerbo (FLSujeto, FLOD, FLOI)))))))))$$

en la que se puede apreciar que los cuantificadores presentan la estructura ya comentada anteriormente: *cuantificador (variable, restricción de proposición, cuerpo de proposición)* anidándose de manera sucesiva en la parte más externa de la fórmula lógica final.

Por ejemplo, para la frase *Un niño compró un regalo a una niña* tendríamos:

- ◇ Como estructura de huecos de la oración simple $EHOS = o (sn (un\ niño), sv (v (compró), sn (un\ regalo), sp (a\ una\ niña)))$ ³⁰.
- ◇ Como estructura de huecos del sujeto $EHSujeto = sn (un\ niño)$.
- ◇ Como estructura de huecos del sintagma verbal $EHSintagmaVerbal = sv (v (compró), sn (un\ regalo), sp (a\ una\ niña))$.
- ◇ Como fórmula lógica del sujeto: $FLSujeto = X$ y como cuantificador³¹: $CuantificadorSujeto = existe (X, niño (X), _)$.
- ◇ Como fórmula lógica del sintagma verbal:
 - $FLVerbo = comprar (FLSujeto, FLOD, FLOI)$.
 - $FLOD = Y$ y como cuantificador: $CuantificadorOD = existe (Y, regalo (Y), _)$.

²⁹ En el sistema SUP habrá un predicado Prolog para calcular la fórmula lógica de cada constituyente de la gramática. Cada uno de estos predicados recibirá como parámetro la estructura de huecos del constituyente devolviendo la fórmula lógica del mismo y los posibles cuantificadores que haya encontrado. El nombre de estos predicados estará formado por la inicial *fl* seguida del nombre del constituyente. Por ejemplo para el sintagma verbal tendremos el predicado *flSintagmaVerbal*.

³⁰ Para intentar simplificar los ejemplos, las estructuras de huecos se muestran abreviadas, debiendo entenderse la estructura *sn (un niño)* como *sn (det (un), n (niño))* además de no incluir los argumentos referentes a la variable y la estructura *conc* descritas en el apartado 4.4.3.1.

³¹ El tercer argumento del cuantificador que aparece con el símbolo “_” se corresponderá con una variable Prolog sin instanciar que nos permitirá anidar los cuantificadores del modo visto anteriormente.

- $FLOI = Z$ y como cuantificador: *CuantificadorOI* = existe (Z , niña (Z), _
).
- ◇ Finalmente la fórmula lógica de la oración simple: $FLOS = \text{existe } (X, \text{niño } (X), \text{existe } (Y, \text{regalo } (Y), \text{existe } (Z, \text{niña } (Z), \text{comprar } (X, Y, Z))))))$.

```

*****
Recibe como entrada la estructura de huecos de la oración declarativa analizada: EHO
Devolviendo su fórmula lógica en FL
*****/

Predicado fOracion (EHO, FL)
componentesOracionCoordinada (EHO, EHOOracionSimple, EHConjuncion, EHRestoDeOraciones)
fOracionSimple (EHOOracionSimple, FLOracionSimple)
Si existe (EHConjuncion)
    fOracion (EHRestoDeOraciones, FLRestoDeOraciones)
Si EHConjuncion == 'y'
    FL = FLOracionSimple & FLRestoDeOraciones
Sino
    FL = FLOracionSimple # FLRestoDeOraciones
FSi
Sino
    FL = FLOracionSimple
FSi
FPredicado
*****

Recibe como entrada la estructura de huecos de una Oración Simple: EHOS
Devolviendo su Fórmula Lógica en FLOS
*****/

Predicado fOracionSimple (EHOS, FLOS)
componentesOracionSimple (EHOS, EHSujeto, EHSintagmaVerbal)
fSujeto (EHSujeto, FLSujeto, CuantificadorSujeto),
fSintagmaVerbal (EHSintagmaVerbal, FLVerbo, FLOD, FLOI, FLCC, FLNegacion,
    CuantificadorOD, CuantificadorOI, CuantificadorCC)
FLOS = CuantificadorSujeto (CuantificadorOD (CuantificadorOI (CuantificadorCC (
    FLCC (FLNegacion (FLVerbo (FLSujeto, FLOD, FLOI)))))))
FPredicado

```

Figura 60. Algoritmo para el cálculo de la fórmula lógica.

4.4.4.2 ESPECIFICACIÓN DEL LENGUAJE DE LA FÓRMULA LÓGICA

El sentido de las palabras servirán como *átomos* o *constantes* de la representación. Estas constantes serán clasificadas en función de los tipos de cosas que describen. Por ejemplo, las constantes que describen objetos en el mundo, incluyendo objetos abstractos tales como eventos o situaciones, son llamados *términos*. Las constantes que describen relaciones y propiedades son llamados *predicados*. Una *proposición* en el lenguaje está formado de un predicado seguido por un número apropiado de términos, con sus argumentos: predicados con un único argumento se llamarán *predicados unarios*, predicados con dos argumentos serán *predicados binarios*, y finalmente los predicados con n argumentos constituirán los *predicados n-arios*.

Las proposiciones complejas se construyen usando *conectivas lógicas*. El Cálculo de Predicados de Primer Orden (CPPO) contiene conectivas como la *disyunción* (\vee) o la *conjunción* (\wedge ó también $\&$).

Con proposiciones simples sólo definimos subconjuntos limitados, normalmente oraciones de un sólo verbo. Para oraciones más complejas, deberemos definir construcciones semánticas adicionales, como por ejemplo los *cuantificadores*. En el CPPO hay dos cuantificadores: *Todo* (\forall) y *Existe* (\exists), pero en cualquier idioma es necesario añadir más cuantificadores como por ejemplo: *muchos*, *cada*, *algún*, etc.

En los próximos subapartados se expone una relación de reglas o hipótesis sobre los detalles de representación de las fórmulas lógicas de las oraciones y de sus componentes. Se presentarán en primer lugar las hipótesis relativas a las conectivas lógicas admitidas en el lenguaje de fórmula lógica, a continuación los elementos que introducen la cuantificación, después la representación elegida para la coordinación de constituyentes, y por último se describirá la representación de cada constituyente de la gramática. Esta especificación del lenguaje de la fórmula lógica se llevará a cabo sobre la gramática *SUG* de oraciones declarativas descrita en el apéndice A. Es importante resaltar que el lenguaje elegido para la fórmula lógica es fácilmente modificable, siendo nuestro objetivo el mostrar cómo se puede llevar a cabo, y no el definir un lenguaje único y válido para todo lingüista.

4.4.4.2.1 Oraciones declarativas

Recordemos que en las oraciones declarativas se afirma algún hecho como cierto, distinguiéndose de las oraciones interrogativas, las cuales se realizan con la intención de obtener una respuesta atendiendo a la semántica de dicha oración. Dentro de las oraciones interrogativas, nos podemos encontrar con diversos tipos como las interrogativas de cierto-falso, las interrogativas de tipo cantidad o las interrogativas de tipo general. Todos estos tipos de oraciones indican diferentes relaciones entre el hablante y el contenido de la proposición de la frase.

Para distinguir entre esos tipos de frases, todas las fórmulas lógicas de las oraciones declarativas vendrán incluidas dentro del único argumento del predicado unario *decl(S)* donde *S* es la fórmula lógica de los constituyentes de la oración. De este modo las fórmulas de oraciones interrogativas de cierto-falso, podrían especificarse dentro del predicado *siNo(S)*, eligiéndose otros predicados para el resto de tipos de frases.

4.4.4.2.2 Conectivas lógicas

Las conectivas lógicas del lenguaje que utilizaremos para construir las fórmulas lógicas son las siguientes:

- ◆ Escogeremos $\&$ para la conectiva lógica *and*.
- ◆ Tomaremos $\#$ para la conectiva *or*.
- ◆ Y finalmente *no (...)* corresponderá a la conectiva lógica *not*.

4.4.4.2.3 Fórmula lógica de los cuantificadores

Dotaremos de la siguiente estructura a todo cuantificador: *cuantificador (variable, restricción de proposición, cuerpo de proposición)*. Hemos establecido los siguientes cuantificadores:

- ◆ Para el caso de encontrarnos con artículos determinados (*el, la, las, etc.*), adjetivos posesivos (*mi, tu, su, etc.*), demostrativos (*ese, este, etc.*) o indefinidos (*algún, alguna, etc.*) dentro del sintagma nominal, lo representaremos por medio del cuantificador *def (variable, restricción de proposición, cuerpo de proposición)*. De este modo *El perro* se traduciría a *def (X, perro (X), ...)*.
- ◆ En caso de aparecer artículos indeterminados singulares como *un, uno* o *una*, se traduciría por *existeUn (variable, restricción de proposición, cuerpo de proposición)*. Por ejemplo, *Un perro* se transformaría en *existeUn (X, perro (X), ...)*.
- ◆ Para los artículos indeterminados plurales como *unos* o *unas*, su cuantificador sería *existeVarios (variable, restricción de proposición, cuerpo de proposición)*. Por ejemplo, *Unos perros* se transformaría en *existeVarios (X, perro (X), ...)*.
- ◆ Para los adjetivos indefinidos *todo, toda, todos, todas* y *cada* se les asociará el siguiente cuantificador: *todo (variable, restricción de proposición, cuerpo de proposición)*. Así *Todos los perros* se pondría *todo (X, perro (X), ...)*.
- ◆ Con los adjetivos numerales u ordinales como *dos, tres* o *primero* se corresponderán con el cuantificador *existe (adjetivo, variable, restricción de proposición, cuerpo de proposición)*. Por ejemplo, *Dos perros* se convertiría en *existe (dos, X, perro (X), ...)*.
- ◆ Y finalmente los comparativos del tipo *adverbioDeComparación + adjetivoNumeral (más de cuatro, menos de cuatro, etc.)* se convertirían en el cuantificador *existe (adverbioDeComparación, adjetivoNumeral, variable, restricción de proposición, cuerpo de proposición)*. Como en el caso de *Más de cuatro perros*, con su equivalente *existe (más, cuatro, X, perro (X), ...)*.

Para los casos en que aparezcan varios cuantificadores en la misma oración, estableceremos la siguiente hipótesis sobre la precedencia o ámbito de los mismos: la cuantificación del sujeto domina a la cuantificación introducida por los complementos del verbo, y la cuantificación introducida por el complemento de un nombre domina a la cuantificación introducida por dicho nombre.

En el corpus de frases utilizado para la comprobación de esta gramática hemos encontrado casos en que los sintagmas nominales no van determinados por un cuantificador, como por ejemplo el sintagma nominal que aparece dentro del sintagma preposicional *De color negro*. Para estos casos hemos querido distinguirlos de los demás en los que sí van precedidos por un cuantificador, al no asignarle ninguno de los cuantificadores enumerados anteriormente. De este modo, la oración *Pepe fuma puros*, se traduciría en *decl (fumar (Pepe, puros (X)))* para distinguirla de *Pepe fuma dos puros* que sería *decl (existe (dos, X, puro (X), fumar (Pepe, X)))*.

4.4.4.2.4 Fórmula lógica de la coordinación

Tal y como ya comentamos anteriormente pueden distinguirse las siguientes interpretaciones de los plurales relacionados con la coordinación:

- ◆ En primer lugar el *plural colectivo* en el que la propiedad del predicado debe aplicarse al conjunto entero de entidades afectadas, como en la oración *Juan y María se encontraron*, la cual se representaría como: *encontrarse (Juan, María)*.
- ◆ En segundo lugar el *plural distributivo* en el que la propiedad debe distribuirse entre todas las combinaciones posibles. Por ejemplo, *Juan y María saben Latín y Griego*, cuya representación sería: *saber (Juan, Latín) ∧ saber (María, Latín) ∧ saber (Juan, Griego) ∧ saber (María, Griego)*.
- ◆ Y finalmente un *plural respectivo*, el cual debe distribuirse apropiadamente. En la oración *Juan y María saben Latín y Griego respectivamente* la representación asociada sería: *saber (Juan, Latín) ∧ saber (María, Griego)*.

En la representación de la fórmula lógica hemos resuelto la coordinación indicando de forma explícita los elementos coordinados, sustituyendo la conjunción por la conectiva lógica correspondiente (por ejemplo, la conjunción *y* se sustituirá en la fórmula por la conectiva *&*, y la conjunción *o* por *#*). De este modo la oración *Juan, Antonio y Pepe compraron un juguete* se vería transformada a la fórmula lógica *decl (existeUn (X, juguete (X), comprar ((Juan & Antonio & Pepe), X)))*. Y la oración *Ana compró un perro o un gato* se transformaría a *decl (existeUn (X, perro (X), existeUn (Y, gato (Y), comprar (Ana, X # Y)))*.

Siguiendo esta estrategia de representación de la coordinación de constituyentes, trataríamos con los anteriores casos de plurales producidos por la coordinación de la siguiente manera:

- ◆ El plural colectivo de la frase *Juan y María se encontraron* lo representaríamos como: *decl (encontrar ((Juan & María), (Juan & María)))*.
- ◆ En cuanto al plural distributivo de *Juan y María saben Latín y Griego*, sería: *decl (saber ((Juan & María), (Latín & Griego)))*, dejándose para una posterior fase de interpretación contextual la distribución entre los distintos elementos coordinados.
- ◆ Y finalmente el plural respectivo *Juan y María saben Latín y Griego respectivamente*, se distinguiría del anterior caso distributivo por la presencia del adverbio *respectivamente*, eligiendo como representación asociada: *decl (saber (Juan, Latín) & saber (María, Griego))*.

Además de la coordinación de constituyentes *SUG* también trata con el caso especial de coordinación que se da sin necesidad de conjunciones denominado *yuxtaposición*. El ejemplo típico de yuxtaposición es el que ocurre con los adjetivos calificativos, por el que un número indefinido de adjetivos puede modificar a un nombre siempre que su significado sea coherente. Para estos casos, la fórmula lógica elegida para representar la yuxtaposición de constituyentes consistirá en la fórmula de cada uno de los constituyentes unida por el operador lógico *&*. Por ejemplo, para la oración *Pedro se ha comprado un coche azul verdoso brillante* se representaría por medio de la fórmula *decl (existeUn (X, coche (X) & azul (X) & verdoso (X) & brillante (X), comprar (Pedro, X)))*.

4.4.4.2.5 *Fórmula lógica de los verbos y sintagmas verbales*

Los verbos son predicados n -arios en los que n depende del número de argumentos del verbo. Es importante resaltar que el mismo verbo puede tener diferente número de argumentos, como *romper*, el cual puede aparecer en diferentes oraciones con distintas aridades: *Juan rompió la ventana con el martillo*; *Juan rompió la ventana* y *La ventana se rompió*. Así podríamos tener distintas aridades de un mismo verbo, tal y como lo describe Allen en [3]:

- ◆ Con aridad tres en: $def(X, ventana(X), def(Y, martillo(Y), romper(Juan, X, Y)))$.
- ◆ con aridad dos en: $def(X, ventana(X), romper(Juan, X))$.
- ◆ Y finalmente con aridad uno en: $def(X, ventana(X), romper(X))$.

Para representar estas diferentes aridades de cada verbo lo hemos resuelto asignando a cada verbo una aridad fija, e indicando explícitamente en cada uno de los argumentos si han aparecido o no en la frase. De este modo, las oraciones anteriores las representaríamos como:

- ◆ $def(X, ventana(X), def(Y, martillo(Y), romper(Juan, X, Y)))$
- ◆ $def(X, ventana(X), romper(Juan, X, noHayCC))$
- ◆ $def(X, ventana(X), romper(X, noHayOD, noHayCC))$

Hemos utilizado esta representación en lugar de la propuesta por Allen, puesto que la consideramos más adecuada para representar los roles de agente, tema e instrumento de cada sintagma nominal. De este modo podríamos distinguir entre los significados de las frases: *La ventana se rompió* y *Juan la rompió*, de tal modo que consideraríamos que *la ventana* ocupa el rol de tema, es decir, lo que se rompió, y *Juan* el rol de agente, o sea, el que realiza la acción. Si la oración está bien formada el sintagma nominal con función de sujeto será el agente de la oración, por lo que distinguiremos entre ambas oraciones de la siguiente manera: $def(X, ventana(X), romper(noHaySUJ, X, noHayCC))$ y $def(X, pronombre(X), romper(Juan, X, noHayCC))$. De este modo, siempre se colocaría como primer argumento del verbo el sintagma nominal con rol de agente, en el segundo el que tenga rol de tema y en los restantes los complementos circunstanciales que tengan diferentes roles como el de instrumento o lugar. De este modo se ha fijado el número de argumentos del verbo a tres, los cuales se corresponderán en este orden al sujeto, objeto directo y objeto indirecto. En caso que no aparezcan estos modificadores se colocará en su lugar las constantes *noHaySuj*, *noHayOD* o *noHayOI*. En los sucesivos ejemplos que mostremos sobre fórmulas lógicas, tan sólo se mostrarán estas constantes en caso que sea necesario, buscando simplificar los ejemplos.

Respecto a los verbos copulativos *ser* o *estar* utilizaremos la siguiente representación. El verbo copulativo no aparecerá en la fórmula lógica, unificando las variables de cada uno de sus modificadores (objeto directo e indirecto) con las del sujeto de la oración. Así podemos mostrar los siguientes ejemplos:

- ◆ *Pepe es abogado* $\rightarrow decl(abogado(Pepe))$
- ◆ *Tu perro es gris azulado* $\rightarrow decl(def(X, perro(X), gris(X) \& azulado(X)))$
- ◆ *Un amigo mio es de Alicante* $\rightarrow decl(existeUn(X, amigo(X), de(Alicante, X)))$

En cuanto a los *complementos circunstanciales* se colocarán de forma anidada incluyendo como uno de sus argumentos al predicado n-ario correspondiente al verbo. Hemos considerado los siguientes tipos de complemento circunstancial con sus representaciones de fórmula lógica:

- ◆ Cuando el complemento circunstancial esté formado por un adverbio y adjetivos calificativos, por ejemplo *Pepe canta muy fuerte*, se representará por la fórmula: *decl (muy (fuerte (cantar (Pepe))))*. En el caso que aparezcan varios adjetivos calificativos coordinados, como en *Pepe canta muy fuerte y claro*, se representaría por *decl (muy (fuerte (cantar (Pepe)) & claro (cantar (Pepe)))*.
- ◆ Cuando esté compuesto por un único adverbio de tiempo o de lugar, entonces este adverbio constituirá un predicario unario en el que su único argumento es ocupado por el predicado correspondiente a la fórmula lógica del verbo. Por ejemplo, en *Pepe canta hoy* se traduciría por *decl (hoy (cantar (Pepe)))*.
- ◆ Asimismo en caso de estar formado por un sintagma preposicional, como en *Pepe canta en Alicante*, utilizaremos la representación: *decl (en (Alicante, cantar (Pepe)))*. En el caso que haya coordinación de sintagmas preposicionales, como en *Pepe canta en Alicante y en Valencia*, sería *decl (en (Alicante, cantar (Pepe)) & en (Valencia, cantar (Pepe)))*.

4.4.4.2.6 Fórmula lógica de los sintagmas nominales

Hemos distinguido entre diversos tipos de sintagmas nominales en función de la palabra que realiza la función de núcleo del sintagma nominal: tipo nombre común, nombre propio, adjetivo, verbo en infinitivo, pronombre con función de sujeto y pronombre con función de sintagma preposicional. De nuevo, cada uno de ellos tendrá una fórmula lógica diferente:

- ◆ El sintagma nominal cuyo núcleo es un nombre común o los formados por un adjetivo, introduce un predicado unario cuyo argumento será la variable del sintagma que también utilizarán el resto de modificadores (cuantificadores, adjetivos calificativos, etc.). Por ejemplo, *el perro gris azulado de Antonio ha roto un cristal* se correspondería con *decl (def (X, perro (X) & gris (X) & azulado (X) & de (Antonio, X), existeUn (Y, cristal (Y), romper (X, Y))))*. En la oración cuyo núcleo es un adjetivo: *Aquel gris corre mucho*, se traduciría a *decl (def (X, gris (X), mucho (correr (X))))*.
- ◆ Cuando el núcleo está constituido por un nombre propio entonces no hay ninguna variable en su representación al introducir un término del lenguaje de fórmula lógica. Así la oración *Pedro compró un juguete*, se traduciría a *decl (existeUn (X, juguete (X), comprar (Pedro, X)))*.
- ◆ Para los sintagmas nominales formados por un verbo infinitivo se forma un predicado unario cuyo argumento corresponderá a la variable del sintagma nominal, como en el caso de *Juan sabe jugar al tenis*, que se transformaría a *decl (def (X, tenis (X), saber (Juan, jugar (X)))*.
- ◆ En cuanto a los sintagmas nominales formados por un pronombre los utilizaremos en el módulo de resolución de problemas lingüísticos para detectar las situaciones de anáfora, con lo que sustituiremos estos constituyentes por sus antecedentes. De todos modos, para los casos en que no se puedan resolver, hemos elegido la representación de un predicado unario con functor el propio pronombre y con

argumento una variable. Por ejemplo, en *Yo he comprado un libro* se traduciría a *decl (existeUn (X, libro (X), comprar (yo (Y), X)))*.

4.4.4.2.7 Fórmula lógica de los sintagmas preposicionales

Los sintagmas preposicionales introducen un predicado con dos argumentos excepto cuando introducen un sintagma preposicional previsto en la lista de huecos del elemento al que modifican (por ejemplo el objeto indirecto del verbo). El primer argumento es una variable que hace referencia a la entidad núcleo del sintagma preposicional (su sintagma nominal) y el segundo corresponde a la fórmula lógica de la entidad a la que modifica. Por ejemplo: *El gato de tu hermana arañó a Pepe*, tendría como fórmula lógica *decl (def (X, hermana (X), existeUn (Y, gato (Y) & de (X, Y), arañar (Y, Pepe)))*.

En caso que aparezcan varios sintagmas preposicionales coordinados, aplicaremos las mismas hipótesis que las ya comentadas para representar la coordinación de constituyentes, es decir, en la fórmula lógica se incluirán de forma explícita conectados por la correspondiente conectiva lógica. De este modo, la oración *Juan invitó a Pedro, Antonio y Luis* se traduciría a *decl (invitar (Juan, Pedro & Antonio & Luis))*.

5. Propuesta de resolución de la anáfora

En este capítulo expondremos nuestra propuesta de algoritmo para el tratamiento de la anáfora pronominal y de tipo adjetivo. Gracias al estudio realizado anteriormente de las aproximaciones al tratamiento de la anáfora existentes actualmente, podemos encuadrar nuestro algoritmo dentro de los sistemas integrados democráticos con un enfoque basado en restricciones (eliminan antecedentes factibles de una determinada anáfora) y preferencias (seleccionan uno entre varios candidatos posibles). Como sistema integrado, se basa en el conocimiento, es decir, maneja una serie de fuentes de información que se consideran necesarias para el correcto tratamiento de la anáfora. Y dentro de los sistemas integrados utiliza un enfoque democrático para coordinar estas fuentes de información, es decir, que aquellas entidades que pueden ser susceptibles de convertirse en antecedente pueden surgir por igual de aportaciones de cualquier fuente de información.

Organizaremos este capítulo en dos secciones. En la primera precisaremos la información que utilizamos y cómo la integramos con el sistema SUP descrito anteriormente. Y en la siguiente sección determinaremos los cambios necesarios para adaptar el algoritmo propuesto al trabajo con textos no restringidos. Para llevar a cabo esta adaptación realizaremos un análisis parcial sobre la salida de un etiquetador que nos obligará a adecuar las restricciones c-dominio a la información sintáctica de que dispondremos. Al limitar la cantidad de información de que se dispondrá en el algoritmo, podemos incluir nuestro sistema dentro de los estudiados en el apartado 3.2.2.3 *Sistemas pobres en conocimiento*.

5.1 Algoritmo de resolución de la anáfora

El algoritmo que aquí se propone se activará después de la fase de análisis sintáctico tal y como aparece en la Figura 61. Como entrada recibirá la estructura de huecos generada por este análisis sintáctico y como salida devolverá esta estructura de huecos modificada, donde se habrán resuelto las expresiones anafóricas. Siguiendo el esquema de trabajo de

SUP anteriormente explicado, se partirá de esta estructura de huecos modificada para obtener la fórmula lógica correspondiente.

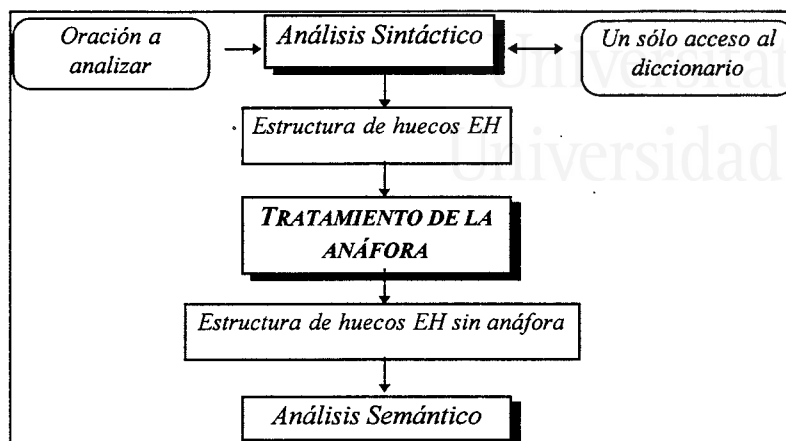


Figura 61. Fase de activación del módulo de tratamiento de la anáfora dentro del sistema SUP.

En la Figura 62 se muestra un ejemplo de la aplicación de este algoritmo para la frase *Pedro y Ana entraron y él compró*, mostrando la salida de cada una de las fases. En primer lugar se presenta la estructura de huecos obtenida tras el análisis sintáctico; a continuación la salida del módulo de tratamiento de la anáfora en el que se detecta la expresión anafórica formada por el pronombre personal con función de sujeto *él*, y entre los dos posibles antecedentes se selecciona al sintagma nominal formado por un nombre propio *Pedro*, puesto que es el único que concuerda en número, género y persona con el pronombre. Como salida del módulo de tratamiento de la anáfora se devuelve la estructura de huecos modificada en la que se ha sustituido el sintagma nominal tipo pronombre por el correspondiente antecedente, listando seguidamente el árbol sintáctico modificado junto con la fórmula lógica obtenida a partir de esa nueva estructura de huecos.

Es importante destacar que en el prototipo que hemos desarrollado en Prolog de este sistema se ha utilizado la gramática de oraciones declarativas en notación SUG expuesta en el apéndice A. Todos los ejemplos que mostremos sobre el tratamiento de la anáfora han utilizado esta misma gramática. El código correspondiente a la implementación del algoritmo de resolución de la anáfora que aquí proponemos en Prolog se muestra en el apéndice H.

En esta sección detallaremos el algoritmo que proponemos para la resolución de la anáfora. Explicaremos cómo afrontamos la distinción entre correferencia y no correferencia entre la expresión anafórica y su antecedente. También detallaremos las fuentes de información que intervendrán en el proceso de selección del antecedente, y finalmente, profundizaremos en el sistema de restricciones y preferencias de cada tipo de anáfora que se trata en este trabajo.

```

***** ESTRUCTURA DE HUECOS OBTENIDA TRAS EL ANÁLISIS SINTÁCTICO *****
orac(conc(pl,masc,terc),_0310,oracSimple(conc(pl,masc,terc),_0330,sn(conc(pl,masc,terc,sustAdj,suj),_03
5C,snCoordinado(c(sing,masc,terc,sustAdj,suj),_0390,snSimple(conc(sing,masc,terc,sustAdj,suj),_03D8,s
nSimplePodado(conc(sing,masc,terc,sustAdj,suj),Pedro,_0404,_0408,_040C,sust(c(sing,masc,propio
o),_0454,Pedro),_0414,_0418,_041C,_0420,_0424,_0428,_042C)),conj(_0470,_0474,y),coord(_03BC,_03
C0,snSimple(conc(sing,fem,terc,sustAdj,suj),_0494,snSimplePodado(conc(sing,fem,terc,sustAdj,suj),A
na,_04C0,_04C4,_04C8,sust(c(sing,fem,propio),_0510,Ana),_04D0,_04D4,_04D8,_04DC,_04E0,_04E4,
_04E8)),_03C8,_03CC))),sv(conc(pl,masc,terc,transIntrans),_0530,_0534,_0538,verbo(c(pl,terc,pasado,tra
nsIntrans),_0574,entraron),_0540,_0544,_0548,_054C,_0550)),conj(_05A0,_05A4,y),orac(conc(_0590,masc
c,terc),_05B4,oracSimple(conc(sing,masc,terc),_05E0,sn(conc(sing,masc,terc,pron,suj),_060C,snCoordina
do(c(sing,masc,terc,pron,suj),_0640,snSimple(conc(sing,masc,terc,pron,suj),_0670,snSimplePodado(con
c(sing,masc,terc,pron,suj),_0698,pron(c(sing,masc,terc,pers,suj),_06CC,él),_06A0,_06A4)),_0648,_06
4C)),sv(conc(sing,masc,terc,transIntrans),_06F4,_06F8,_06FC,verbo(c(sing,terc,pasado,transIntrans),_073
8,compró),_0540,_0708,_070C,_0710,_0550)),_05BC,_05C0))

***** MÓDULO DE RESOLUCIÓN DE LA ANÁFORA *****

Anáfora tipo pronSuj:
** SINT.NOMINAL SIMPLE:
** PRONOMBRE: él
— Sol.Anáf.
** SINT.NOMINAL SIMPLE:
** SUSTANTIVO: Pedro

***** ESTRUCTURA DE HUECOS MODIFICADA TRAS LA RESOLUCIÓN DE LA ANÁFORA **
orac(conc(pl,masc,terc),_0310,oracSimple(conc(pl,masc,terc),_0330,sn(conc(pl,masc,terc,sustAdj,suj),_03
5C,snCoordinado(c(sing,masc,terc,sustAdj,suj),_0390,snSimple(conc(sing,masc,terc,sustAdj,suj),_03D8,sn
SimplePodado(conc(sing,masc,terc,sustAdj,suj),Pedro,_0404,_0408,_040C,sust(c(sing,masc,propio),_045
4,Pedro),_0414,_0418,_041C,_0420,_0424,_0428,_042C)),conj(_0470,_0474,y),coord(_03BC,_03C0,snSi
mplePodado(conc(sing,fem,terc,sustAdj,suj),_0494,snSimplePodado(conc(sing,fem,terc,sustAdj,suj),Ana,_04C0,
_04C4,_04C8,sust(c(sing,fem,propio),_0510,Ana),_04D0,_04D4,_04D8,_04DC,_04E0,_04E4,_04E8)),_03C
8,_03CC))),sv(conc(pl,masc,terc,transIntrans),_0530,_0534,_0538,verbo(c(pl,terc,pasado,transIntrans),_05
74,entraron),_0540,_0544,_0548,_054C,_0550)),conj(_05A0,_05A4,y),orac(conc(_0590,masc,terc),_05B4,
oracSimple(conc(sing,masc,terc),_05E0,sn(conc(sing,masc,terc,pron,suj),_060C,snCoordinado(c(sing,masc
c,terc,pron,suj),_0640,snSimple(conc(sing,masc,terc,pron,suj),_0670,snSimplePodado(conc(sing,masc,
terc,sustAdj,suj),Pedro,_0404,_0408,_040C,sust(c(sing,masc,propio),_0454,Pedro),_0414,_0418,_041
C,_0420,_0424,_0428,_042C)),_0648,_064C)),sv(conc(sing,masc,terc,transIntrans),_06F4,_06F8,_06FC,v
erbo(c(sing,terc,pasado,transIntrans),_0738,compró),_0540,_0708,_070C,_0710,_0550)),_05BC,_05C0))

***** LISTADO DEL ÁRBOL SINTÁCTICO ***

** ORACION SIMPLE:
** SINT.NOMINAL:
** SINT.NOMINAL SIMPLE:
** SUSTANTIVO: Pedro
** CONJUNCION: y
** SINT.NOMINAL SIMPLE:
** SUSTANTIVO: Ana
** SINT.VERBAL:
** VERBO: entraron
** CONJUNCION: y
** ORACION SIMPLE:
** SINT.NOMINAL:
** SUSTANTIVO: Pedro
** SINT.VERBAL:
** VERBO: compró

***** FÓRMULA LÓGICA OBTENIDA ***
*** decl(entraron(Pedro & Ana,noHayOD,noHayOI) & compró(Pedro,ñoHayOD,noHayOI))

```

Figura 62. Salida para la frase: "Pedro y Ana entraron y él compró".

Este algoritmo se resume en la Figura 63 y está basado en el esquema general seguido por los sistemas integrados democráticos con un enfoque basado en restricciones y preferencias. Hemos preferido el esquema democrático frente al consultivo puesto que coincidimos con la opinión de diversos autores: los valores que aporta cada fuente de información son siempre relativos y no podemos decidir de antemano qué fuente es la que aportará la información decisiva. Igualmente hemos adoptado el enfoque basado en restricciones y preferencias en lugar del basado únicamente en preferencias, ya que siguiendo el estudio comparativo entre ambos enfoques realizado por Mitkov en [114], aunque no hay una gran diferencia de precisión entre ambos (83% de precisión para la basada en restricciones y preferencias, y un 82% para la basada en la técnica de razonamiento con incertidumbre con un $CF_{umbra} = 0,7$) el primero parece ser más exacto si se tratan de modo adecuado las excepciones a las restricciones.

En este algoritmo se procesará secuencialmente cada oración del texto a través de sus correspondientes estructuras de huecos. En cada una de estas estructuras realizará una búsqueda de expresiones anafóricas y antecedentes de izquierda a derecha. El modo en que se detecta cada una de estas expresiones es por medio de la estructura de huecos que identifica a cada constituyente de la gramática de modo único (gracias a su functor y número de argumentos). Por ejemplo, para detectar un pronombre personal, buscaremos una estructura de huecos con functor *pron* y dos argumentos. El segundo argumento corresponderá a la variable que utilizará en la fórmula lógica final, y el primero atañe a la estructura *conc* que almacenará su información morfológica y el tipo de pronombre. Dentro de esta estructura *conc*, se exigirá que su cuarto argumento contenga el valor *pers* que indica que se trata de un pronombre personal.

Como resultado del análisis sintáctico de una oración obtenemos su estructura de huecos en **SS1**.

Por cada expresión anafórica que encontremos en **SS1**:

Seleccionamos los antecedentes de las **X** oraciones anteriores en función del tipo de anáfora en **L0**.

Aplicamos restricciones a **L0**, devolviendo como resultado **L1**:

En caso que:

$|L1| = 1$ Entonces:

Éste será el antecedente elegido

$|L1| > 1$ Entonces:

Aplicar preferencias a **L1** dando como resultado **L2**

Se escogerá el primer antecedente de **L2**.

Se actualiza **SS1** con los antecedentes de cada anáfora devolviendo como resultado **SS2**.

Figura 63. Algoritmo para la resolución de la anáfora.

Los tipos de expresiones anafóricas que vamos a tratar son la pronominal, la anáfora superficial numérica y la de tipo adjetivo. Sus antecedentes van a ser siempre sintagmas nominales, por lo que su detección es similar: una estructura de huecos con functor *sn* y como núcleo un nombre común o propio. Puesto que vamos a tratar con la anáfora discursiva, nos vemos obligados a almacenar cada posible antecedente junto con información de su situación dentro del discurso. Toda esta información se consultará cuando se detecte una expresión anafórica. En función del tipo de anáfora obtendremos el número de oraciones que vamos a considerar en el proceso de selección de antecedentes. En el caso de las expresiones anafóricas pronominales buscaremos el antecedente en las dos

oraciones anteriores a la actual (en la que aparece el pronombre), mientras que para los sintagmas nominales tipo adjetivo, lo buscaremos en las cuatro anteriores³². El tipo de dato elegido para almacenar la información referente a cada antecedente y su posición en el discurso es la lista Prolog. Para cada antecedente se almacenará su estructura de huecos y otra estructura que almacene su posición en el discurso en la cabeza de la lista.

En la Figura 64 se muestra la lista de antecedentes que utilizará el prototipo para el texto ejemplo. Se puede apreciar la lista de antecedentes ordenados de derecha a izquierda³³ de las dos primeras oraciones. Cada uno de estos antecedentes se correspondería con su estructura de huecos en la que se almacena la información morfológica, pero además se ve acompañado por otra estructura de nombre *para* que incluye su información configuracional de la posición en la que aparece dentro del discurso. Esta estructura tiene cinco argumentos:

- ◆ El primero de estos argumentos se refiere al número de la oración en la que aparece. Este número se representa por medio de una variable Prolog sin instanciar, de ahí que para la primera oración de esta figura, aparezca el valor *_600* y para la segunda *_10717*. Este número será único para cada oración del discurso, y todos los antecedentes que aparezcan en esa oración compartirán el mismo valor.
- ◆ El segundo argumento se corresponde con el número de cláusula en el que aparece el antecedente. Por ejemplo, en la segunda oración aparecen dos cláusulas separadas por la coma, con los verbos *es* y *produce*. De este modo el antecedente *una modificación en la calidad espectral de la palabra* tiene el valor *2*, mientras que *el retardo* tiene el valor *1*.
- ◆ El tercer argumento almacena su posición respecto al verbo, con dos posibles valores: *av* (*antes del verbo*) y *dv* (*después del verbo*). Por ejemplo, dentro de la misma cláusula de la segunda oración, el antecedente *una modificación en la calidad espectral de la palabra* tiene el valor *dv*, mientras que *el eco* tiene el valor *av*.
- ◆ El cuarto se refiere al número de constituyente dentro de la oración según una numeración secuencial que empieza desde uno dentro de cada nueva oración. Por ejemplo, el antecedente *el retardo del eco* tiene el valor *1*.
- ◆ Y finalmente el quinto, se vincula con el número de subconstituyente coordinado. Por ejemplo, si tuviésemos el siguiente sintagma nominal coordinado: *Pedro, Juan y Antonio*, tendríamos que el primer sintagma nominal, *Pedro*, tendría en este argumento el valor *1*, mientras que *Juan* tendría valor *2*, y *Antonio* el *3*. Esta información será particularmente útil para resolver la *anáfora superficial numérica*, o sea, referencias del tipo *el primero*, *el segundo*, etc. También es importante destacar, que para resolver las referencias al sintagma nominal coordinado en su conjunto, también se almacenará un nuevo antecedente con

³² Esta diferencia en cuanto al número de oraciones consultadas según se trate de un pronombre o de un sintagma nominal tipo adjetivo, viene dada porque los segundos tienen una mayor carga léxica que los pronombres, por lo que permiten referencias a mayor distancia.

³³ Esta ordenación dentro de la lista de antecedentes también tiene su importancia, ya que gracias a ella almacenamos el concepto de distancia respecto a la expresión anafórica. Es decir, el primer antecedente que se consulte de esta lista será el más próximo a la expresión anafórica.

número plural y con el valor 0 en este argumento. Un ejemplo de este último caso sería la siguiente frase: *[Pedro, Juan y Antonio], compraron un libro. Ellos, se lo regalaron a Rosa*, en el que el pronombre ellos tiene como antecedente al sintagma nominal en su conjunto, es decir, *Pedro, Juan y Antonio*, que se detecta dentro de la lista de antecedentes gracias a que tiene este quinto argumento con valor 0 y número plural.

Frases:	
<i>El retardo del eco viene determinado principalmente por el retardo en ambos sentidos del trayecto de transmisión a cuatro hilos. Cuando el retardo es pequeño, el eco para el oyente produce una modificación en la calidad espectral de la palabra.</i>	
Lista de antecedentes que aparecen en esas frases:	
<ul style="list-style-type: none"> • una modificación en la calidad espectral de la palabra <i>para</i>(_10717,2,dv,8,1) • la calidad espectral de la palabra <i>para</i>(_10717,2,dv,8,1) • la palabra <i>para</i>(_10717,2,dv,8,1) • el eco para el oyente <i>para</i>(_10717,2,av,6,1) • el oyente <i>para</i>(_10717,2,av,6,1) • el retardo <i>para</i>(_10717,1,av,1,1) • el retardo en ambos sentidos del trayecto de transmisión a cuatro hilos <i>para</i>(_600,1,dv,5,1) 	<ul style="list-style-type: none"> • ambos sentidos del trayecto de transmisión a cuatro hilos <i>para</i>(_600,1,dv,5,1) • el trayecto de transmisión a cuatro hilos <i>para</i>(_600,1,dv,5,1) • transmisión a cuatro hilos <i>para</i>(_600,1,dv,5,1) • cuatro hilos <i>para</i>(_600,1,dv,5,1) • El retardo del eco <i>para</i>(_600,1,av,1,1) • el eco <i>para</i>(_600,1,av,1,1)

Figura 64. Ejemplo de la lista de antecedentes.

En cuanto a las restricciones y preferencias que vamos a tener en cuenta en este algoritmo, conviene adelantar que se adaptarán a cada tipo de expresión anafórica. Por ejemplo, en el caso de la anáfora pronominal la concordancia en número, género y persona se considerará una restricción mientras que en la anáfora tipo adjetivo se considerará como una preferencia. Además la restricción c-dominio sólo se aplicará para la anáfora pronominal y no para la de tipo adjetivo. A continuación detallamos el modo en que aplicamos las restricciones y preferencias utilizadas en el algoritmo.

5.1.1 Determinación de la correferencia

Este sistema de tratamiento de la anáfora también tiene la posibilidad de definir si la expresión anafórica y su antecedente van a ser correferentes o no. En caso que sí lo sean, en la fórmula lógica final ambos compartirán la misma variable, y en caso contrario cada uno tendrá una variable distinta. El hecho de compartir la variable es posible debido a que en el proceso de resolución de la anáfora, también se unifica la variable que se almacena en la estructura de huecos de cada constituyente (ver la Figura 54 en la página 132). Un ejemplo de correferencia se presenta en la Figura 65, en la que se trabaja sobre la frase *Luisa y Ana compraron una pera verde y una manzana roja ayer y Pedro eligió la verde*. En esta frase

se detecta la expresión anafórica *la verde*, formada por un sintagma nominal en el que se ha elidido el sustantivo, anáfora a la que denominamos de tipo adjetivo. En este caso se seleccionan como posibles antecedentes únicamente los sintagmas nominales *una pera verde* y *una manzana roja*, rechazando a *Luisa* y *Ana*, ya que son sintagmas nominales con una estructura diferente (su núcleo es un nombre propio, mientras que se buscan sintagmas nominales con un núcleo formado por un nombre común). Se escogerá como solución final al primer antecedente, o sea, *una pera verde*, ya que comparte el mismo adjetivo (*verde*) que la expresión anafórica, coincidiendo en número y género, además comparten los mismos modificadores. Esta correferencia se refleja por la variable *_0670* (en la notación habitual de las fórmulas lógicas aparecería como una letra mayúscula, por ejemplo *X*, sin embargo aquí se muestra la salida directa del prototipo computacional, por lo que aparece como una variable sin instanciar referenciada por un número precedido del carácter subrayado).

```

Anáfora tipo adj:
** SINT.NOMINAL SIMPLE:
** DETERMINANTE 1:
  ** ARTICULO: la
** ADYACENTE ADJETIVO:
  ** ADJETIVO SIMPLE: verde

Lista de antecedentes seleccionados:
** SINT.NOMINAL SIMPLE:
** DETERMINANTE 1:
  ** ARTICULO: una
** SUSTANTIVO: manzana
** ADYACENTE ADJETIVO:
  ** ADJETIVO SIMPLE: roja

** SINT.NOMINAL SIMPLE:
** DETERMINANTE 1:
  ** ARTICULO: una
** SUSTANTIVO: pera
** ADYACENTE ADJETIVO:
  ** ADJETIVO SIMPLE: verde

— Sol.Anáf.
** SINT.NOMINAL SIMPLE:
** DETERMINANTE 1:
  ** ARTICULO: una
** SUSTANTIVO: pera
** ADYACENTE ADJETIVO:
  ** ADJETIVO SIMPLE: verde

***** FÓRMULA LÓGICA OBTENIDA ***
*** decl(existeUn(_0670,pera(_0670) & verde(_0670),existeUn(_07E4,manzana(_07E4) & roja(_07E4),
ayer(compraron(Luisa & Ana,_0670 & _07E4,noHayO)))) &
def(_0670,pera(_0670) & verde(_0670), eligió(Pedro, _0670,noHayO)))

```

Figura 65. Ejemplo de correferencia en: “Luisa y Ana compraron una pera verde y una manzana roja ayer y Pedro eligió la verde”.

Un ejemplo de no correferencia entre la expresión anafórica y su antecedente sería el mostrado en la Figura 66 para la frase: *Luisa compró una pera verde y otra roja*. En este

caso, en la fórmula lógica final, la expresión anafórica *otra roja* no compartirá la misma variable que su antecedente *una pera verde*.

<p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ADJETIVO SIMPLE: <i>otra</i></p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE: <i>roja</i></p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO: <i>una</i></p> <p>** SUSTANTIVO: <i>manzana</i></p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE: <i>verde</i></p> <p>Antec. y anaf. no co-refieren</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ADJETIVO SIMPLE: <i>otra</i></p> <p>** SUSTANTIVO: <i>manzana</i></p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE: <i>roja</i></p> <p>***** FÓRMULA LÓGICA OBTENIDA ***</p> <p>*** decl(existeUn(_0590,manzana(_0590) & verde(_0590),def(_0704, manzana(_0704) & roja(_0704),compró(Luisa,_0590 & _0704,noHayO)))</p>
--

Figura 66. Ejemplo de no correferencia en: "Luisa compró una manzana verde y otra roja".

5.1.2 Información utilizada en el algoritmo

En esta subsección analizaremos por separado las distintas fuentes de información que utilizaremos en nuestro algoritmo: información léxica, morfológica, sintáctica y semántica.

5.1.2.1 INFORMACIÓN LÉXICA Y MORFOLÓGICA

La concordancia en número, género y persona, se comprobará por medio de la unificación de las estructuras *conc* descritas en la estructura de huecos generada por el sistema SUP. De este modo, en la frase *Pedro, Rosa y Luisa fueron al cine. Él compró palomitas y ellas compraron pipas*, el pronombre *él* tendrá la siguiente información: *conc (sing, masc, terc, pron, suj)*, y el antecedente *Pedro* tendrá *conc (sing, masc, propio)* por lo que la unificación de los argumentos correspondientes al número y género sería totalmente

válida. Sin embargo el pronombre *ellas* no cumpliría esta concordancia: *conc (pl, fem, terc, pron, suj)* al tener como segundo argumento el valor *fem*.

Existen excepciones a la concordancia en género y número que han de tratarse de manera adecuada para evitar errores en la detección del antecedente correcto. Estas excepciones se pueden resumir básicamente en las siguientes tres situaciones:

- ◆ Referencias a grupos de sintagmas nominales singulares ya estén coordinados, como en [*Juan y Pedro*]_i, *fueron a Paris. Ellos_i estuvieron allí durante cinco días*, o no: *Juan_i le dijo a Pedro_j que ellos_{i,j} deberían llamar ahora*. El primero de estos casos ya explicamos cómo lo resolvíamos mediante la estructura *paral* en la que por cada sintagma nominal coordinado, aparte de los sintagmas nominales individuales que lo componen, añadimos un nuevo antecedente con número plural y valor *0* en el quinto argumento de la estructura *paral* que englobe a todo el sintagma nominal coordinado (en el ejemplo anterior sería *Juan y Pedro*). En cuanto al segundo caso no lo trataremos en el presente trabajo, aunque una posible heurística podría ser que en el caso que la expresión anafórica estuviese en plural, no hubiese ningún antecedente en plural y que encontrásemos varios antecedentes con los mismos rasgos semánticos que el pronombre se podría construir un antecedente que los englobase a todos ellos y que constituiría la solución final.
- ◆ Referencias a uno de los elementos de un conjunto ocasionados principalmente por los cuantificadores de los sintagmas nominales. Por ejemplo: *En el zoo, un mono correteaba entre [dos elefantes]_i. Uno_i empujó al mono*. La detección de estos cuantificadores dentro de nuestro sistema es sencilla puesto que tenemos almacenada en la estructura de huecos de cada sintagma nominal toda su información sintáctica.
- ◆ Nombres colectivos que aceptan referencias en singular y en plural, los cuales se tratarán en el diccionario, permitiendo su entrada tanto con número singular como plural. Por ejemplo, en *El clero es conservador. A (ellos / éste) no les gusta el cambio*, donde el pronombre plural (*ellos*) referenciaría a los miembros del conjunto, mientras que el pronombre singular (*éste*) referenciaría al conjunto en su totalidad. Algo similar ocurre con determinados nombres propios sobre los que no se sabe a priori si tienen género masculino o femenino, como en el caso del nombre *Bresson* en *Who is this Bresson?* el cual tendría un tratamiento similar en el diccionario.

En cuanto a la información léxica, utilizaremos la correspondiente a la propia raíz de cada palabra a la hora de comprobar el parecido entre dos constituyentes, por ejemplo a la hora de resolver la anáfora de tipo adjetivo del siguiente ejemplo: *Luisa compró una pera verde y Ana compró dos verdes* en el que se concluye el parecido entre los sintagmas nominales *una pera verde* y *dos verdes*.

5.1.2.2 INFORMACIÓN SINTÁCTICA

La información sintáctica que utilizamos en el algoritmo de resolución de la anáfora la obtenemos fundamentalmente de la estructura de huecos que nos devuelve el sistema al analizar cada frase del texto. Este tipo de información se divide en dos tipos: restricciones c-dominio y paralelismo sintáctico, información que pasamos a detallar a continuación.

Las restricciones c-dominio, tal y como las explicábamos en la subsección 3.1.3, consisten en utilizar cierta información sintáctica que nos permite eliminar ciertos antecedentes no válidos para determinadas anáforas pronominales. Esto quiere decir, que estas restricciones sólo se utilizarán para resolver ciertos pronombres, o sea, no se aplicará para la anáfora tipo adjetivo. En nuestro sistema las podemos llevar a cabo gracias a la información sintáctica almacenada en la estructura de huecos devuelta tras el análisis de cada oración. Las restricciones c-dominio que incorporamos al sistema son las propuestas por Reinhart en [141]:

1. Un sintagma nominal completo se debe interpretar como no correferencial con cualquier sintagma nominal completo al que c-domine.
2. Un pronombre debe interpretarse como no correferencial con cualquier sintagma nominal al que c-domine.
3. Un pronombre reflexivo debe interpretarse como correferencial únicamente con un sintagma nominal que lo c-domine dentro de un espacio sintáctico definido: su *categoría mínima de gobierno*³⁴.
4. Un pronombre no reflexivo debe interpretarse como no correferencial con cualquier sintagma nominal que lo c-domine y que pertenezca a su categoría mínima de gobierno.

Comprobaremos la condición de c-dominio entre dos nodos A y B de una estructura sintáctica (A c-domina a B si y sólo si el primer nodo que domina a A , también domina a B) por medio de la estructura de huecos devuelta tras la fase de análisis sintáctico. En este caso, el obtener el primer nodo que domina a A es una operación sencilla puesto que se trata de saltar al nivel anterior del árbol de análisis, en el caso de la estructura de huecos: obtener el functor de la estructura de huecos donde está contenida la estructura A . Por ejemplo, en la Figura 67, con la frase *Juan entró y él compró*, obtenemos el nodo que c-domina al sintagma nominal formado por el pronombre *él* por medio de la estructura de huecos: *oracS* (*sn (él), sv (compró)*). En este caso *oracS* se correspondería en el gráfico con *Orac₃*. Y para el sintagma nominal *Juan* por medio de: *oracS* (*sn (Juan), sv (entró)*), o sea *Orac₂*. Una vez obtenidos los nodos dominantes, la comprobación de la restricción 2 es muy simple, ya que *Orac₂ ≠ Orac₃*, por lo que podrían ser correferentes.

³⁴ Estos conceptos ya se vieron en la subsección 3.1.3. A modo de recordatorio, la *categoría mínima de gobierno* de un nodo A , se define como el menor nodo que determine la oración o un sintagma nominal que c-domine a A y a su nodo principal (el del constituyente al que pertenezca A , por ejemplo el nombre de un sintagma nominal, o el verbo de un sintagma verbal).

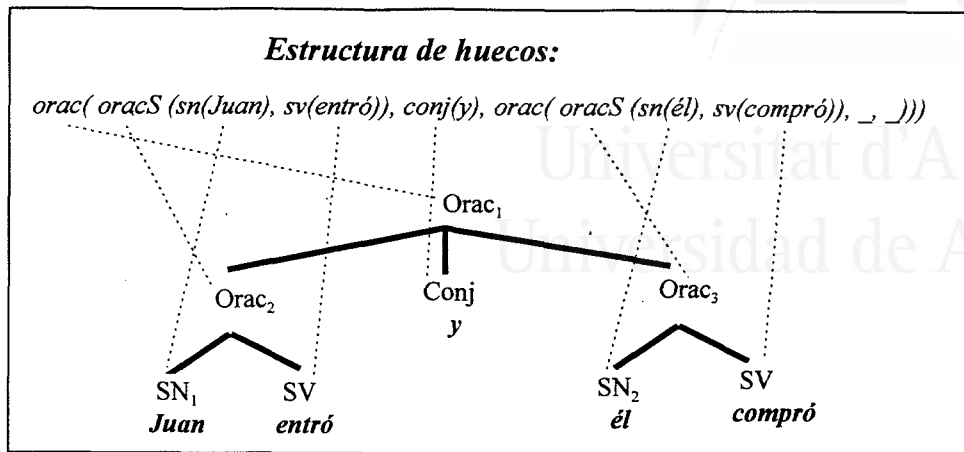


Figura 67. Ejemplo de comprobación de restricciones *c*-dominio por medio de la estructura de huecos de SUP en: "Juan entró y él compró".

La comprobación de la *categoría mínima de gobierno* de un nodo *A* (el menor nodo que determine la oración o un sintagma nominal que *c*-domine a *A* y a su nodo principal, es decir, el del constituyente al que pertenezca *A*, por ejemplo el nombre de un sintagma nominal, o el verbo de un sintagma verbal) de nuevo se llevará a cabo recorriendo de manera adecuada la estructura de huecos. En la Figura 68 se trabaja sobre la frase: *Marga compró para ella*. En este ejemplo, obtendremos la categoría mínima de gobierno para el sintagma nominal formado por el pronombre *ella*, por medio de un recorrido en sentido ascendente de la estructura de huecos: *oracS (sn (Marga), sv (v (compró), sp (prep (para), sn (ella))))*, desde *sn (ella)* hasta encontrar una estructura con nombre *oracS* o *sn*. En este caso se corresponderá en el gráfico con *Orac₁*. Lo mismo ocurriría con el antecedente *sn (Marga)* cuya categoría mínima de gobierno también coincide con *Orac₁*, por lo que gracias a la restricción 4 podemos concluir que ambos no pueden ser coreferentes.

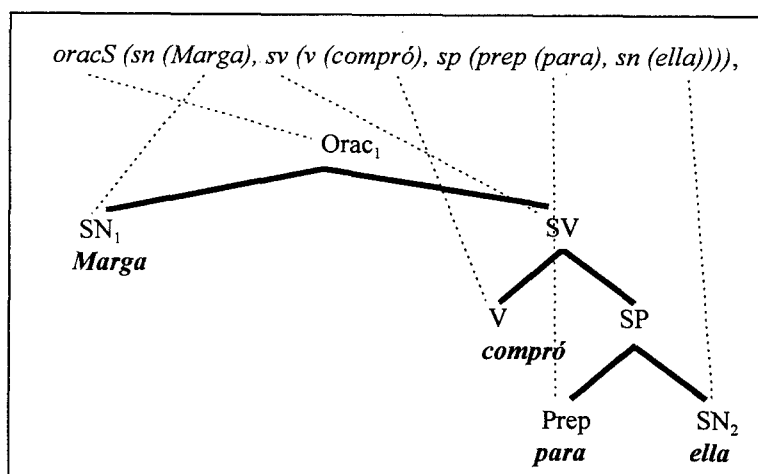


Figura 68. Restricciones *c*-dominio en "Marga compró para ella".

Esta misma restricción número 4 también nos ayuda a concluir que en: *la hermana de ella*, el pronombre *ella* nunca podrá coreferir con *la hermana* ya que tal y como se muestra en la Figura 69, pertenecen a la misma categoría mínima de gobierno: el sintagma nominal

SN_1 , con núcleo el sustantivo *hermana* y con un sintagma preposicional que lo modifica de *ella*.

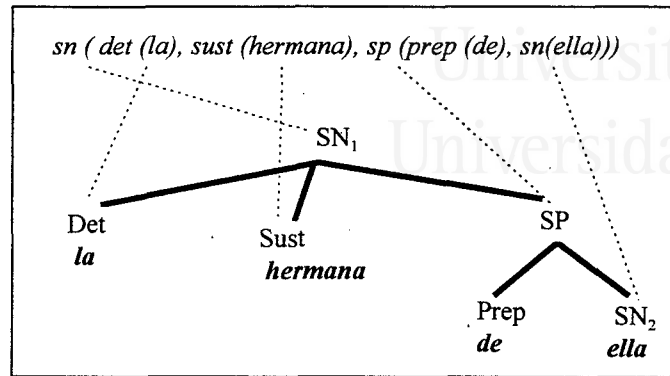


Figura 69. Restricciones c-dominio en "La hermana de ella".

Debido a esta misma restricción, tampoco se permitirá la correferencia en sintagmas coordinados en *Compré un libro para Marga y para ella* (Figura 70) o en *Marga y ella compraron un libro*, en los que no se permitirá la correferencia entre *Marga* y *ella*, ya que su categoría mínima de gobierno sería la propia oración en la que están incluidas. Sin embargo, en *Compré un libro para Marga y para la hermana de ella* sí que se permitiría la correferencia ya que tal y como se muestra en la Figura 71 la categoría mínima de gobierno de *ella* será SN_2 , en la cual no está incluida *Marga*.

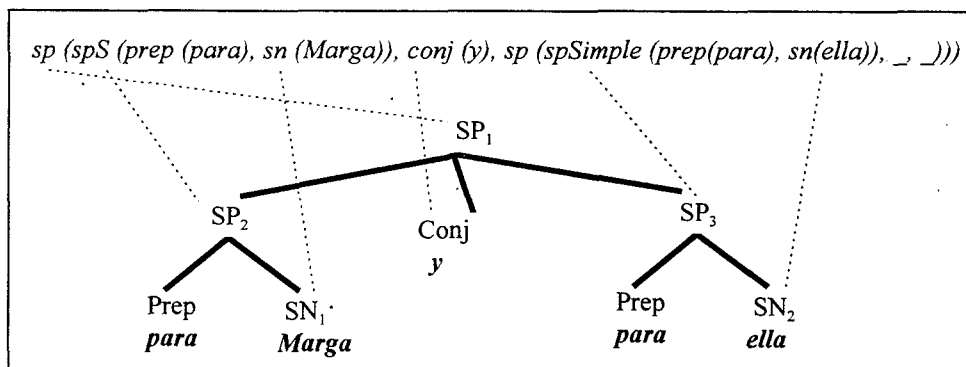


Figura 70. Restricciones c-dominio en "Para Marga y para ella".

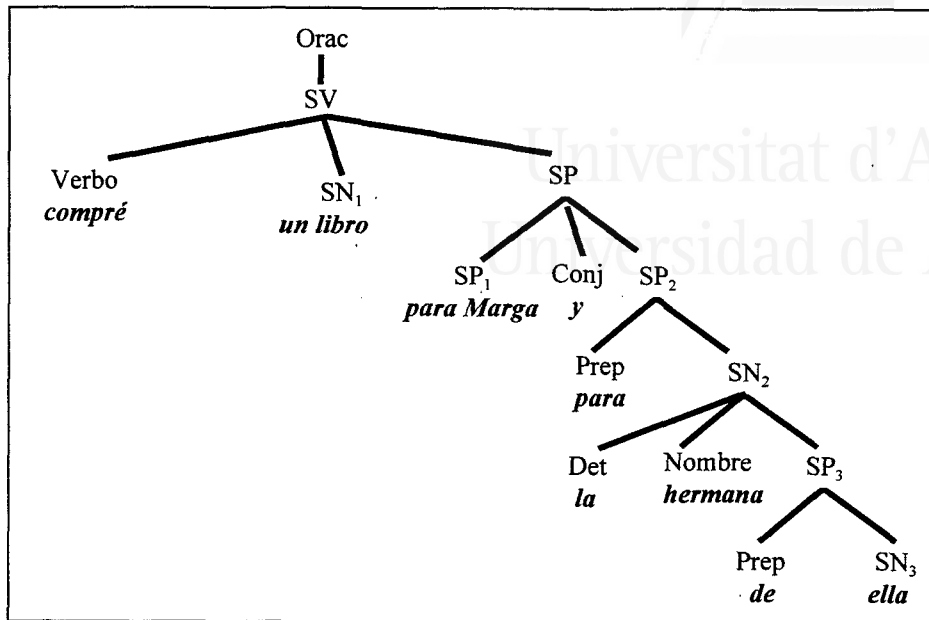


Figura 71. Restricciones c-dominio en “Compré un libro para Marga y para la hermana de ella”.

Y para finalizar con el apartado de las restricciones c-dominio, merece la pena distinguir entre las dos siguientes frases: *María, dijo que ella, vio a Pedro* y *Éste es el hombre, que él, vio*. En la primera sí que se permite la correferencia entre *María* y *ella* ya que la categoría mínima de gobierno del pronombre es *Orac₂* en la que no está incluida *María* tal y como se muestra en la Figura 72. Sin embargo, en la segunda (ver Figura 73) no se permitiría esa correferencia del pronombre *él* con *el hombre* ya que su categoría mínima de gobierno es *SN₂* (no sería *Orac₂* ya que es el menor nodo que determine la oración o un sintagma nominal que c-domine a *A* y a su nodo principal en este caso *hombre*) en la que está incluida *el hombre*. En nuestro prototipo se distinguen ambos casos gracias a la estructura de huecos que nos indica en cada caso el nodo principal. En la segunda frase tendremos: *sn (det (el), nombre (hombre), oracRelativo (pron (que), orac (sn (él), sv (vio)))*) en la que se puede apreciar que *oracRelativo* modifica al *nombre* (que es el nodo principal), mientras que en la primera aparecería *sv (v (dijo), oracRelativo (pron (que), orac (...))*) en la que *oracRelativo* modifica al verbo.

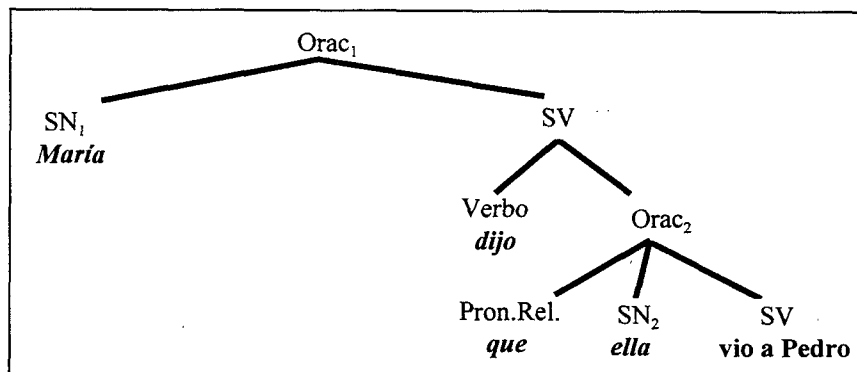


Figura 72. Restricciones c-dominio en “María dijo que ella vio a Pedro”.

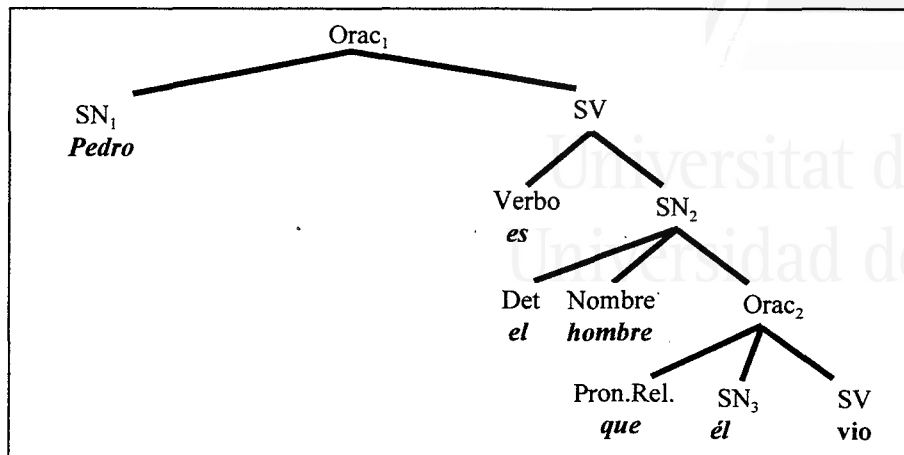


Figura 73. Restricciones c-dominio en “Éste es el hombre que él vio”.

Por otra parte, otro factor a tener en cuenta en la resolución de expresiones anafóricas es el paralelismo sintáctico. La información respecto al paralelismo sintáctico se obtiene de nuevo a partir de la estructura de huecos de la oración analizada. Este paralelismo se interpretará como la preferencia de una expresión anafórica por aquel antecedente que se encuentre en la misma posición sintáctica. Por ejemplo, en la Figura 74 se muestra la estructura de huecos obtenida al analizar la oración coordinada *Ana miró a la perra y la gata, y ella prefirió la segunda*, donde se aprecia la resolución del pronombre personal *ella* por medio del paralelismo sintáctico (preferencia por el antecedente con función de sujeto). También podemos observar la resolución de la anáfora superficial numérica correspondiente a la expresión anafórica *la segunda* gracias a la estructura de huecos de la coordinación de sintagmas nominales, que nos permite el acceso a cualquier constituyente coordinado en el orden que nos interese (primero, segundo, último, etc.), luego en este caso podemos seleccionar al sintagma nominal *la gata* como solución de *la segunda*.

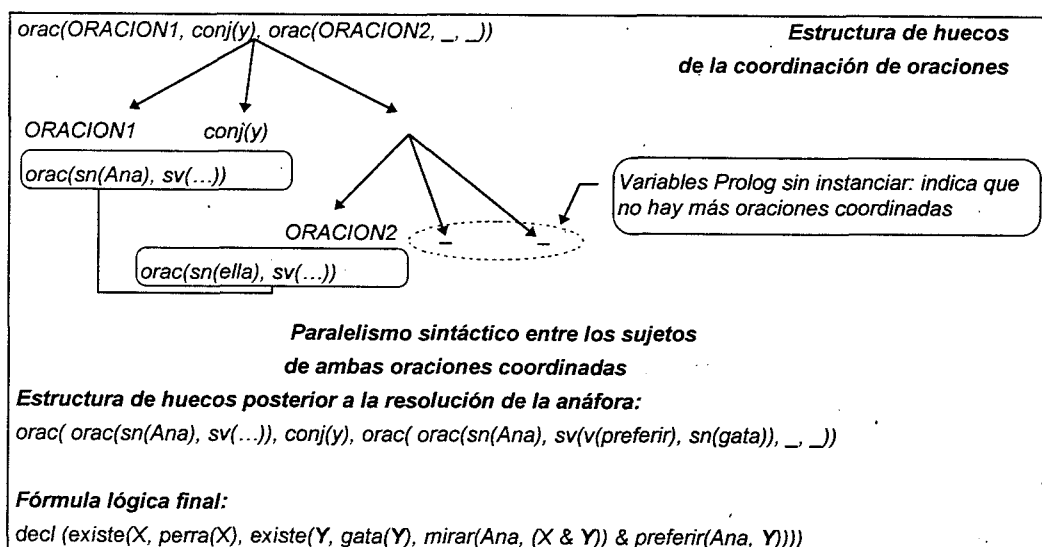


Figura 74. Aplicación del paralelismo sintáctico en: “Ana miró a la perra y la gata, y ella prefirió la segunda”.

5.1.2.3 INFORMACIÓN SEMÁNTICA

La consistencia semántica se comprobará por medio de la integración del módulo IRSAS en el sistema. El método IRSAS [122][121] proporciona información semántica en el proceso de análisis sintáctico. La estrategia que proporciona este método es asociar tipos semánticos (conjunto de rasgos) a las palabras y a los diferentes constituyentes de las oraciones a través de las estructuras de huecos. Se sigue un modelo estático ya que todos los rasgos deben conocerse a priori y están clasificados en una jerarquía de rasgos.

Jerarquía de rasgos semánticos en IRSAS:

Relaciones de división y herencia:

ontolog(entidad,[[concreto,abstracto]]) .

ontolog(entidad,[individual,colectiva]).

ontolog(abstracto,[[temporal,espacial,mental]]) .

ontolog(mental,[[estado,accion]]) .

ontolog(concreto,[[organico,inorganico]]) .

ontolog(organico,[[vegetal,animal]]) .

ontolog(animal,[[humano,nohumano]]) .

ontolog(animal,[[macho,hembra]]) .

ontolog(inorganico,[[geografia,fisica]]) .

ontolog(geografia,[[ciudad,montana]]) .

ontolog(fisica,[[solido,liquido,gas]]) .

ontolog(liquido,[[bebible,nobebible]]) .

ontolog(solido,[[comestible,nocomestible]]) .

ontolog(solido,[[vivienda,novivienda]]) .

ontolog(novivienda,[[monumento,vehiculo,prensa]]) .

Relaciones de implicación:

ontolog2(animal,[solido]).

ontolog2(ciudad,[solido]).

Diccionario:

esPalabra(ratón, [sustantivo(singular, masculino, nohumano)]).

esPalabra(coche, [sustantivo(singular, masculino, inorganico)]).

esPalabra(comió, [verbo(singular, masculino, animal, comestible)]).

esPalabra(queso, [sustantivo(singular, masculino, comestible)]).

Estructuras “conc” de cada posible antecedente:

“el ratón” conc(singular, masculino, nohumano)

“el coche” conc(singular, masculino, inorganico)

Comprobación de la consistencia semántica mediante IRSAS:

compatible (nohumano, animal).	Éxito: ANTECEDENTE SELECCIONADO
compatible (inorganico, animal).	Fracaso: ANTECEDENTE RECHAZADO

Figura 75. Aplicación de IRSAS para la comprobación de consistencia semántica en el texto: “El ratón, se paró cerca del coche. Éste, se comió un trozo de queso”.

Los rasgos semánticos se registran en la entrada léxica de cada palabra y se usarán en todas las reglas de la gramática como un argumento más (de igual modo que la información morfológica). De este modo se almacenarán automáticamente en la estructura de huecos de cada constituyente (dentro de *conc*), pudiéndose utilizar tal y como se muestra en la Figura 75. En esta figura se trabajará sobre el texto: *El ratón, se paró cerca del coche. Éste, se comió un trozo de queso*, en la que aparecen dos posibles antecedentes (*el ratón* y *el coche*) para el pronombre (*Éste*). El verbo de la oración en la que aparece el pronombre es *comer*, el cual tiene la siguiente estructura de rasgos semánticos: *animal* para su sujeto, y *comestible* para su objeto, y puesto que el pronombre actúa como sujeto, heredará el rasgo semántico *animal* (y su objeto, *un trozo de queso*, tendrá el rasgo *comestible*). El primer antecedente, *el ratón*, tiene el rasgo *animal*, mientras que el segundo, *el coche*, tiene el de *inorganico*. Dos rasgos semánticos serán compatibles si cumplen la condición de emparejamiento. La verificación de la consistencia semántica será llevada a cabo por el predicado: *compatible (RasgoSemantico1, RasgoSemantico2)*, el cual fracasará en caso que no exista consistencia semántica, y tendrá éxito en caso contrario. Para el ejemplo de esta figura, obviamente seleccionará el primer antecedente *el ratón* ya que fracasa el predicado *compatible* para el antecedente *el coche*.

5.1.3 Sistema de restricciones y preferencias

Cada expresión anafórica tendrá asociado un conjunto diferente de restricciones y preferencias, por ello en esta subsección vamos a detallar cada una de ellas en función de los tipos de anáfora que vamos a tratar: anáfora pronominal y anáfora tipo adjetivo.

5.1.3.1 RESTRICCIONES

Las restricciones tienen la función de eliminar candidatos que definitivamente no pueden ser antecedente de una determinada anáfora. Esta función de eliminación ha de ser indudablemente muy cuidadosa, es decir, no ha de existir la posibilidad de eliminar el antecedente correcto de la anáfora. Comentamos esto por las situaciones de excepción de la concordancia en número y género entre la anáfora y su antecedente ya vistas anteriormente, las cuales hay que tener en cuenta.

A la hora de aplicar cualquier restricción, recibiremos como entrada una lista de candidatos que estará ordenada por las posiciones que ocupan en el texto de derecha a izquierda, es decir, la cabeza de la lista la ocupará el candidato más cercano a la expresión anafórica. Como salida devolveremos una nueva lista con los candidatos que cumplen la restricción.

Para el caso de la anáfora pronominal consideraremos como restricciones la concordancia en número, género y persona, las restricciones c-dominio y la consistencia semántica. Para la anáfora tipo adjetivo sólo consideraremos como restricción el exigir como antecedentes sintagmas nominales que tengan como núcleo un nombre común y la consistencia semántica.

Un ejemplo de la aplicación de estas restricciones sobre la anáfora pronominal sería la siguiente frase: *Luisa compró queso y zumo. Finalmente comió éste*, en la que

inicialmente aparecen tres posibles candidatos: [*Luisa, queso, zumo*] para el pronombre *éste*. Aplicaremos en primer lugar la restricción referente a la concordancia en número, género y persona, con lo que nos quedaremos con sólo dos candidatos: [*queso, zumo*]. A continuación ejecutamos las restricciones c-dominio que como podemos observar cumplen todos los candidatos, con lo que la lista no se ve modificada. Y finalmente aplicamos la consistencia semántica mediante el método IRSAS. En este caso el verbo de la oración en la que aparece el pronombre es *comer*, el cual tiene la siguiente estructura de rasgos semánticos: *animal* para su sujeto, y *comestible* para su objeto, y puesto que el pronombre actúa como objeto, heredará el rasgo semántico *comestible*. El primer antecedente, *queso*, tiene el de *comestible*, mientras que el segundo, *zumo*, tiene el rasgo *bebible*. La compatibilidad entre dos determinados rasgos semánticos se llevará a cabo mediante el predicado Prolog: *compatible (RasgoSemantico1, RasgoSemantico2)*, que fracasará en caso que no exista consistencia semántica y tendrá éxito en caso contrario. Para este ejemplo, obviamente seleccionará el primer antecedente *queso* ya que fracasa el predicado *compatible* para el antecedente *zumo*, por lo que la lista final quedará con un sólo candidato que se considerará como solución final: [*queso*].

Para el caso de la anáfora tipo adjetivo, la primera restricción la llevaremos a cabo a partir de la estructura de huecos generada tras el análisis sintáctico de cada frase. En este tipo de anáfora necesitamos distinguir entre sintagmas nominales cuyo núcleo sea un nombre común, un nombre propio, un adjetivo, un pronombre con función de sujeto, un pronombre con función de complemento y un verbo en infinitivo. Cada uno de estos sintagmas nominales son fácilmente identificables por sus estructuras de huecos: diferente número y tipo de argumentos. De este modo las expresiones anafóricas de este tipo estarán formadas por sintagmas nominales tipo adjetivo, y los posibles antecedentes a considerar han de ser de tipo nombre común. Por ejemplo, en la frase: *Luisa y Ana compraron un libro, una pera verde oscura y una manzana roja ayer y Marga se comió la verde*, se detecta la anáfora formada por el sintagma nominal tipo adjetivo *la verde*, y como posibles antecedentes la siguiente lista: [*Marga, una manzana roja, una pera verde oscura, un libro, Ana, Luisa*]. Después de aplicar la primera restricción nos quedaremos únicamente con los sintagmas nominales cuyo núcleo sea un nombre común, rechazando a *Luisa, Ana* y *Marga* ya que son sintagmas nominales tipo nombre propio. La segunda restricción, la consistencia semántica, se aplicará sobre los restantes candidatos: [*una manzana roja, una pera verde oscura, un libro*], de los cuales los dos primeros tienen el rasgo semántico *comestible*, mientras que el *libro* tiene el de *inorgánico*. El verbo de la cláusula donde aparece la anáfora tipo adjetivo, *comer*, requiere un objeto de tipo *comestible* por lo que se eliminará el candidato *libro*, quedando los dos últimos: [*una manzana roja, una pera verde oscura*] que se pasarán como entrada del módulo de preferencias.

El orden en que se apliquen estas restricciones no es importante tal y como ya advirtió Mitkov en [114] (en su trabajo comparó dos estrategias integradas democráticas, una basada en un enfoque con restricciones y preferencias, y otra con sólo preferencias), ya que su objetivo es eliminar candidatos. Sin embargo es evidente que computacionalmente será preferible ejecutar en primer lugar las restricciones más rápidas y sencillas de comprobar, ya que disminuirá el número de operaciones necesarias para comprobar las restantes restricciones más complejas.

5.1.3.2 PREFERENCIAS

Según hemos visto en el algoritmo de resolución de la anáfora, las preferencias se aplicarán sólo si hay más de un posible antecedente tras aplicar las restricciones con el fin de ordenar estos antecedentes. Esta ordenación se lleva a cabo mediante diversos criterios (paralelismo sintáctico, proximidad, etc.). Para ello se designan diversos niveles de preferencia por cada tipo de anáfora (ver Figura 76) de tal manera que se irán aplicando secuencialmente empezando por los de nivel 1 (mayor nivel de preferencia). Si aplicamos la preferencia de nivel 1 y nos queda sólo un antecedente, escogeríamos éste como solución. Si no nos ha quedado ningún antecedente, aplicamos el siguiente nivel de preferencia a la lista inicial de antecedentes. En caso contrario cuando todavía nos queda más de un antecedente, entonces aplicamos la de siguiente nivel con los antecedentes que nos quedan, así sucesivamente hasta que quede un sólo antecedente o no queden más niveles de preferencia. Si se da la situación en la que no quedan más niveles de preferencia por comprobar y todavía queda más de un antecedente, entonces aplicamos la preferencia de proximidad, es decir, nos quedamos con el antecedente más cercano a la expresión anafórica, que se corresponderá con la cabeza de la lista de antecedentes que ha quedado.

<i>Nivel de preferencia</i>	<i>Anáfora pronominal no reflexiva</i>	<i>Anáfora tipo adjetivo</i>
<i>1</i>	<i>Los antecedentes que estén en la misma oración</i>	<i>Los antecedentes que tengan la misma estructura de modificadores</i>
<i>2</i>	<i>Los que tengan la misma posición relativa al verbo</i>	<i>Los que tengan el mismo valor en esos modificadores</i>
<i>3</i>	<i>Los que tengan la misma posición relativa en la oración</i>	<i>Los que tengan el mismo género y número</i>
<i>4</i>	<i>Los que tengan la misma posición dentro del constituyente coordinado</i>	_____

Figura 76. Criterios de preferencia.

Podemos observar un ejemplo en *Luisa entró a la tienda. Pedro dijo que ella era graciosa*. Al aplicar las restricciones de la anáfora pronominal nos quedarían los siguientes antecedentes: [*la tienda, Luisa*], y al aplicar la preferencia de nivel 1 nos quedaría la lista vacía, por lo que volveremos a aplicar sobre la anterior lista de candidatos la siguiente preferencia, la de nivel 2. Al aplicar la preferencia por los candidatos que tengan la misma posición relativa al verbo nos quedaremos con un sólo candidato: [*Luisa*] ya que el pronombre se encuentra antes del verbo *era*, al igual que *Luisa* está antes del verbo *entrar*. Al haber un sólo candidato nos quedaremos con éste como antecedente del pronombre, finalizando el proceso de aplicación de preferencias.

Siguiendo el ejemplo del apartado anterior dedicado a la anáfora adjetivo: *Luisa y Ana compraron un libro, una pera verde oscura y una manzana roja ayer y Marga se comió la verde*, detectábamos la anáfora formada por el sintagma nominal tipo adjetivo *la verde*, y

tras aplicar las restricciones como posibles antecedentes la siguiente lista: *[una manzana roja, una pera verde oscura]* que se tomarán como entrada del módulo de preferencias. Al aplicar la primera preferencia nos quedaremos con los mismos candidatos ya que ambos tienen la misma estructura de modificadores: *determinante + nombre común + adjetivo calificativo*. En la segunda preferencia es cuando se selecciona el segundo candidato: *una pera verde oscura* ya que la lista de adjetivos yuxtapuestos³⁵ de la expresión anafórica está incluida en la de este candidato.

Es importante destacar, que en caso que el antecedente seleccionado para una anáfora de tipo adjetivo no coincida en número y género o no tenga los mismos modificadores entonces tendremos la certeza de que no existe correferencia entre ambos. Un ejemplo podría ser el siguiente: *Luisa compró una pera verde y otra roja*, en la que la anáfora *otra roja* introducirá un nuevo objeto en el discurso: *otra pera roja*.

Los motivos que nos han llevado a elegir este sistema de manejo de las preferencias en lugar de una asignación de valores numéricos a cada criterio de preferencia son principalmente dos:

- ◆ Al no asignar estos valores numéricos eliminamos la posible crítica de la arbitrariedad de los mismos, siendo necesario tan sólo el determinar el orden de aplicación: en primer lugar los que tengan mayor prioridad.
- ◆ Al establecer una relación secuencial de ejecución de preferencias facilitamos su posible modificación al ser un esquema muy flexible en el que podemos fácilmente añadir nuevos niveles de preferencia o modificar el orden de aplicación de los mismos, sin que afecte al resto del algoritmo.

5.2 Aplicación del algoritmo sobre textos no restringidos

En esta sección explicaremos los cambios que han sido necesarios para adaptar el algoritmo al trabajo con textos no restringidos. El texto no restringido sobre el que se han realizado los experimentos es el utilizado en CRATER³⁶. Este corpus contiene el manual de la International Telecommunications Union CCITT también conocido como *The Blue Book*. Consiste en una colección de textos de telecomunicaciones y contiene cinco millones de palabras etiquetadas automáticamente por la versión española del etiquetador de Xerox.

³⁵ Los adjetivos calificativos pueden aparecer yuxtapuestos según la gramática *SUG* que utilizamos. Esta yuxtaposición en la estructura de huecos se manifiesta por medio de una lista de adjetivos: *[verde, oscura]* y *[roja]* para los candidatos, y *[verde]* para la expresión anafórica, con lo que la comprobación del parecido entre estos modificadores se reduce a la inclusión entre listas: *[verde] ⊂ [verde, oscuro]* y *[verde] ⊄ [roja]*.

³⁶ Corpus Resources And Terminology ExtRaction. http://lola.llf.uam.es/~fernando/project/es_corpus.html

Los cambios que hemos realizado sobre el sistema los concretaremos en tres subsecciones. En la primera mostraremos los cambios realizados sobre la entrada que recibe el sistema, que suponen la eliminación del diccionario y el trabajo sobre la salida de un etiquetador. En la segunda subsección presentaremos el analizador parcial que trabajará sobre la salida del etiquetador. Y finalizaremos con las modificaciones sobre el algoritmo a las que nos hemos visto obligados, tanto referentes a la disminución de la cantidad de información de que dispondremos como referentes a la modificación de las restricciones.

5.2.1 Entrada del sistema

Tal y como acabamos de comentar la entrada del sistema se adaptará al formato del corpus etiquetado de manera similar a como realizan diversos sistemas actualmente: por ejemplo Kennedy y Boguraev [93], Mitkov y Stys [118], Stuckardt [158], Baldwin [13] o Williams, Harvey y Preston en [168].

Cada palabra etiquetada con Xerox tiene la siguiente sintaxis: (*palabra, forma base, etiqueta*), por ejemplo: (*conexiones, conexión, NCFP*). La etiqueta que le asigna Xerox a cada palabra contiene su información morfológica (número, género y persona) y el tipo de palabra de que se trata. Por ejemplo, *NCFP* se corresponderá con un nombre común femenino plural, y *VLFIIIP* a un verbo en futuro indicativo en primera persona del plural.

Para poder trabajar con esta salida del etiquetador Xerox, en primer lugar tendremos que crear la lista de palabras que utiliza el analizador SUP (ver la Figura 56 de la página 134). Esta lista se crea antes de empezar el análisis sintáctico de la oración y consiste en una lista de estructuras con functor *w* y aridad dos (el primer argumento es la palabra de la frase como un átomo Prolog y el segundo es la lista de entradas léxicas de la palabra). Debido a la modularidad del sistema, la adaptación a las nuevas etiquetas del diccionario (en este caso las etiquetas definidas en Xerox) no necesita modificar las de la gramática, sino tan sólo necesitamos definir el procedimiento que realice la transformación correspondiente. En la Figura 77 mostramos un fragmento de este procedimiento, en el que por ejemplo la etiqueta *ADJGFP* se transforma en la utilizada por la gramática *SUG*: *adjSimple (pl, fem, cal)*.

```

tipoWBB('ADJGFP', TPal) :- TPal=[ adjSimple(pl,fem,cal) ].
tipoWBB('ADJGFS', TPal) :- TPal=[ adjSimple(sing,fem,cal) ].
tipoWBB('ADJGMP', TPal) :- TPal=[ adjSimple(pl,masc,cal) ].
tipoWBB('ADJGMS', TPal) :- TPal=[ adjSimple(sing,masc,cal) ].
tipoWBB('NCFP', TPal) :- TPal=[ sust(pl,fem,comun) ].
tipoWBB('NCFS', TPal) :- TPal=[ sust(sing,fem,comun) ].
tipoWBB('NCMP', TPal) :- TPal=[ sust(pl,masc,comun) ].
tipoWBB('NCMS', TPal) :- TPal=[ sust(sing,masc,comun) ].

```

Figura 77. Código Prolog que realiza la transformación de etiquetas Xerox a *SUG*.

En la Figura 78 se muestra un diagrama sobre la integración en el sistema del traductor de etiquetas Xerox a etiquetas de la gramática *SUG*. En este diagrama se observan las fases mencionadas: en primer lugar se recibe la salida del etiquetador, a continuación se

traduce a la lista de entradas léxicas que utiliza SUP que servirá de entrada para el analizador sintáctico parcial que describiremos en la próxima subsección.

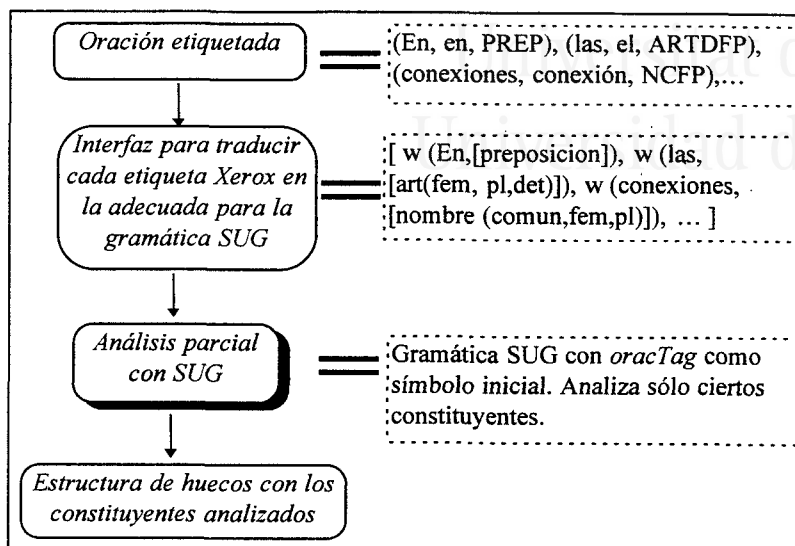


Figura 78. Esquema de trabajo del traductor de etiquetas Xerox a SUG.

Frase del corpus número 199:

"En las conexiones largas o de longitud media , es probable que la fuente principal de ruido de circuito estribe en los sistemas de transmisión analógica, ya que en ellos la potencia de ruido suele ser proporcional a la longitud del circuito"

Etiquetas del Corpus: (En, en, PREP), (las, el, ARTDFP), (conexiones, conexión, NCFP), (largas, largo, ADJGFP), (o, o, CC), (de, de, PREP), (longitud, longitud, NCFS), (media, medio, ADJGFS), (, , , , CM), (es, ser, VSP13S), (probable, probable, ADJGMS), (que, que, CQUE), (la, el, ARTDFS), (fuente, fuente, NCFS), (principal, principal, ADJGFS), (de, de, PREP), (ruido, ruido, NCMS), (de, de, PREP), (circuito, circuito, NCMS), (estribe, estribar, VLPS3S), (en, en, PREP), (los, el, ARTDMP), (sistemas, sistema, NCMP), (de, de, PREP), (transmisión, transmisión, NCFS), (analógica, analógico, ADJGFS), (, , , , CM), (ya que, ya que, CSUBF), (en, en, PREP), (ellos, él, PPX3MP), (la, el, ARTDFS), (potencia, potenciar, NCFS), (de, de, PREP), (ruido, ruido, NCMS), (suele, soler, VLP13S), (ser, ser, VSINF), (proporcional, proporcional, ADJGFS), (a, a, PREP), (la, el, ARTDFS), (longitud, longitud, NCFS), (del, del, PDEL), (circuito, circuito, NCMS), (, , , , FS)

Etiquetas de la gramática SUG:

[w(En,[prepSimple(_3161,_3162)]),w(las,[art(pl,fem,det)]),w(conexiones,[sust(pl,fem,comun)]),w(largas,[adjSimple(pl,fem,cal)]),w(o,[conj]),w(de,[prepSimple(_4281,_4282)]),w(longitud,[sust(sing,fem,comun)]),w(media,[adjSimple(sing,fem,cal)]),w(,[conj]),w(es,[verbo(sing,terc,presente,copul)]),w(probable,[adjSimple(sing,masc,cal)]),w(que,[conj]),w(la,[art(sing,fem,det)]),w(fuente,[sust(sing,fem,comun)]),w(principal,[adjSimple(sing,fem,cal)]),w(de,[prepSimple(_6523,_6524)]),w(ruido,[sust(sing,masc,comun)]),w(de,[prepSimple(_6972,_6973)]),w(circuito,[sust(sing,masc,comun)]),w(estribe,[verbo(sing,terc,presente,_7424)]),w(en,[prepSimple(_7647,_7648)]),w(los,[art(pl,masc,det)]),w(sistemas,[sust(pl,masc,comun)]),w(de,[prepSimple(_8321,_8322)]),w(transmisión,[sust(sing,fem,comun)]),w(analógica,[adjSimple(sing,fem,cal)]),w(,[conj]),w(ya que,[conj]),w(en,[prepSimple(_9437,_9438)]),w(ellos,[pron(pl,masc,terc,pers,suj)],pron(pl,masc,terc,pers,sp)),w(la,[art(sing,fem,det)]),w(potencia,[sust(sing,fem,comun)]),w(de,[prepSimple(_10346,_10347)]),w(ruido,[sust(sing,masc,comun)]),w(suele,[verbo(sing,terc,presente,_10798)]),w(ser,[verbo(_11021,_11022,infinitivo,copul)]),w(proporcional,[adjSimple(sing,fem,cal)]),w(a,[prepSimple(_11472,_11473)]),w(la,[art(sing,fem,det)]),w(longitud,[sust(sing,fem,comun)]),w(de,[prepSimple(sn,cc)]),w(el,[art(sing,masc,det)]),w(circuito,[sust(sing,masc,comun)]),w(,[FS]))

Figura 79. Ejemplo de traducción de etiquetas Xerox a SUG.

En la Figura 79 podemos ver un ejemplo de una frase del corpus junto con la traducción que realizamos de las etiquetas Xerox a las entradas utilizadas en la gramática SUG.

5.2.2 Análisis sintáctico parcial

Según Abney en [1], tradicionalmente los analizadores realizaban análisis exactos de las oraciones, y sólo aceptaban aquellas que eran reconocidas por la gramática. Para ello, se suponía un universo de discurso cerrado donde la gramática era completa para ese universo, es decir, que el conjunto de reglas reconocían todas sus frases, y por tanto el objetivo era encontrar el mejor analizador para la gramática.

Sin embargo, cuando se trata de textos no restringidos ocurre que los analizadores no pueden hacer análisis completos por falta de información léxica y de reglas gramaticales, añadiéndose además la dificultad que plantea la longitud de las oraciones y la propia ambigüedad de la gramática. Los analizadores parciales son una respuesta a esta dificultad. Las técnicas de análisis parciales recuperan información sintáctica relevante de los textos no restringidos sacrificando completitud y profundidad de análisis. La idea que plantean estos sistemas es la de realizar un análisis de estructuras de información (generalmente pequeñas) que sean relevantes y que puedan recuperarse con poca información sintáctica, en contraposición al sistema de análisis completo que intenta recuperar estructuras de información grandes que requieren mucha información, tal como información semántica y morfológica.

Según Brill y Mooney [20] en los últimos años en el campo del procesamiento del lenguaje natural junto con estas técnicas de análisis parcial, han surgido métodos empíricos que emplean técnicas de aprendizaje para extraer automáticamente conocimiento lingüístico de un corpus en lenguaje natural, en lugar de sistemas que requieren que este conocimiento lingüístico sea introducido manualmente.

En este trabajo pretendemos ofrecer una aproximación computacional al tratamiento de la anáfora discursiva que se pueda aplicar a cualquier texto independientemente del dominio. Para conseguir este objetivo es evidente que tenemos que superar el escollo del análisis sintáctico del texto a tratar, el cual presenta graves problemas principalmente ocasionadas por la ambigüedad estructural (ligamiento de constituyentes) o determinación de roles o funciones sintácticas (sujeto u objeto directo). Dados estos problemas inherentes al análisis sintáctico y de acuerdo con el trabajo de Stuckardt [158], el objetivo de conseguir un análisis sintáctico único es un objetivo poco realista en el estado actual de desarrollo del campo del procesamiento del lenguaje natural, por lo que se plantea la alternativa del análisis sintáctico parcial frente al análisis sintáctico completo, en el que se obvian estos problemas realizándose un análisis superficial del texto.

El término análisis parcial hace referencia a una serie de técnicas que intentan la recuperación de la información sintáctica de un corpus de texto no restringido sacrificando la completitud y la profundidad del análisis. La idea general de estas técnicas es la de fraccionar el análisis en pequeñas piezas que pueden recuperarse con poca información sintáctica sin necesidad de buscar información léxica para asociar los términos entre sí. Esas pequeñas piezas con sentido sintáctico es lo que se conoce en la literatura de los

análisis parciales como *chunk*, definidas como “secuencias de elementos con cierto sentido sintáctico alrededor de un núcleo o cabecera”.

Existen diversas propuestas de analizadores parciales en la literatura sobre el tema. Por ejemplo, Church [37] define una técnica que reconoce *chunks* nominales tomando como entrada la salida de un etiquetador basado en un modelo de Markov que marcaba estos *chunks* nominales insertando corchetes abiertos y cerrados entre pares de etiquetas, como por ejemplo: *[n]* v *[det n]*. Ramshaw y Marcus [139] también plantean una técnica de codificación de los *chunks* mediante etiquetas con corchete.

Basándose en el analizador de Ramshaw y Marcus, Hindle plantea el analizador *Fidditch* [79][78] como uno de los analizadores parciales de mayor éxito y uno de los más antiguos. El *Fidditch* nace como una forma de tratamiento de los textos no restringidos ampliando el de Marcus con un formalismo de reglas simplificado que permiten incluir una gramática extensa capaz de reconocer los marcadores de límite de frase, el sujeto y el predicado. Se trata de un analizador determinista por lo que se puede implantar fácilmente como un autómata de estados finitos, teniendo como principal característica su velocidad siendo capaz de procesar hasta 5600 palabras por segundo.

En [21] Brill plantea un método mixto que trataría el texto mediante un etiquetado con corchetes y por otra parte usaría técnicas de aprendizaje obtenidas desde un corpus de entrenamiento basándose en que se pueden conseguir análisis muy rápidos si previamente se consume tiempo en entrenamientos. Partiendo del razonamiento anterior, Vilain y Palmer [161] estudian algunas técnicas que permiten una sustanciosa mejora en las velocidades del aprendizaje.

Voutilainen [163] y posteriormente Karlsson [88] definen el analizador parcial ENGCG que utiliza un analizador léxico donde se etiqueta cada palabra con su categoría gramatical y sus funciones sintácticas posibles. Así se desambigua la función sintáctica de cada palabra de la misma forma que su categoría gramatical mediante reglas de coordinación de patrones. El análisis tiene como desventaja que sólo considera las relaciones sintácticas de las palabras entre sí y no con sus elementos rectores aunque consigue restringir el conjunto de análisis posibles.

Schwarz [148] define el analizador Copsy para identificar dependencias en sintagmas nominales mediante reglas de coordinación de patrones. Tiene como desventaja la necesidad de reglas relevantes, precisas y con poco coste computacional para que la velocidad se mantenga en niveles aceptables.

Joshi y Srinivas [84] describen un analizador parcial basado en técnicas de etiquetado como las de Voutilainen mediante *LTAG* (*Lexicalized Tree Adjoining Grammar*) donde cada árbol contiene un ítem léxico único que puede enlazarse mediante un grafo de dependencias. De esta forma cada palabra puede aparecer en múltiples árboles que representan las diferentes estructuras sintácticas posibles. El análisis seleccionará un único árbol para cada palabra mediante búsquedas de Viterbi.

Voutilainen y Padró [162] muestran un analizador híbrido completo en el que se realiza un análisis parcial según un modelo lingüístico con *Constraint Grammar* que se adaptan para poder incluir la información estadística obtenida de un corpus de entrenamiento. De esta forma se combinan reglas lingüísticas con la información estadística para incrementar la desambiguación a costa de disminuir su cobertura alcanzándose una

precisión del 96'1% y una cobertura del 97'2%. El analizador es capaz de identificar verbos, premodificadores, cabeceras nominales y adverbiales y algunos postmodificadores. Es un analizador de sintagmas nominales.

En la literatura expuesta hemos podido evidenciar la existencia de múltiples analizadores parciales destinados en su mayoría a la obtención de sintagmas nominales. En nuestro trabajo [101] planteamos un sistema de análisis parcial que además de los sintagmas nominales permite extraer otras estructuras sintácticas como los sintagmas verbales o los sintagmas preposicionales, permitiendo además la incorporación al análisis de otros tipos de información (morfológica y semántica). Esta característica nos permitirá proporcionar una mayor cantidad de información para aplicaciones posteriores, lo que ampliará considerablemente sus aportaciones tanto a técnicas de extracción de información como a técnicas estocásticas de ligamiento de subárboles sintácticos para formar constituyentes de orden superior (para mejorar la completitud del análisis).

Para realizar nuestro análisis parcial trabajaremos directamente sobre la salida del corpus etiquetado por medio de la incorporación de reglas a la gramática *SUG* descritas en la Figura 80. Esta gramática, en vistas de la complejidad de definir un analizador sintáctico que acepte cualquier texto no restringido, lo que hace es analizar cualquier sintagma nominal o preposicional coordinado, pronombre personal, verbo o conjunción que aparezca en el texto en cualquier orden desechando cualquier otra palabra que encuentre.

```

oracTag(_,_) ++>
  << SP:sp(,) >>,
  << SN:sn(,_,sustAdj,_) >>,
  << SNP:sn(,_,pron,_) >>,
  << PRON:pron(,_,pers,_) >>,
  << V:verbo(Numero,Persona,Tiempo,TipoSV) >>,
  << C:conj >>,
  <# [.,],
  oracTagSuelto(SP,SN,SNP,PRON,V,C)
  #>.

oracTagSuelto(SP,SN,SNP,PRON,V,C) ++>
  <## ( {(noExiste(SP),noExiste(SN),noExiste(SNP),noExiste(PRON),noExiste(V),noExiste(C))} ,[X]),
  ( _ , _ )
  ##>,
  oracTag(_,_).

```

Figura 80. Gramática *SUG* relajada para textos no restringidos.

De esta manera tan sólo tendremos que definir las reglas gramaticales correspondientes a los constituyentes a analizar (*sn*, *sp*, *pron*, *v*, *conj*). Para definir estos constituyentes, utilizaremos las mismas reglas gramaticales descritas en la gramática *SUG* de oraciones declarativas del apéndice A, salvo que se sustituirá su símbolo inicial (que correspondía a *orac* en *coordinated* (*orac* (Numero, Genero, Persona), *oracSimple* (, , , , _)) por *oracTag*.

Este análisis parcial es posible gracias a la flexibilidad que nos ofrecen los constituyentes opcionales definidos en *SUG* (constituyentes definidos entre el operador <<

>>, por ejemplo << C:conj >>). Su esquema de proceso se puede resumir del siguiente modo:

- ◆ Dentro del constituyente *oracTag*:
 - ◇ En primer lugar se intenta analizar un sintagma preposicional coordinado. En caso que lo encuentre lo almacena en la estructura de huecos correspondiente.
 - ◇ Después se pasa a analizar un sintagma nominal coordinado cuyo núcleo sea un nombre común, un nombre propio o un adjetivo (se corresponde con la regla gramatical del sintagma nominal de tipo *sustAdj*). En caso de encontrarlo se guarda en la estructura de huecos.
 - ◇ Continúa del mismo modo con el resto de constituyentes que se buscan: un sintagma nominal cuyo núcleo sea un pronombre, un verbo y una conjunción.
 - ◇ En caso que a continuación venga el final de la frase, o sea, que se analice un punto y final: [':.'], entonces se considera que ha finalizado el análisis de la frase actual.
 - ◇ En caso contrario se ejecuta el constituyente *oracTagSuelto*.
- ◆ Dentro del constituyente *oracTagSuelto*:
 - ◇ Se detectará si se ha analizado alguno de los constituyentes anteriores por medio de las etiquetas: *SP*, *SN*, *SNP*, *PRON*, *V* y *C*, en cuyo caso fracasará el predicado Prolog *noExiste (SP)*.
 - ◇ En caso que se haya analizado alguno de esos constituyentes continua el análisis.
 - ◇ En caso contrario se analiza una palabra “suelta”: [X]. Esta palabra se almacenará en la estructura de huecos con la etiqueta: *PALABRA*. Y a continuación continua el análisis.

En la Figura 81 mostramos el resultado del análisis parcial de una oración del corpus. En este resultado, se aprecia cómo las palabras que no han sido analizadas aparecen con la etiqueta *PALABRA* (en este caso se corresponde con *por lo general* que se ha etiquetado como adverbio). Ya que no utilizamos información semántica en el análisis, el problema añadido con que nos encontramos es que el ligamiento de los constituyentes entre sí es erróneo en algunas ocasiones. En esta figura podemos observar cómo se han coordinado incorrectamente los sintagmas nominales *ruido*, *la satisfacción* en la que el segundo sintagma nominal coordinado realiza la función de sujeto del siguiente verbo: *disminuye*.

Podemos encontrar un ejemplo de un tipo de sintagma no previsto en la gramática en el sintagma nominal que aparece en otra oración del corpus (la oración completa aparece en el apéndice F): *cabina telefónica cerrada*. En este sintagma nominal el modificador *telefónica* se ha etiquetado como un adjetivo y *cerrada* como un verbo en participio. Ambos modifican al nombre *cabina*, pero en la gramática del apéndice A no se ha previsto un adyacente adjetivo formado por yuxtaposición de adjetivos calificativos y verbos en participio, por lo que en el análisis sintáctico parcial resultante se devolverá por una parte un sintagma nominal: *cabina telefónica* y por otro lado un verbo en participio: *cerrada*.

Por lo general , para cualquier tipo de ruido , la satisfacción disminuye en forma monótona a medida que aumenta la potencia de aquél .

** ORACION ANALIZADA PARCIALMENTE:

** PALABRA: **Por lo general**

** CONJUNCION: ,

** SINT.PREPOSICIONAL:

** SINT.PREPOSICIONAL SIMPLE:

** PREPOSICION:

** PREPOSICION SIMPLE: **para**

** SINT.NOMINAL:

** SINT.NOMINAL SIMPLE:

** DETERMINANTE 1:

** ADJETIVO SIMPLE: **cualquier**

** SUSTANTIVO: **tipo**

** SINT.PREPOSICIONAL:

** SINT.PREPOSICIONAL SIMPLE:

** PREPOSICION:

** PREPOSICION SIMPLE: **de**

** SINT.NOMINAL:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO: **ruido**

** CONJUNCION: ,

** SINT.NOMINAL SIMPLE:

** DETERMINANTE 1:

** ARTICULO: **la**

** SUSTANTIVO: **satisfacción**

** VERBO: **disminuye**

** SINT.PREPOSICIONAL:

** SINT.PREPOSICIONAL SIMPLE:

** PREPOSICION:

** PREPOSICION SIMPLE: **en**

** SINT.NOMINAL:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO: **forma**

** ADYACENTE ADJETIVO:

** ADJETIVO SIMPLE: **monótona**

** CONJUNCION: **a medida que**

** VERBO: **aumenta**

** SINT.NOMINAL:

** SINT.NOMINAL SIMPLE:

** DETERMINANTE 1:

** ARTICULO: **la**

** SUSTANTIVO: **potencia**

** SINT.PREPOSICIONAL:

** SINT.PREPOSICIONAL SIMPLE:

** PREPOSICION:

** PREPOSICION SIMPLE: **de**

** SINT.NOMINAL:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE: **aqué**

Figura 81. Resultado del análisis parcial de una oración del corpus.

5.2.3 Transformación del algoritmo de tratamiento de la anáfora

Tal y como hemos descrito en anteriores ocasiones, el sistema de análisis sintáctico propuesto generará una estructura de huecos que mostrará los constituyentes analizados y el orden en que se han analizado. Esta estructura de huecos se utilizará en el algoritmo para la resolución de la anáfora ya descrito anteriormente. Para trabajar con el corpus *the blue book*, este algoritmo no varía en su esquema general de funcionamiento puesto que recibe una estructura de huecos, la procesa buscando antecedentes y expresiones anafóricas

devolviendo como salida esta estructura de huecos modificada en la que se han resuelto estas anáforas. Sin embargo sí que se ha tenido que adaptar en dos aspectos:

- ◆ En primer lugar, no se dispone de información semántica, ya que la salida del etiquetador únicamente nos ofrece la información morfológica de cada palabra: número, género, persona y categoría de la palabra (nombre, artículo, adjetivo, etc.), y tampoco podemos beneficiarnos de ninguna otra herramienta en la que esté la información semántica de los cinco millones de palabras de las que se compone el corpus.
- ◆ En segundo lugar, se va a realizar un análisis sintáctico parcial del corpus, por lo que no se dispone de toda la información sintáctica para poder determinar las restricciones c-dominio ni el paralelismo sintáctico.

La primera variación nos limitará en cuanto al porcentaje de éxito que obtengamos de nuestro estudio sobre el corpus, puesto que este tipo de información es absolutamente indispensable para resolver algunos casos de anáfora. Esta limitación se detecta en muchos sistemas desarrollados por diversos investigadores, y aún así proporcionan buenos porcentajes de éxito en el tratamiento de la anáfora. Esta limitación hace que nuestro sistema se pueda incluir dentro de los denominados *pobres en conocimiento*, explicados en el apartado 3.2.2.3.

La segunda variación nos permitirá trabajar con cualquier tipo de texto, pero nos limitará la cantidad de información sintáctica disponible. De este modo contaremos con la información sintáctica referente a un sintagma nominal coordinado completo, pero no tendremos la información que nos diga a qué constituyente está modificando este sintagma nominal. Sin embargo, propondremos una serie de reglas heurísticas que permitan suplir esta carencia.

Teniendo en cuenta el tipo de información sintáctica disponible y siguiendo el trabajo de Stuckardt [158], el sistema propuesto se puede considerar robusto ya que es capaz de superar la inexactitud del análisis sintáctico, trabajando con estructuras sintácticas deficientes o incompletas. Según se recordará, Stuckardt clasifica este tipo de sistemas robustos en dos tipos: *modelos de descripciones superficiales* y *modelos de descripciones deficientes*.

El trabajo de Kennedy y Boguraev [93] se encuadra dentro del primer tipo. Nuestro trabajo se incluye dentro del segundo tipo, ya que utiliza la información sintáctica asociada a sintagmas nominales y preposicionales coordinados, verbos y conjunciones.

Al igual que en la aplicación de nuestro sistema a textos restringidos, disponemos de la información sintáctica correspondiente a sintagmas nominales y preposicionales coordinados que nos permitirá aplicar las siguientes restricciones c-dominio:

- ◆ En el ejemplo de la Figura 69 página 158: *La hermana de ella*, conoceremos la no correferencia entre el nombre *hermana* y el pronombre *ella*, ya que se analiza como un sintagma nominal completo: *sn (det (la), n (hermana), sp (prep (de), sn (ella)))*, no permitiéndose la correferencia de ningún pronombre que se encuentre dentro de un sintagma nominal con ningún otro componente que también esté incluido en el mismo.
- ◆ En la Figura 70 de la página 158: *Para Marga y para ella*, también determinaremos la no correferencia entre *Marga* y *ella* gracias a la estructura de

huecos correspondiente a la coordinación de sintagmas preposicionales (en el análisis parcial que realizamos se devolvería analizado como un sintagma preposicional conjunto) en la que no se permite la correferencia. Un caso similar lo encontraríamos en el siguiente ejemplo de coordinación de sintagmas nominales: *Marga y ella*, en el que no se permitiría tampoco la correferencia.

- ◆ Sin embargo en la Figura 71 de la página 159: *Compré un libro para Marga y para la hermana de ella*, sí que se permitiría la correferencia entre *Marga* y *ella* ya que el sintagma nominal en que aparece el pronombre está a su vez dentro del sintagma preposicional ligado. Este caso seremos capaces de detectarlo puesto que vamos a contar con toda la información sintáctica de la coordinación de sintagmas preposicionales: *sp (sp (prep (para), sn (Marga)), conj (y), sp (prep (para), sn (det (la), sust (hermana), sp (prep (de), sn (ella))))*).
- ◆ En la Figura 73 de la página 160: *Éste es el hombre que él vio*, tampoco permitiríamos la correferencia gracias a que se dispone de toda la información sintáctica de la oración de relativo que modifica a *el hombre*: *sn (det (el), n (hombre), oracRel (...que él vio...))*.

Sin embargo en este análisis parcial que realizamos no disponemos de la información sintáctica que nos permita establecer la no correferencia entre *Marga* y *ella* en el ejemplo de la Figura 68 página 157: *Marga compró para ella*, ya que tan sólo contamos con una secuencia de constituyentes: *sn (Marga), v (compró), sp (para ella)*, y no sabemos la relación entre todos ellos para poder establecer los conceptos de *c-dominio entre dos nodos* ni el de la *categoría mínima de gobierno de un nodo A*. Algo similar ocurre con el ejemplo de la Figura 67 página 157: *Juan entró y él compró*, en el que sí que pueden correferir *Juan* y *él*, ya que sólo disponemos de la siguiente secuencia de constituyentes: *sn (Juan), v (entró), conj (y), pron (él), v (compró)*.

Para suplir esta carencia en frases similares a los dos últimos ejemplos, hemos propuesto la siguiente heurística para poder establecer la no correferencia entre un pronombre no reflexivo y un sintagma nominal:

Un pronombre no reflexivo y un sintagma nominal no podrán ser correferentes en el caso que se encuentren en la misma cláusula.

Y para determinar la cláusula en que se encuentra cada constituyente utilizaremos la estructura *paral* que incluye la posición en la que un constituyente aparece dentro del discurso. Esta estructura tiene cinco argumentos: el número de oración, número de cláusula, posición respecto al verbo (antes o después), número de constituyente dentro de la oración y número de subconstituyente coordinado. Para ello utilizaremos el segundo argumento que nos indicará la presencia de dos constituyentes en la misma cláusula, pero como estamos realizando un análisis sintáctico parcial del texto, para determinar esta información consideramos la siguiente heurística:

Supondremos que entramos en una nueva cláusula siempre que analicemos una conjunción que no se utilice para la coordinación de constituyentes.

De este modo, en el ejemplo anterior: *Juan entró y él compró* con la siguiente secuencia de constituyentes analizados: *sn (Juan), v (entró), conj (y), pron (él), v (compró)*, gracias a *conj (y)* y aplicando la anterior heurística concluimos que *Juan* y *él* se encuentran

en cláusulas diferentes, por lo que pueden correferir ya que en sus estructuras $paral_{Juan} (1, 1, av, 1, 1)$ y $paral_{el} (1, 2, av, 4, 1)$ no coincide el segundo argumento referente al número de cláusula en que se encuentran.

Igualmente en *Marga compró para ella*, contamos con la siguiente secuencia de constituyentes: *sn (Marga)*, *v (compró)*, *sp (para ella)*, y puesto que no aparece ninguna conjunción del tipo requerido en la heurística anterior determinamos la no correferencia entre *Marga* y *ella*, ya que en sus estructuras $paral_{Marga} (1, 1, av, 1, 1)$ y $paral_{ella} (1, 1, dv, 3, 1)$ coincide el número de cláusula en que se encuentran.

Para aclarar esta última heurística en la determinación del número de cláusula mostramos el resultado del siguiente ejemplo: *Marga y ella compraron un libro* en el que aparece una conjunción pero ésta es utilizada para la coordinación de sintagmas nominales: *sn (sn (Marga), conj (y), sn (ella))*, *v (compraron)*, *sn (un libro)*, por lo que todos los constituyentes de esa frase se considerarán en la misma cláusula: $paral_{Marga} (1, 1, av, 1, 1)$ y $paral_{ella} (1, 1, av, 1, 2)$.

Igualmente es fácil imaginar que realizar un análisis parcial del texto dificulta el modo de determinar el paralelismo sintáctico entre constituyentes puesto que únicamente disponemos de una secuencia de sintagmas analizados. Para resolver este inconveniente también utilizaremos la estructura *paral* asociada a cada constituyente, más concretamente consultaremos la posición que ocupan ambos constituyentes respecto al verbo (tercer argumento: antes o después del verbo) y el número de oración y cláusula. Siguiendo el ejemplo de la Figura 74 página 160: *Ana miró a la perra y la gata. Ella prefirió la primera* tenemos la siguiente secuencia de constituyentes analizados para la primera oración: *sn (Ana)*, *v (miró)*, *sp (prep (a), sn (la perra), conj (y), sn (la gata))*, y los siguientes para la segunda: *sn (ella)*, *v (prefirió)*, *sn (det (la), adj (segunda))*. Para seleccionar el sintagma nominal paralelo sintácticamente al pronombre *ella* utilizaremos sus estructuras *paral*: $paral_{Ana} (1, 1, av, 1, 1)$ y $paral_{ella} (2, 1, av, 1, 1)$, en las que sus dos primeros argumentos nos indican que se encuentran en la cláusula número uno de diferentes oraciones, y ocupan la misma posición respecto al verbo: *av* (antes del verbo).

Siguiendo el mismo ejemplo, la resolución del sintagma nominal de tipo adjetivo ordinal: *sn (det (la), adj (segunda))*, nos obligará a buscar un sintagma nominal coordinado que tenga en el quinto argumento de su estructura *paral* el valor 2. Este sintagma lo encontramos en *la gata* que tiene la siguiente estructura: $paral_{gata} (1, 1, av, 3, 2)$.

En el siguiente ejemplo: *Pedro entró con Juan y compraron un regalo para él*, para el pronombre *él* tenemos tres posibles antecedentes: *un regalo*, *Juan* y *Pedro*, con las siguientes estructuras: $paral_{regalo} (1, 2, dv, 6, 1)$, $paral_{Juan} (1, 1, dv, 3, 1)$, $paral_{Pedro} (1, 1, av, 1, 1)$ y $paral_{el} (1, 2, dv, 7, 1)$. El primero, *un regalo*, se eliminará al encontrarse en la misma oración y cláusula que el pronombre. Entre los dos antecedentes restantes se decantará por *Juan* ya que está después del verbo (*dv*) al igual que el pronombre.

Como conclusión de este apartado mostramos en la Figura 82 la salida de nuestro sistema sobre una frase del corpus. En esta frase se detecta la anáfora pronominal *ellos* de tipo *pronSP* (significa que el pronombre ha sido hallado dentro de un sintagma preposicional: *en ellos*) y selecciona como antecedente *los sistemas de transmisión analógica*. Para el pronombre tenemos la siguiente estructura: $paral_{ellos} (_69934, 2, av, 9, 1)$ que nos indica que se encuentra en la oración con identificador *_69934*, en la cláusula número dos, antes del verbo de esta cláusula, que ha sido el constituyente analizado número

9 y que ocupa la posición número 1 dentro del mismo en caso de aparecer coordinado. Y para su antecedente: *paral_{antec.}* (*_69934, 1, dv, 7, 1*) que nos indica que se encuentra en la misma oración que el pronombre pero en distinta cláusula (por lo que pueden correferir), después del verbo y como constituyente analizado número 7. También se muestra la información morfológica de la expresión anafórica y su antecedente, la cual está almacenada en sus estructuras *conc*: *conc_{ellos}* (*pl, masc, terc, pron, sp*) y *conc_{antec.}* (*pl, masc, terc, sustAdj, sp*) que nos muestra la concordancia en número, género y persona entre ambos.

"En las conexiones largas o de longitud media , es probable que la fuente principal de ruido de circuito estribe en los sistemas de transmisión analógica , ya que en ellos la potencia de ruido suele ser proporcional a la longitud del circuito"

Anáfora tipo pronSP en:

** SINT.NOMINAL SIMPLE:

conc(pl,masc,terc,pron,sp)

** PRONOMBRE: **ellos**

Con paralelismo: *paral(_69934,2,av,9,1)*

Antecedente seleccionado:

** SINT.NOMINAL SIMPLE:

conc(pl,masc,terc,sustAdj,sp)

** DETERMINANTE 1:

conc(pl,masc)

** ARTICULO: **los**

** SUSTANTIVO: **sistemas**

** SINT.PREPOSICIONAL:

** SINT.PREPOSICIONAL SIMPLE:

conc(sn,_4320)

** PREPOSICION: **de**

** SINT.NOMINAL:

conc(sing,fem,terc,sustAdj,sp)

** SINT.NOMINAL SIMPLE:

conc(sing,fem,terc,sustAdj,sp)

** SUSTANTIVO: **transmisión**

** ADYACENTE ADJETIVO:

conc(sing,fem)

** ADJETIVO SIMPLE: **analógica**

Con paralelismo: *paral(_69934,1,dv,7,1)*

Figura 82. Ejemplo de aplicación del algoritmo sobre textos no restringidos.

6. Experimentos y resultados

En este capítulo vamos a mostrar las pruebas realizadas y resultados obtenidos en la resolución de la anáfora pronominal y tipo adjetivo que proponemos en este trabajo sobre nuestro sistema SUP. Estas pruebas se organizarán en tres secciones. En la primera, se busca evaluar el funcionamiento del analizador SUP descrito anteriormente. En la segunda, se evalúa el algoritmo de resolución de la anáfora aquí propuesto sobre casos de anáfora que suceden en textos restringidos. Y en la última sección se muestran los resultados obtenidos al aplicar este mismo algoritmo sobre textos no restringidos, en los que se limita la cantidad de información de que dispone el sistema, al realizar un análisis sintáctico parcial del texto.

6.1 Evaluación del comportamiento del sistema SUP

En esta sección mostramos las pruebas realizadas del sistema SUP con una gramática en notación SUG con una cobertura lingüística restringida a oraciones declarativas. Esta gramática será utilizada durante los posteriores experimentos que realicemos sobre el tratamiento de la anáfora. Estas pruebas se describen a lo largo de las tres próximas subsecciones. En la primera se describe dicha gramática, en la siguiente se muestra el corpus de oraciones declarativas utilizado para comprobar su correcto funcionamiento (análisis de frases correctas y obtención de sus fórmulas lógicas, y detección de frases declarativas incorrectas), y en la última subsección se explican los resultados obtenidos y se realiza un estudio de la eficiencia del algoritmo de análisis utilizado. Este estudio de eficiencia no es exhaustivo sino tan sólo una pequeña muestra de su funcionamiento con un determinado conjunto de frases.

6.1.1 Gramática

A continuación describimos la gramática en notación SUG con una cobertura lingüística restringida a oraciones declarativas que será utilizada durante los posteriores experimentos que realicemos sobre el tratamiento de la anáfora. Como rasgos generales que permiten hacerse una idea del tamaño de la gramática en cuestión, decir que dispone de 19

símbolos no terminales, 8 terminales, 17 reglas de producción SUG, 4 hechos *coordinated*, 1 hecho *juxtaposition*, 5 hechos *fusion* (que de no haberlas fusionado supondrían 12 reglas de producción SUG), 8 hechos *basicWord*, y 470 hechos *isWord* (lo que supone un diccionario de 470 palabras). Seguidamente se explica con mayor detalle algunos puntos relacionados con esta gramática.

En la definición de las estructuras del lenguaje se han empleado todas las características que facilita el formalismo SUG y que lo diferencia de las gramáticas DCG. Como símbolo inicial tenemos a *orac*, que analizará la coordinación de oraciones declarativas simples correctas con un determinado *Numero*, *Genero* y *Persona*, que no tiene por qué coincidir entre las sucesivas oraciones coordinadas.

Las oraciones declarativas simples están determinadas por el símbolo no terminal *oracSimple* que se compone de un sintagma nominal opcional y un sintagma verbal obligatorio. Este mismo constituyente analizará tanto oraciones personales como impersonales. En caso de oraciones impersonales (se detectaría por el verbo de tipo impersonal: *impers* o *impersOD*) se exigirá que no exista sujeto, hecho que se comprobará mediante el predicado *noExiste(SN)*. En el caso de oraciones personales en las que sí que puede existir el sujeto, se exige que el sujeto y el sintagma verbal concuerden en número, género y persona.

Como constituyentes con capacidad de ser coordinados, aparecen las oraciones, los sintagmas nominales, los sintagmas preposicionales, las preposiciones y los adjetivos calificativos, capacidad que se distingue al aparecer el hecho SUG *coordinated*. Además consideraremos la posibilidad de la yuxtaposición de adjetivos calificativos.

En la coordinación de sintagmas nominales, no se exigirá que concuerden en número, género, persona o tipo; sin embargo sí que se exigirá que concuerden en función, es decir, no se consideraría válida la siguiente coordinación: *Yo y tí*, ya que aunque representan la coordinación de sintagmas nominales de tipo pronombre, tienen distinta función. También cabe destacar cómo se calcula el número, género y persona resultante de la coordinación de sintagmas nominales por medio del predicado *Prolog { ngpResultante (SNC, Numero, Genero, Persona) }*, de este modo si todos los sintagmas nominales son de género femenino, el género resultante lo será a su vez femenino, pero en caso que aparezca uno masculino, el resultante será masculino. Respecto a la persona resultante, si todas son tercera persona, el resultado será tercera; si no, con que haya una primera persona, el resultado será primera, y en caso contrario será segunda persona.

Se han diseñado diferentes tipos de sintagmas nominales en función de la palabra que ejerza función de núcleo, como tipo *sustAdj* (un nombre común o propio, o un adjetivo calificativo), tipo *pron* con función de sujeto (el núcleo será un pronombre, por ejemplo *tú*, *yo*, *él*, etc.), tipo *pron* con función de sintagma preposicional (por ejemplo *por tí*, *por él*, etc.) y finalmente tipo *verboInf* (el núcleo será un verbo en infinitivo como en *el jugar al fútbol*, *fumar puros*, etc.).

Los constituyentes opcionales y el hecho SUG *fusion*, se han utilizado frecuentemente para reducir el número de reglas gramaticales con el objetivo de mejorar la eficiencia de la gramática. Podemos resaltar como ejemplo de reducción del número de reglas a la fusión de las reglas de un sintagma nominal tipo nombre común, nombre propio y tipo adjetivo calificativo en una sola regla, tal y como se muestra en la Figura 83.

Regla gramatical del sintagma nominal de tipo nombre común o propio	Regla gramatical del sintagma nominal de tipo adjetivo calificativo
<<DET1:det(Número, Género)>>.	<<DET1:det(Número, Género)>>.
<<DET2:det2(Número, Género, TipoDet2)>>.	<<DET2:det2(Número, Género, TipoDet2)>>.
{fifthen(TipoDet2==ord, existe(DET1))}.	{fifthen(TipoDet2==ord, existe(DET1))}.
(<<adyAdj(Número, Género)>>, SUST:sust(Número, Género, Tipo), {fifthen((Tipo==comun, Funcion==suj), (existe(DET1) ; existe(DET2)))}, <<pron(Número, Género, __pos, compl) >>, <<adyAdj(Número, Género)>>),	(<<adyAdj(Número, Género)>>, - {fifthen(Funcion==suj, (existe(DET1) ; existe(DET2)))}, <<pron(Número, Género, __pos, compl) >>, -),
<< sp(sn,ISP) >>.	<< sp(sn,ISP) >>.
<< aposicion(Número, Género) >>.	<< aposicion(Número, Género) >>.
<< gerundioSN >>.	<< gerundioSN >>.
<< oracRelat >>.	<< oracRelat >>.
Resultado de la fusión	
<<DET1:det(Número, Género)>>.	
<<DET2:det2(Número, Género, TipoDet2)>>.	
{fifthen(TipoDet2==ord, existe(DET1))}.	
(<<adyAdj(Número, Género)>>, SUST:sust(Número, Género, Tipo), {fifthen((Tipo==comun, Funcion==suj), (existe(DET1) ; existe(DET2)))}, <<pron(Número, Género, __pos, compl) >>, <<adyAdj(Número, Género)>>), ; (<<adyAdj(Número, Género)>>, {fifthen(Funcion==suj, (existe(DET1) ; existe(DET2)))}, <<pron(Número, Género, __pos, compl) >>).	
<< sp(sn,ISP) >>.	
<< aposicion(Número, Género) >>.	
<< gerundioSN >>.	
<< oracRelat >>.	

Figura 83. Fusión de sintagmas nominales tipo sustantivo y tipo adjetivo.

Aparte de la mejora de eficiencia de la gramática por medio de los componentes opcionales y el hecho *SUG fusion*, también se ha aplicado un proceso de poda mediante el predicado *poda(sn, Tipo, Funcion)*. Este predicado se ejecuta a la hora de detectar el tipo de sintagma nominal (*Tipo* y *Funcion*) y detecta si realmente viene a continuación un sintagma nominal. Si las palabras que vengan a continuación no corresponden a un sintagma nominal (por ejemplo que sea el punto y final de la frase o una preposición) entonces el predicado *poda* ejecutaría el predicado *Prolog fail*, asegurándose que no continuará el análisis actual de la frase por las reglas gramaticales correspondientes a los sintagmas nominales. Las reglas *Prolog* correspondientes a este predicado *poda*, se muestran en la Figura 84.

```

snSimple(Numero,Genero,Persona,Tipo,Funcion) ++>
  poda(sn,Tipo,Funcion), % Obtener TipoSn, FuncSn o podar
  snSimplePodado(Numero,Genero,Persona,Tipo,Funcion).
%----- Poda del árbol de búsqueda -----
poda(Lista, sn, TipoSn, _) :-
  ifthenelse(Lista=[Cab1Lista,Cab2Lista|_],
    true,
    %Sino
    (Lista=[Cab1Lista], Cab2Lista=Cab1Lista),
    esSn(Cab1Lista),
    ifthenelse(es(_pron(_,_,_)),Cab1Lista,
      ifthen( ( es(_pron(_,_rel,_),Cab2Lista) ; es(_verbo(_,_gerundio,_),Cab2Lista) ;
        es(_conj,Cab2Lista) ; \+ (esSn(Cab2Lista))),
        TipoSn=pron ),
      %Sino
      ifthenelse(( es(_verbo(_,_infinitivo,_),Cab1Lista) ; ( es(_art(_,_),Cab1Lista),
        es(_verbo(_,_infinitivo,_),Cab2Lista) ) ),
        TipoSn=verboInf,
        %Sino
        TipoSn=sustAdj)).
%----- Comprobación de si viene a continuación un SN -----
esSn(Palabra) :- % Condiciones con las que nos aseguramos
  % que a continuación no hay un SN
  ifthen( ( Palabra=w('.',_);
    (es(_verbo(_,_Tiempo,_),Palabra), nonvar(Tiempo),
      Tiempo\==infinitivo, Tiempo\==participio) ;
    es(_adv(af),Palabra) ;
    ( es(_pron(_,_pers,compl),Palabra),
      \+ (es(_art(_,_),Palabra)), \+ (es(_adjSimple(_,_),Palabra)) ) ;
    es(_conj,Palabra) ; es(_prepSimple(_,_),Palabra) ),
  fail ).

```

Figura 84. Algoritmo de poda.

En el apéndice A se han incluido todas las reglas gramaticales en notación SUG con una cobertura lingüística restringida a las oraciones declarativas, en las que se han añadido una serie de comentarios para facilitar su comprensión, así como ejemplos de frases que aceptaría la gramática o que debería rechazar (estos últimos vendrán precedidos por el especificador *, como por ejemplo la frase incorrecta gramaticalmente **Lo hice mi*).

6.1.2 Corpus de prueba de frases declarativas

Seguidamente vamos a describir el corpus de frases declarativas empleado para comprobar el funcionamiento de la gramática SUG de oraciones declarativas comentada en el apartado anterior. En esta comprobación, pretendemos examinar la fórmula lógica generada en caso que la frase sea gramaticalmente correcta, y también pretendemos verificar que es capaz de detectar frases agramaticales, para evitar la posible sobregeneración de dicha gramática.

Este corpus se muestra en el apéndice B, junto con los resultados obtenidos para el análisis de cada una de las frases que lo componen. Este corpus contiene 387 oraciones, con un total de 2953 palabras y una longitud media de 7,6 palabras por frase. Cada una de estas frases tiene la estructura que reseñamos a continuación. Cada frase está almacenada en un hecho Prolog de aridad tres. El primer argumento corresponde con un índice mediante el cual se puede acceder a esa frase. El segundo consiste en un pequeño comentario sobre el tipo de frase a la que corresponde, por ejemplo *coordinación de oraciones* o *sucesión de adjetivos calificativos*. Y el último argumento será la propia oración expresada como una lista en la que cada uno de sus elementos será una palabra de la frase. También destacar que se han incluido en dicho corpus un grupo de oraciones que se utilizaron para comprobar el funcionamiento del algoritmo de resolución de la elipsis propuesto en los artículos [53][128]. Hay un total de 48 frases con elipsis verbal, elipsis del sujeto y de complementos del verbo. Estas frases van desde el número 240 hasta la 344 del corpus.

6.1.3 Resultados obtenidos

Los resultados que devuelve el sistema SUP al trabajar sobre este corpus de oraciones declarativas se muestran en el apéndice B. Estos resultados abarcan a las fórmulas lógicas obtenidas de las frases correctas, la comprobación de que una frase es incorrecta si no es analizada por la gramática y en caso que sea posible muestra el motivo por el que no ha sido aceptada (por ejemplo que no aparezcan los objetos de un sintagma verbal transitivo), y finalmente los tiempos empleados para el análisis.

<u>Análisis de las oraciones correctas</u>								
Núm. Pal.	Tiempo Análisis	Tiempo Elipsis	Tiempo FL	Tiempo Total	% Análisis	% Elipsis	% FL	Núm. Orac.
2953	2419	85	370	2874	84.1684064	2.95755045	12.87404315	387

<u>Análisis de las oraciones incorrectas</u>	
**** Tiempo TOTAL empleado	= 676 centésimas de segundo
**** Núm. Palabras	= 710
**** Núm. Oraciones	= 129

Figura 85. Informe de tiempos empleados para el análisis del corpus de frases declarativas.

En cuanto a los informes del tiempo empleado para este análisis, se corresponden a la ejecución del sistema SUP en el intérprete Arity Prolog sobre un PC Pentium a 120 MHz. En estos informes analizamos el tiempo total en centésimas de segundo. Para las oraciones declarativas analizadas correctamente por el sistema, se obtuvo la siguiente información: número total de palabras (2953) y oraciones analizadas (387), y el tiempo empleado distribuido entre los distintos módulos del sistema: la fase de análisis sintáctico (84%), resolución de la elipsis (3%) y la obtención de la fórmula lógica (13%). Una de las ventajas que presenta el sistema SUP es que el tiempo dedicado a la obtención de la fórmula lógica no se vería incrementado al añadir sucesivos módulos de resolución de problemas

lingüísticos, ya que funciona a partir de la estructura de huecos que devuelven dichos módulos, es decir, se ejecuta en una fase posterior que no se ve afectada por el desarrollo de dichos módulos. Esta afirmación se ve corroborada al realizar este mismo análisis de tiempos añadiendo el módulo de resolución de la anáfora, con lo que el sistema empleó el mismo tiempo para el módulo de obtención de la fórmula lógica. En la Figura 85 se detallan los resultados obtenidos.

También hemos medido la importancia del algoritmo de poda en el módulo de análisis sintáctico. Para ello hemos repetido el análisis del corpus anterior pero eliminando el predicado de poda correspondiente, con lo que hemos obtenido un incremento de tiempos valorable en un 8'23%, con lo que se demuestra la importancia de esta herramienta en el análisis.

o(589, "['Antonio', canta, ']).

o(590, "['más', de, cuatro, perros, míos, hacen, ']).

o(591, "['más', de, cuatro, muy, grandes, perros, de, color, gris, hacen, en, ellos, ']).

o(593, "['más', de, cuatro, muy, grandes, y, viejos, perros, de, color, gris, y, ellos, hacen, en, ellos, ']).

o(595, "['más', de, cuatro, muy, grandes, y, viejos, perros, de, color, gris, y, de, ellos, y, ellos, hacen, en, y, por, ellos, ']).

o(596, "['más', de, cuatro, muy, grandes, y, viejos, perros, de, color, gris, y, de, ellos, y, ellos, hacen, en, y, por, ellos, y, corren, ']).

o(597, "['más', de, cuatro, muy, grandes, y, viejos, perros, de, color, gris, y, de, ellos, que, no, vienen, y, ellos, hacen, en, y, por, ellos, y, corren, ']).

o(598, "['más', de, cuatro, muy, grandes, y, viejos, perros, de, color, gris, y, de, ellos, que, no, vienen, 'mañana', a, jugar, con, ellos, y, ellos, no, les, hacen, a, ellos, en, y, por, ellos, y, no, corren, ']).

o(599, "['más', de, cuatro, muy, grandes, y, viejos, perros, míos, muy, grandes, de, color, gris, y, de, ellos, que, no, vienen, 'mañana', a, jugar, con, ellos, y, ellos, no, les, hacen, contando, a, ellos, en, y, por, ellos, no, fuman, ']).

o(600, "['más', de, cuatro, muy, grandes, viejos, y, viejos, grandes, rojos, perros, míos, muy, grandes, de, color, gris, y, de, ellos, y, de, ellos, que, no, vienen, 'mañana', a, jugar, con, ellos, y, ellos, no, les, hacen, contando, a, ellos, en, y, por, y, por, ellos, no, fuman, no, fuman, ']).

o(601, "['más', de, cuatro, muy, grandes, viejos, rojos, y, viejos, grandes, rojos, perros, míos, muy, grandes, de, color, gris, y, de, ellos, y, de, ellos, y, de, ellos, que, no, vienen, 'mañana', a, jugar, con, ellos, y, ellos, y, ellos, y, ellos, no, les, hacen, contando, a, ellos, en, y, por, y, por, y, por, ellos, no, fuman, no, fuman, no, fuman, ']).

o(602, "['más', de, cuatro, muy, grandes, viejos, rojos, azules, y, viejos, grandes, rojos, azules, perros, míos, muy, grandes, de, color, gris, y, de, ellos, y, de, ellos, y, de, ellos, y, de, ellos, que, no, vienen, 'mañana', a, jugar, con, ellos, y, ellos, y, ellos, y, ellos, y, ellos, no, les, hacen, contando, a, ellos, en, y, por, y, por, y, por, y, por, ellos, no, fuman, no, fuman, no, fuman, no, fuman, ']).

o(603, "['más', de, cuatro, muy, grandes, viejos, rojos, azules, verdes, y, viejos, grandes, rojos, azules, verdes, perros, míos, muy, grandes, de, color, gris, y, de, ellos, y, de, ellos, y, de, ellos, y, de, ellos, y, de, ellos, que, no, vienen, 'mañana', a, jugar, con, ellos, y, ellos, y, ellos, y, ellos, no, les, hacen, contando, a, ellos, en, y, por, y, por, y, por, y, por, y, por, ellos, no, fuman, no, fuman, no, fuman, no, fuman, ']).

o(604, "['más', de, cuatro, muy, grandes, viejos, rojos, azules, verdes, grandes, y, viejos, grandes, rojos, azules, verdes, grandes, perros, míos, muy, grandes, de, color, gris, y, de, ellos, y, de, ellos, y, de, ellos, y, de, ellos, y, de, ellos, que, no, vienen, 'mañana', a, jugar, con, ellos, y, ellos, y, ellos, y, ellos, y, ellos, y, ellos, no, les, hacen, contando, a, ellos, en, y, por, y, por, y, por, y, por, y, por, y, por, ellos, no, fuman, no, fuman, no, fuman, no, fuman, no, fuman, ']).

Figura 86. Colección de oraciones empleada para la evaluación de SUP.

Igualmente hemos realizado un breve estudio de eficiencia del analizador SUP con la misma gramática en notación SUP de oraciones declarativas descrita hasta ahora, pero sin incluir el módulo de resolución de la elipsis. Ya que SUP está implementado en Prolog, como sistema de análisis sintáctico utiliza el propio del lenguaje de programación Prolog, es decir, utiliza un algoritmo de análisis descendente, de izquierda a derecha con retroceso. Este tipo de análisis, como es sabido trae complejidades exponenciales en su peor caso a la hora de aplicarlo al lenguaje natural, aunque este peor caso sea poco probable y difícil de alcanzar. Sin embargo hemos conseguido mejorar esta complejidad exponencial con el

formalismo SUG al reducir considerablemente el número de ramas del árbol de búsqueda, al disminuir el número de reglas gramaticales mediante el operador opcional, el hecho fusion, y al utilizar el algoritmo de poda ya descrito anteriormente.

Para este estudio utilizaremos las frases declarativas mostradas en la Figura 86. Se han seleccionado estas frases en lugar de las pertenecientes al corpus descrito en el apéndice B, porque estas frases pretenden obligar al analizador SUP a que explore todas las ramas del árbol de búsqueda, con lo que tendríamos el peor caso posible de análisis, y además, porque se produce un crecimiento gradual de la longitud de las frases, para de este modo calcular la complejidad en función de dicha longitud.

Palabras Orac.Corr.	Tiempo (Cseg.)	Orden de complejidad	Ajuste Min.C.	Palabras Orac.Inc	Tiempo (Cseg.)	Orden de complejidad	Ajuste Min.C.
3	43,75	-0,46	39	4	25,25	0,26	34
6	31,75	0,50	42	7	29,25	0,92	41
13	46,75	1,12	53	14	74	0,03	58
17	63,25	0,40	60	18	76,75	0,0032	67
22	70,25	2,71	71	23	77	1,36	79
24	89	0,49	76	25	86,25	0,34	84
28	96	0,63	87	29	90,75	0,92	94
37	114,5	0,78	114	38	116,5	0,65	115
42	126,5	1,42	131	43	126,25	0,76	127
54	181	1,06	180	55	152,5	1,28	156
66	224	2,05	237	67	196,5	0,59	185
78	316	1,01	304	79	216,75	0,80	214
90	365,5	2,29	381	91	243	0,59	243
99	455		444	100	257		265

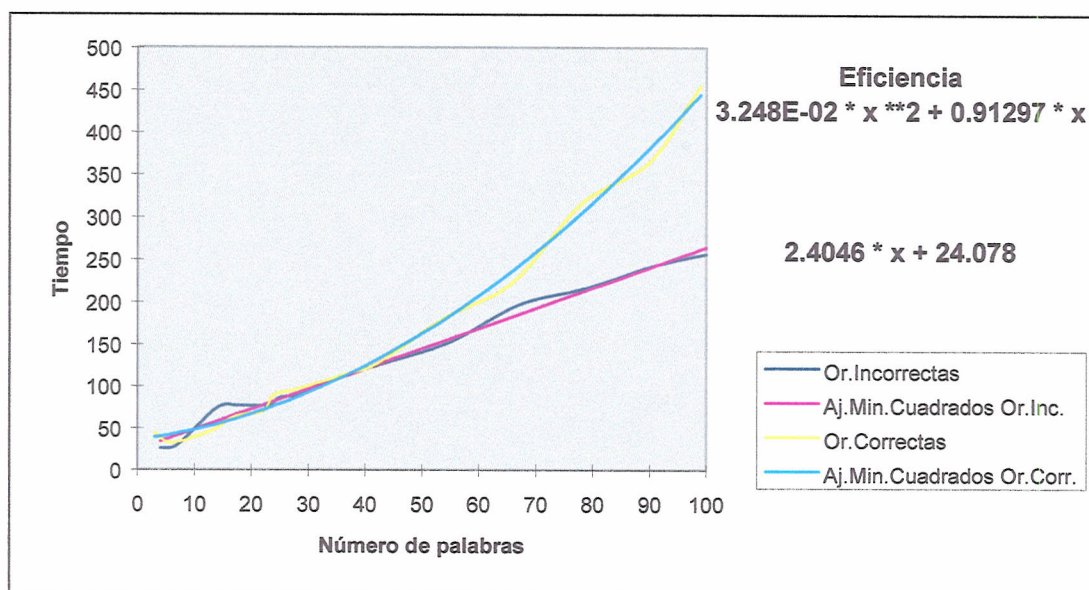


Figura 87. Eficiencia de las oraciones de la Figura 86.

Para esta evaluación se han tomado los tiempos que invirtió SUP en el análisis de las frases anteriormente descritas. Esta ejecución se realizó sobre un PC 386 - 40 MHz sobre el intérprete Arity Prolog 5.1. La complejidad la daremos en función del número de palabras de la frase, es decir, consideramos constante el tamaño de la gramática y el tamaño del diccionario. Esta complejidad se calcula a partir del tiempo invertido en el análisis de cada una de las frases, por medio de la siguiente fórmula: $n^{\text{Complejidad}}$, con n el número de palabras de la frase, y $\text{Complejidad} = (\log(T_i/T_0) / (\log(N_i/N_0)))$, con T_0 el tiempo de analizar la frase con N_0 palabras, y T_i el tiempo de analizar la siguiente frase del corpus con N_i palabras. Es importante destacar el objetivo de este experimento no es el realizar un exhaustivo estudio de la complejidad del analizador SUP, sino tan sólo el mostrar resultados de una implementación de una gramática SUG suficientemente amplia, que pueda darnos una idea del buen funcionamiento del algoritmo de análisis descendente con componentes opcionales y con el algoritmo de poda comentado anteriormente.

En la Figura 87 se muestran los resultados obtenidos en el análisis de cada una de las frases de la Figura 86: la longitud en palabras, el tiempo de análisis en centésimas de segundo, y finalmente el orden³⁷ de complejidad en función del número de palabras de la oración. Tal y como se constata, la complejidad del algoritmo tiene un coste cuadrático en función del número de palabras de la frase. Para justificar este resultado hemos realizado un ajuste por mínimos cuadrados, obteniendo la función: $3,248E-02 \cdot x^2 + 0,91297 \cdot x$.

En las cuatro últimas columnas de la Figura 87 se muestran los resultados obtenidos al analizar las oraciones anteriormente descritas en la Figura 86, pero añadiendo a cada frase una palabra al final: una conjunción. Con esto se busca intentar añadir ambigüedad al analizador, con el objetivo de obligarle a explorar todas las ramas del árbol de búsqueda, ya que una nueva conjunción podría suponer el encontrar un nuevo componente coordinado. En este caso, los resultados obtenidos son mejores que los obtenidos anteriormente, con un orden de complejidad siempre inferior a n^2 . Esta mejora en el resultado, intuimos que pueda deberse a que la frase resultante es gramaticalmente incorrecta, con lo que gracias a los algoritmos de poda correspondientes, se mejora la eficiencia del analizador en estos casos. Nuevamente realizamos un ajuste por mínimos cuadrados, obteniendo la siguiente función: $2,4046 \cdot x + 24,078$.

Finalmente en la Figura 87, también se muestra una gráfica en la que se observa la evolución de estas complejidades respecto al número de palabras, pudiéndose apreciar un comportamiento lineal en lugar de exponencial (crítica principal de los algoritmos de análisis descendente). En esta gráfica aparece la distribución de tiempos obtenida en nuestros experimentos junto con el ajuste realizado por mínimos cuadrados.

³⁷ Por este orden de complejidad entendemos el exponente de n . Es decir, para un valor de 2 en la quinta columna de estas tablas suponemos una complejidad de n^2 .

6.2 Experimento 1: tratamiento de la anáfora en textos restringidos

En esta sección presentamos los resultados obtenidos al aplicar el sistema SUP junto con el algoritmo de tratamiento de la anáfora que aquí proponemos sobre una colección de frases con anáfora de cobertura restringida. En estas frases, que se muestran en el apéndice C, aparecen casos de anáfora tanto intraoracional como interoracional. Por cada anáfora que se detecta, SUP muestra la siguiente salida:

- ◆ En primer lugar lista la expresión anafórica indicando el tipo de anáfora. Por ejemplo:
 - ◇ *pronSP* indica que se trata de un sintagma nominal cuyo núcleo es un pronombre que aparece dentro de un sintagma preposicional.
 - ◇ Del mismo modo *pronSuj* indica que es un sintagma nominal formado por un pronombre que no va dentro de un sintagma preposicional.
 - ◇ Y finalmente *adj* señala una expresión anafórica formada por un sintagma nominal de tipo adjetivo (aquel que está formado por un determinante y un adjetivo calificativo).
- ◆ A continuación se muestra la lista de antecedentes que han pasado las restricciones del tipo de anáfora. En caso que no haya ninguno muestra el mensaje: --- *OJO: Anáfora no resuelta o exófora.*
- ◆ Entre todos los posibles antecedentes seleccionará uno al aplicarle las preferencias asociadas, mostrando el antecedente elegido.

```

Oración Correcta Nº 1010 -> Anaf.pron.c-c.Sec.con sigte. [Ana, eligió, esta,]
***** MÓDULO DE RESOLUCIÓN DE LA ANÁFORA *****
Anáfora tipo pronSuj:
** SINT.NOMINAL SIMPLE:
** PRONOMBRE: esta
Lista de antecedentes seleccionados: --- OJO: Anáfora no resuelta o exófora
Oración Correcta Nº 1011 -> Anaf.pron.c-c. [Marga, lo, compró, para, ella,]
***** MÓDULO DE RESOLUCIÓN DE LA ANÁFORA *****
Anáfora tipo pronPersonal:
** PRONOMBRE: lo
Lista de antecedentes seleccionados: --- OJO: Anáfora no resuelta o exófora

Anáfora tipo pronSP:
** SINT.NOMINAL SIMPLE:
** PRONOMBRE: ella
Lista de antecedentes seleccionados:
** SINT.NOMINAL SIMPLE:
** SUSTANTIVO: Ana
--- Sol.Anáf.
** SINT.NOMINAL SIMPLE:
** SUSTANTIVO: Ana

```

Figura 88. Análisis de dos frases del corpus de frases con anáfora.

Por ejemplo, en la Figura 88 se analizan dos frases mostrando la salida obtenida. Respecto al primer pronombre que aparece en la primera frase, el sistema no encuentra ningún antecedente ya que el único sintagma nominal incumple las restricciones c-dominio, por lo que lo considera como exófora. Algo similar ocurre con el pronombre *lo* de la siguiente oración, pero para el pronombre *ella* sí que aparece un antecedente posible: el sujeto de la oración anterior *Ana* que es elegido como solución. En el apéndice D se muestra la salida de oraciones con anáfora pronominal, y en el apéndice E las que presentan anáfora de tipo adjetivo.

6.3 Experimento 2: tratamiento de la anáfora en textos no restringidos

En esta sección presentamos el experimento realizado del tratamiento de la anáfora en textos no restringidos. El texto sobre el que se ha aplicado el algoritmo es parte del ya comentado anteriormente *the blue book* (9.600 palabras). Pretendemos comprobar por una parte el funcionamiento del analizador parcial y por otra el funcionamiento del algoritmo de tratamiento de la anáfora pronominal y de tipo adjetivo sobre las estructuras sintácticas incompletas que nos devuelve el análisis sintáctico parcial.

Este analizador parcial recordamos que consiste en la gramática para analizar sintagmas nominales y preposicionales coordinados, pronombres, verbos y conjunciones de la Figura 80 página 170 y la gramática de oraciones declarativas del apéndice A. Con esto se consiguió analizar un 81% de las palabras del texto, que consistirá precisamente en una secuencia de estos constituyentes que acabamos de enumerar. Y en cuanto a su relación con el objetivo principal de este trabajo, la resolución de la anáfora, el analizador detectó el 100% de las expresiones anafóricas que aquí se tratan: pronombres y sintagmas nominales de tipo adjetivo.

También realizamos otro experimento para probar el funcionamiento del analizador parcial en el que intentamos analizar sintagmas nominales y preposicionales coordinados y sintagmas verbales, obteniendo los siguientes resultados: de un total de 9.600 palabras se analizaron un 76,76% de palabras relevantes desechando 2.231 palabras, efectuándose la extracción de 2.577 sintagmas nominales, 1.288 sintagmas preposicionales y 401 sintagmas verbales.

En cuanto a los resultados obtenidos en el tratamiento de la anáfora, las frases en las que se detectó una anáfora tienen una longitud media de 48 palabras, obteniendo una precisión del 83% en la detección de la posición del antecedente. Este criterio para valorar el éxito de nuestro sistema quiere decir que dados los problemas añadidos por la ambigüedad estructural y la carencia de información semántica, consideraremos como acierto el caso que dé como solución a todo el antecedente o parte de él. Por ejemplo, en la frase número 90 del corpus:

\ 43.20 @ punto de referencia boca (PRB) @ E : mouth reference point
(MRP) F : point de référence bouche (PRB) \ Punto situado a una

distancia de 25 mm frente a los labios de una boca humana típica (o una boca artificial) y sobre el eje de ésta (véase la figura A-1/P.64)

la solución que da nuestro sistema para el pronombre *ésta* es el sintagma nominal *una boca artificial*, cuando la solución completa sería: *una boca humana típica (o una boca artificial)* la cual supone una coordinación de sintagmas nominales no contemplada en nuestra gramática ya que el segundo sintagma nominal está delimitado por paréntesis. Por ello dentro de este 83% de precisión consideraremos este ejemplo como éxito.

El motivo de no alcanzar el 100% de éxito es principalmente debido a los problemas ocasionados por la carencia de información semántica. Por ejemplo, en la frase número 42 del corpus:

Calidad de transmisión 41.01 @ choque acústico (en telefonía) @ E : acoustic shock (in telephony) F : choc acoustique (en téléphonie) \ Conjunto de perturbaciones , pasajeras o permanentes del funcionamiento del oído , o del sistema nervioso , que puede sufrir el usuario de un auricular telefónico , como consecuencia de una brusca e importante elevación de la presión acústica producida por éste

la solución que da nuestro sistema para el pronombre *éste* es el sintagma nominal *el usuario de un auricular telefónico*, cuando la solución real sería únicamente: *auricular telefónico* ya que solamente un *auricular* (y no un *usuario*) es capaz de producir una *importante elevación de la presión acústica*.

Los resultados obtenidos se muestran en el apéndice F. En este caso sólo se han mostrado las frases que presentan los casos de anáfora estudiados en este trabajo. Tampoco se imprime su árbol sintáctico, sino tan sólo la secuencia de palabras separadas por blancos. Por cada antecedente o expresión anafórica también se muestra su estructura *paral* empleada en la resolución de la anáfora.

6.4 Análisis comparativo de resultados

Hemos aplicado sobre textos restringidos y no restringidos el algoritmo de resolución de la anáfora que aquí proponemos. Para afrontar el reto de los textos no restringidos hemos disminuido la cantidad de información con que cuenta el sistema: léxica, morfológica e información sintáctica reducida. Al reducir la cantidad de información que se utiliza para resolver la anáfora, pasamos a incluir nuestro sistema dentro de un grupo denominado *sistemas pobres en conocimiento*. Una aportación importante de nuestro sistema es que precisamente realiza un análisis sintáctico parcial del texto, extrayendo de modo automático la información indispensable para la resolución de la anáfora, y proponemos un modo de empleo de las restricciones c-dominio para que puedan aplicarse sobre información sintáctica incompleta

Aún reduciendo esta información, hemos conseguido un porcentaje de éxito o precisión del 83% en la resolución de la anáfora pronominal junto con una cobertura del 100%, valores que nos permiten realizar el siguiente estudio comparativo con otros sistemas

similares, aunque destacando que la comparación no se puede llevar a cabo en términos absolutos, ya que es evidente que cada uno de ellos ha trabajado sobre un idioma diferente (la mayoría han aplicado sus sistemas sobre el inglés, aunque también hay trabajos sobre polaco o alemán) y obviamente sobre textos de diferentes estilos:

- ◇ Hobbs en [82] obtuvo un porcentaje de éxito del 81,8% sobre 100 frases con diferentes ocurrencias de los pronombres *he*, *she*, *it*, y *they*, utilizando sólo información sintáctica y morfológica (concordancia en número y género). También es importante recordar que este algoritmo lo ejecutó manualmente, es decir, el análisis sintáctico completo se supone correcto y sin ambigüedades.

Nuestro algoritmo obtiene un 83% teniendo en cuenta también que no realiza un análisis sintáctico completo del texto y que no elimina manualmente las ambigüedades estructurales.

- ◇ Lappin y Leass en [95] trabajan únicamente sobre información sintáctica con un alto porcentaje de análisis correctos: un 85%, sobre manuales de informática.

Nuestro porcentaje de éxito es inferior (83%), pero nuevamente se justifica ya que utilizamos información sintáctica incompleta, mientras que Lappin y Leass no.

- ◇ Kennedy y Boguraev [93] proponen un algoritmo para la resolución de la anáfora pronominal que es una versión modificada del anterior trabajo de Lappin y Leass. Éste trabaja sobre la salida de un etiquetador (*part-of-speech tagger* o *POS tagger*) enriquecido tan sólo con las funciones gramaticales de determinadas palabras. No realiza un análisis sintáctico del texto. Obtienen un 75% de precisión que aunque es menor que el de Lappin y Leass, Kennedy y Boguraev lo justifican indicando que han trabajado sobre textos más variados y menos formales.

El 83% de nuestro algoritmo está muy cercano al 85% de Lappin y Leass teniendo en cuenta que realizamos un análisis sintáctico parcial del texto. Y también supera el 75% de Kennedy y Boguraev que sí que podría competir en igualdad de condiciones con nuestro sistema, también superándolo al permitir el trabajo sobre otros tipos de expresiones anafóricas además de los pronombres, y superándolo en la cantidad de información sintáctica que obtenemos automáticamente.

- ◇ En [157] Stuckardt propone un algoritmo para la resolución de la anáfora pronominal. Este algoritmo está basado principalmente en las restricciones sintácticas derivadas del trabajo de Chomsky en su *teoría de rección y ligamiento*, por lo que necesita un análisis sintáctico completo del texto. Stuckardt estima que su sistema resuelve aproximadamente un 90% de los pronombres aparecidos en textos sobre biografías de arquitectos.

Nuevamente aunque nuestra precisión es inferior, tampoco es lejana respecto al 90% “aproximado” del que habla Stuckardt, realizando un análisis sintáctico completo.

- ◇ En [118] Mitkov y Stys proponen otro sistema que necesita poca cantidad de conocimiento para la resolución de la anáfora pronominal en manuales técnicos tanto en inglés como en polaco. Utiliza la concordancia en número, género y persona como restricción y una serie de *indicadores de antecedente* (*antecedent*

indicators) a modo de preferencias. Este sistema se evaluó sobre un manual técnico en inglés de una fotocopidora Minolta obteniendo una precisión del 95,8%. Para el polaco también lo aplican obteniendo una precisión del 92,1%.

Aunque nuestro sistema ofrece un porcentaje inferior, hay que resaltar que nuestro sistema se puede aplicar a cualquier tipo de textos (es un sistema de propósito general), mientras que el propuesto por Mitkov y Stys utiliza información adaptada al texto sobre el que trabaja: indicadores de antecedente obtenidos tras un análisis previo del texto.

- ◇ En [158] Stuckardt también trabaja realizando un análisis sintáctico parcial, y obtiene un 82% de precisión sobre textos en alemán.
- ◇ En [13] Baldwin presenta un sistema de resolución de la anáfora pronominal denominado *CogNIAC* con un 90% de precisión y un alcance superior al 60%. Ampliándolo a un alcance del 100% obtuvo un 77,9% de precisión.

7. Conclusiones y trabajos futuros

En esta Tesis hemos tratado la resolución de la anáfora, uno de los problemas más difíciles pendientes de solución en el campo de estudio del procesamiento del lenguaje natural. Esta resolución de la anáfora la hemos integrado dentro de un sistema modular denominado SUP (Slot Unification Parser) desarrollado en Prolog que utiliza como mecanismo de almacenamiento de la información sintáctica el formalismo gramatical SUG o gramática de unificación de huecos. Dentro de este sistema SUP hemos añadido nuevas reglas a la gramática que permiten un análisis parcial del texto, de tal modo, que el mismo algoritmo de resolución de la anáfora que aquí proponemos se pueda aplicar tanto a textos restringidos como no restringidos, tanto mediante un análisis sintáctico completo, como mediante un análisis parcial.

En este capítulo resumimos las contribuciones descritas a través de la Tesis, comentamos resultados de experimentos realizados, presentamos conclusiones generales y sugerimos nuevas directrices para trabajos futuros.

7.1 Contribuciones

A continuación, vamos a exponer las contribuciones más destacadas realizadas en esta Tesis, y para ello las clasificamos en los siguientes apartados:

- ◆ Hemos propuesto un nuevo formalismo gramatical: gramática de unificación de huecos (SUG, Slot Unification Grammar) como una extensión de las gramáticas de cláusulas definidas (DCG, Definite Clause Grammar). La principal extensión de este formalismo gramatical es el uso de constituyentes opcionales dentro de cada regla gramatical, y prever la solución de determinados problemas lingüísticos como son la coordinación y yuxtaposición de constituyentes.
- ◆ Hemos desarrollado en Prolog un sistema modular denominado SUP (Slot Unification Parser). Este sistema lo componen una serie de módulos independientes entre sí: módulo de almacenamiento de la información sintáctica

mediante la gramática SUG, módulo del diccionario, módulo de análisis sintáctico, módulo de resolución de problemas lingüísticos y módulo de análisis semántico.

SUP es un sistema completo de procesamiento del lenguaje natural, cuyas características más importantes se pueden resumir en los siguientes puntos:

- ◊ Facilita la resolución modular de problemas lingüísticos (elipsis, extraposición, anáfora, etc.).
 - ◊ Relaja la relación entre cada uno de los componentes del sistema: gramática, diccionario, analizador sintáctico y semántico, y módulo de resolución de problemas lingüísticos.
 - ◊ Simplifica el trabajo con diferentes diccionarios, y la aplicación de análisis sintácticos parciales o relajados sobre textos no restringidos en los que no tenemos la seguridad de obtener módulos de análisis sintáctico lo suficientemente robustos.
 - ◊ Desarrolla un analizador eficiente. Este objetivo se consigue al reducir el número de reglas gramaticales y el número de símbolos no terminales. También se alcanzará este objetivo ya que será capaz de recordar lo que ha sido analizado en una frase y lo que no, disminuyendo el número de intentos de análisis de cada constituyente de la gramática. Sobre este objetivo se realizó la evaluación del comportamiento del sistema SUP ya mostrada en esta Tesis, en la que se observaba que obteníamos complejidades en su caso peor inferiores a n^3 , con n el número de palabras de la frase.
- ◆ Hemos llevado a cabo un estudio completo del fenómeno lingüístico correspondiente a la anáfora:
- ◊ Inicialmente partimos de la relación de la anáfora con otros temas de estudio del procesamiento del lenguaje natural: la elipsis, la extracción de información, la traducción automática y la resolución del ligamiento de sintagmas preposicionales. Mediante estas relaciones con otros campos se ha reseñado la importancia que tiene en la actualidad un correcto tratamiento de la anáfora.
 - ◊ Y a continuación se ha realizado una exhaustiva clasificación desde diferentes aspectos: en función del marco en que sucede, en función de la accesibilidad del antecedente, según el tipo de referencia y según el tipo de expresión anafórica. Son diferentes clasificaciones, todas ellas válidas y no excluyentes entre sí, ya que estas variaciones miden diferentes matices del concepto original y general de anáfora.
- Por ejemplo, en función del marco en que se trata la anáfora, ya sea éste la propia oración o todo el discurso, hablaremos de anáfora intraoracional y anáfora discursiva respectivamente. En función de la accesibilidad del antecedente: *anáfora morfosintáctica*, *semántica* y *pragmática*, de tal modo que la morfosintáctica será la que tenga una mayor accesibilidad, es decir, la que se derreferencia más fácilmente, y la pragmática la que tenga menor accesibilidad. También en función del tipo de referencia que se establezca entre la expresión anafórica y su antecedente hablaremos de *anáfora superficial* y *anáfora profunda*, en la que la profunda establece una relación de correferencia, mientras que la superficial no. Y finalmente podemos encontrarnos distintos tipos de expresiones anafóricas: pronombres, sintagmas nominales definidos, sintagmas nominales formados por el pronombre *uno* (*one*) y algunos modificadores, sintagmas nominales cuyo núcleo sea un adjetivo, adverbios y complementos circunstanciales. Cada una de estas expresiones anafóricas definirá un determinado tipo de anáfora que en

ocasiones se denomina de una manera específica: *anáfora pronominal*, *anáfora de tipo "one"*, *anáfora de tipo adjetivo*, *anáfora superficial numérica*, *anáfora verbal* y *referencias temporales o locales*.

- ◆ Debido a esta exhaustiva clasificación hemos podido centrar el objetivo principal de este trabajo: resolver la anáfora que sucede en un contexto lingüístico, considerando para su resolución no sólo la oración en la que se encuentra la expresión anafórica, sino todo el discurso (anáfora discursiva). Nos hemos centrado en la anáfora morfosintáctica, o sea, la de mayor accesibilidad de su antecedente, tratando tanto referencias superficiales como profundas, tanto relaciones de correferencia como de no correferencia, y hemos resuelto la anáfora pronominal y la de tipo adjetivo, es decir, expresiones anafóricas de tipo pronombre y sintagmas nominales cuyo núcleo sea un adjetivo. Ésta es precisamente una de las principales aportaciones que realizamos a la resolución de la anáfora, ya que la aproximación que proponemos nos permite el tratamiento de otros tipos de anáfora aparte de los habitualmente tratados de tipo pronominal.
- ◆ También hemos llevado a cabo un profundo estudio de las aproximaciones al tratamiento de la anáfora pronominal y de tipo adjetivo existentes actualmente. Nuestra propuesta de resolución de estos tipos de anáfora se encuadra entre los sistemas integrados democráticos con un enfoque basado en restricciones (eliminan antecedentes factibles de una determinada anáfora) y preferencias (seleccionan uno entre varios candidatos posibles). Como sistema integrado, se basa en el conocimiento, es decir, maneja una serie de fuentes de información que se consideran necesarias para el correcto tratamiento de la anáfora: morfológica, léxica, sintáctica y semántica. Y dentro de los sistemas integrados utiliza un enfoque democrático para coordinar estas fuentes de información, es decir, que aquellas entidades que pueden ser susceptibles de convertirse en antecedente pueden surgir por igual de aportaciones de la información morfológica, léxica, sintáctica o semántica.
- ◆ Hemos aplicado sobre textos restringidos y no restringidos el algoritmo de resolución de la anáfora propuesto. Para afrontar el reto de los textos no restringidos hemos disminuido la cantidad de información con que cuenta el sistema: léxica, morfológica e información sintáctica reducida. Al reducir la cantidad de información que se utiliza para resolver la anáfora, pasamos a incluir nuestro sistema dentro de un grupo denominado *sistemas pobres en conocimiento*. Una aportación importante de nuestro sistema es que precisamente realiza un análisis sintáctico parcial del texto, extrayendo de modo automático la información indispensable para la resolución de la anáfora, y proponemos un modo de empleo de las restricciones c-dominio para que puedan aplicarse sobre información sintáctica incompleta. Aún reduciendo esta información, hemos conseguido un porcentaje de éxito o precisión del 83% en la resolución de la anáfora pronominal junto con una cobertura del 100%.

7.2 Trabajos futuros

La extensión inmediata de nuestro trabajo será el aplicar nuestro sistema a otros textos no restringidos de diferentes géneros para poder establecer comparaciones de los resultados obtenidos (nuestro texto es un manual técnico). Al poder comparar resultados sobre la anáfora producida sobre distintos tipos de texto, nos permitirá medir la importancia de la inclusión en el algoritmo de información dependiente del tipo de texto que se va a tratar y valorar la mejora que introduce este tipo de información.

Otra extensión natural del trabajo realizado en esta Tesis, será la resolución de más tipos de anáfora como por ejemplo las descripciones definidas y las referencias locales y temporales. Para intentar la resolución de estos tipos de anáfora, la información semántica pasa a ocupar un papel fundamental, por lo que también se integraría en nuestro sistema la herramienta *WordNet* [108] como soporte de toda la información semántica necesaria para el tratamiento de la anáfora. Esta herramienta, se enmarca dentro del conjunto de diccionarios y corpus electrónicos que aportan información de naturaleza léxica, sintáctica y semántica. El motivo por el que no la hemos utilizado para este trabajo es que está disponible únicamente para el inglés, aunque hay proyectos que intentan trasladar la estructuración de su conocimiento a otros idiomas entre ellos el castellano (*EuroWordNet*).

Otra línea abierta de investigación es la integración de nuestro sistema SUP y su algoritmo de tratamiento de la anáfora sobre un sistema de extracción de información, en el que como ya mostramos en la introducción de esta Tesis, comparten una serie de problemas comunes relativos al análisis del discurso, como son el análisis de sintagmas nominales complejos, el análisis semántico de los mismos y la resolución de correferencias entre ellos. El sistema de extracción de información sobre el que pretendemos integrar nuestro sistema es EXIT [99], sistema que trabaja sobre el dominio restringido de las escrituras notariales.

Otra línea de investigación sería la integración de nuestro sistema sobre una aplicación de traducción automática, en la que como ya hemos descrito anteriormente es fundamental un tratamiento correcto de la anáfora, ante las posibles diferencias existentes entre las expresiones anafóricas de distintas lenguas.

7.3 Resumen final

El principal propósito de esta Tesis ha sido investigar en el campo de la resolución de fenómenos lingüísticos que ocurren en el lenguaje natural, concretamente dentro del problema de la anáfora. La principal contribución ha sido el desarrollo del sistema completo de procesamiento del lenguaje natural SUP, en el que junto a un nuevo formalismo gramatical, SUG, hemos integrado un módulo de resolución de la anáfora pronominal y de tipo adjetivo que sucede tanto en textos restringidos como en no restringidos, sistema que es capaz de realizar tanto un análisis sintáctico completo como un análisis sintáctico parcial del texto.

Hemos realizado dos experimentos. En el primero se proponía mostrar la aplicación del sistema de tratamiento de la anáfora sobre textos restringidos. Y el segundo

experimento aplicaba el mismo sistema sobre textos no restringidos. Según los resultados obtenidos de estos experimentos, podemos concluir que hemos alcanzado los objetivos que inicialmente nos planteamos en este trabajo, ofreciendo un sistema modular de procesamiento del lenguaje natural que permite la resolución de la anáfora.

Universitat d'Alacant
Universidad de Alicante

8. Bibliografía

- [1] Abney, S. Part-of-Speech Tagging and Partial Parsing. In *Corpus-based methods in language and speech processing*. S. Young and G. Bloothoof, Eds., Kluwer Academic. 1997.
- [2] Alcina, J., and Blecua, J.M. *Gramática española*. Ariel Ed. Barcelona, 1975.
- [3] Allen, J. *Natural Language Understanding*. The Benjamin / Cummings Publishing Company, Inc., 1995.
- [4] Alshawi, H. *Memory and Context for Language Interpretation*. Cambridge University Press, 1987.
- [5] Amores, J.G. *A Lexical-Functional Grammar Machine Translation System for Medical Abstracts*. Tesis doctoral. Universidad de Sevilla, 1992.
- [6] Ariel, M. Referring and accessibility. *Journal of Linguistics*, 24 (1998). pp 65-87.
- [7] Arrarte, G. SSG: una gramática del español basada en el formalismo Slot Grammar. En *Actas del VIII Congreso SEPLN* (1993).
- [8] Azzam, S. An Algorithm to Co-Ordinate Anaphora Resolution and PPS Disambiguation Process. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, EACL'95* (1995).
- [9] Azzam, S. Anaphors, pps and disambiguation process for conceptual analysis. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95* (San Mateo, 1995).
- [10] Azzam, S. *Computation of Ambiguities (Anaphors and PPs) in NL texts*. CLAM: The prototype. Ph.D. thesis. Paris Sorbonne University, 1995.
- [11] Azzam, S. Resolving Anaphors in Embedded Sentences. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, ACL'96* (Santa Cruz, Ca., 1996).

- [12] Baldwin, B. *CogNIAC: a discourse processing engine*. Ph.D. dissertation. University of Pennsylvania, Department of Computer and Information Science, 1995.
- [13] Baldwin, B. *CogNIAC: high precision coreference with limited knowledge and linguistic resources*. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [14] Baldwin, B., Reynar, J., Collins, M., Eisner, J., Ratnaparkhi, A., Rosenzweig, J., Sarkar, A., and Srinivas. University of Pennsylvania: Description of the University of Pennsylvania system used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference* (1995).
- [15] Bobrow, D., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., and Winograd, T. Gus, a frame driven dialog system. *Artificial Intelligence*, 8 (1977). pp. 155-173.
- [16] Bobrow. *The high-school algebra problem answering system STUDENT with natural language input*, 1964.(*)
- [17] Botley, S. Comparing demonstrative features in three written English genres. In *Proceedings of the Discourse Anaphora And Resolution Colloquium, DAARC96* (1996).
- [18] Brennan, S., Friedman, M., and Pollard, C. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics, ACL'87* (1987). pp. 155-162.
- [19] Bresnan, J. *The Mental Representation of Grammatical Relations*. MIT Press, 1982
- [20] Brill E., and Mooney R.J. An Overview of Empirical Natural Language Processing. *AI Magazine*, 18,4 (winter 1997).
- [21] Brill, E. *Transformation-Based Learning*. Ph.D. thesis. University of Pennsylvania, 1993.
- [22] Brown, G., and Yule, G. *Discourse Analysis*. Cambridge University Press, 1983.
- [23] Brucart, J.M. *La elisión sintáctica en español*. Universitat Autònoma de Barcelona, 1987.
- [24] Carbonell, J.G., and Brown, R.D. Anaphora Resolution: a Multi-Strategy Approach. In *Proceedings of the 12th International Conference on Computational Linguistics, COLING '88* (Budapest, 1988). Vol I, pp. 96-101.
- [25] Carpenter, B. *ALE. The Attribute Logic Engine user's guide*. Laboratory for Computational Linguistics, Carnegie Mellon University, Pittsburgh, Pa. 1993.
- [26] Carroll, J. Relating Complexity to Practical Performance in Parsing with Wide-Coverage Unification Grammars. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, ACL'94* (1994).

- [27] Carter, D. Control issues in anaphor resolution. *Journal of Semantics*, 7 (1990). pp. 435-454.
- [28] Carter, D. *Interpreting Anaphors in Natural Language*. Ellis Horwood, Ed., Chichester, UK, 1987.
- [29] Carvalho, A. Logic grammars and pronominal anaphora. In *Proceedings of the Discourse Anaphora And Resolution Colloquium, DAARC96* (1996).
- [30] Charniak, E. *Toward a Model of Children's Story Comprehension*. Cambridge, Mass: MIT A.I. Lab TR-266, 1972.
- [31] Chierchia, G. Anaphora and Dynamic Binding. *Linguistics and Philosophy*, 15 (1992). pp. 111-183.
- [32] Chomsky, N. *Aspects of the Theory of Syntax*. MIT Press, 1965.
- [33] Chomsky, N. *Knowledge of Language*. Praeger, New York, 1986.
- [34] Chomsky, N. *La nueva Sintaxis: Teoría de rección y el Ligamiento*, 1988.
- [35] Chomsky, N. *Lectures on Government and Binding*. Foris Publications, Dordrecht, 1981.
- [36] Chomsky, N. *Syntactic Structures*. Mouton, The Hague, 1957.
- [37] Church, K. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of ANLP 88* (Austin, TX, 1988).
- [38] Climent, S. Semantics of Portions and Partitive Nouns for NLP. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING'96* (Copenhagen, Denmark, 1996). Vol. I, pp. 243-248.
- [39] Collins, M. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, ACL'96* (1996).
- [40] Colmerauer, A. *Metamorphosis Grammars*. Informe Interno. Univ. d'Aix-Marseille II, 1975.
- [41] Connolly, D., Burger, J., and Day, D. A machine learning approach to anaphoric reference. In *Proceedings of the International Conference on New Methods in Language Processing* (UMIST, Manchester, 1994).
- [42] Covington, M. *Natural Language Processing for Prolog Programmers*. Prentice Hall, 1994.
- [43] Cowie, J., and Lehnert, W. Information Extraction. *Communications of the ACM*, January 1996.

- [44] Cullingford, R. E. *Inside Computer Understanding: Five Programs Plus Miniatures*. Lawrence Erlbaum, Hillsdale, NJ. In R. C. Schank and C. K. Riesbeck, Eds. 1981.
- [45] Dagan, I., and Itai, A. Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of the 13th International Conference on Computational Linguistics, COLING'90* (1990).
- [46] Dahl, V., and McCord, M. Treating Coordination in Logic Grammars. *AJCL*, 9, 2 (1983).
- [47] Dahlbäck, N. *Representations of Discourse-Cognitive and Computational Aspects*. Ph.D. thesis. Department of Computer and Information Science, Linköping University, Linköping, Sweden, 1991.
- [48] Declerck, T. Dealing with Cross-Sentential Anaphora Resolution in ALEP. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING'96* (Copenhagen, Denmark, 1996). Vol. I, pp. 280-285.
- [49] Di Eugenio, B. The discourse functions of Italian subjects: a centering approach. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING'96* (Copenhagen, Denmark, 1996). Vol. I, pp. 352-357.
- [50] Emele, M.C., and Zajac, R. Typed unification grammars. In *Proceedings of the 13th International Conference on Computational Linguistics, COLING'90* (1990).
- [51] Ersan, E., and Akman, V. *Focusing for pronoun resolution in english discourse: an implementation*. BILKENT UNIVERSITY. Department of Computer Engineering and Information Science. Technical Report BU-CEIS-94-29, 1994.
- [52] Ferrández, A., Palomar, M., Moreno, L. Anaphor resolution in unrestricted texts with partial parsing. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING - ACL* (1998).
- [53] Ferrández, A., Moreno, L., and Palomar, M. Un formalismo para el tratamiento gramatical de la coordinación: Gramática de Unificación de Huecos. *Novatica*, 115 (1995).
- [54] Ferrández, A., Moreno, L., Palomar, M., and Peral, J. Un método de resolución de la anáfora discursiva mediante la unificación. En *Actas del Congreso de la Asociación Española Para la Inteligencia Artificial, CAEPIA'97* (Málaga, 1997).
- [55] Ferrández, A., Moreno, L., Palomar, M., and Peral, J. Un método de resolución de la anáfora discursiva en textos no restringidos mediante la unificación. *Inteligencia Artificial*, 4 (1/98).
- [56] Ferrández, A., Palomar, M., and Moreno, L. A computational approach to pronominal anaphora, one-anaphora and surface count anaphora. In *Proceedings of the Discourse Anaphora And Resolution Colloquium, DAARC98* (1998).

- [57] Ferrández, A., Palomar, M., and Moreno, L. El problema del núcleo del sintagma nominal: ¿elipsis o anáfora?. *Procesamiento del Lenguaje Natural*, 20 (Junio 1997).
- [58] Ferrández, A., Palomar, M., and Moreno, L. Implementación de un diccionario en Prolog mediante un árbol de letras comprimido. *Novatica - Lengua y Tecnologías de la Información*, 133 (Mayo-Junio 1998).
- [59] Ferrández, A., Palómar, M., and Moreno, L. Slot Unification Grammar and anaphora resolution. In *Proceedings of Recent Advances in Natural Language Resolution, RANLP'97* (1997).
- [60] Ferrández, A., Palomar, M., and Moreno, L. Slot Unification Grammar. In *Proceedings of APPIA-GULP-PRODE'97* (Grado, Italy, 1997).
- [61] Ferrández, A., Peral, J., Martínez, P., Saiz, M., and Romero, R. Resolución de la extraposición a izquierdas con las Gramáticas de Unificación de Huecos. *Procesamiento del Lenguaje Natural*, 21 (Julio 1997).
- [62] Fischer, I., Geistert, B., and Görz, G. Incremental anaphora resolution in a chart-based semantics construction framework using λ -DRT. In *Proceedings of the Discourse Anaphora And Resolution Colloquium, DAARC96* (1996).
- [63] Fligelstone, S. Developing a scheme for annotating text to show anaphoric relations. *New directions in English language corpora: methodology, results, software development*, 9 (1992).
- [64] Franchini, E. *Las condiciones gramaticales de la coordinación copulativa en español*. Romanica Helvetica. Vol. 102. Francke Verlag Bern, 1986.
- [65] Gaizauskas, R., and Humphreys, K. Quantitative evaluation of coreference algorithms in an information extraction system. In *Corpus-based and computational approaches to discourse anaphora*. S.P. Botley and A.M. McEnery, Eds., UCL Press. 1997.
- [66] Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H., and Wilks, Y. Description of the LaSIE system as used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference, MUC-6* (1995). pp. 207-220.
- [67] Gazdar, G., Klein, E., Pullum, G., and Sag, I. *Generalized Phrase Structure Grammar*. Oxford, England: Blackwell Publishing and Cambridge, Mass.: Harvard University Press, 1985.
- [68] Götz, T., and Meurers, W.D. Compiling HPSG type constraints into definite clause programs. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, ACL'95* (Massachusetts, 1995).
- [69] Grober, Beardsley and Caramazza. *The implicit verb causality or causal valence*, 1978.
- [70] Grosz, B. Focusing and description in Natural Language Dialogues. In *Elements of Discourse Understanding*. Cambridge: Cambridge University Press. 1981.

- [71] Grosz, B. The representation and use of focus in a system for understanding dialogs. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence, IJCAI* (Cambridge, MA., 1977).
- [72] Grosz, B., and Sidner, C. Attentions, intentions and the structure of discourse. *Computational Linguistics*, 12, 3 (1986). pp. 175-204.
- [73] Grosz, B., Aravind, J., and Weinstein, S. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21, 2 (1995). pp. 203-225.
- [74] Grosz, B., Joshi, A., and Weinstein, S. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics, ACL'83* (1983). pp. 44-50.
- [75] Grosz, B.J., Joshi, A.K., and Weinstein, S. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21, 2 (1995). pp. 203-225.
- [76] Hardt, D. Centering in Dynamic Semantics. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING'96* (Copenhagen, Denmark, 1996). Vol. I, pp. 519-524.
- [77] Heim, Y. *The semantics of definite and indefinite noun phrases*. Dissertation, University of Massachusetts, Amherst, 1982.
- [78] Hindle, D. A parser for text corpora. In *Computational Approaches to the Lexicon*. A. Zampolli, Ed., Oxford University Press, Oxford/New York. 1994.
- [79] Hindle, D. *User manual for Fidditch*. Technical Memorandum #7590-142, Naval Research Laboratory, 1983.
- [80] Hirst, G. *Anaphora in Natural Language Understanding*. Berlin:Springer-Verlag, 1981.
- [81] Hobbs, J. Coherence and co-reference. *Cognitive Science*, 3, 1 (1979). pp. 67-90.
- [82] Hobbs, J. *Pronoun Resolution*. Nueva York: Research Report # 76-1, Department of Computer Sciences. City College, City University of New York, 1976.
- [83] Hobbs, Jerry, R., Douglas, E., Appelt, Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M. FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In *Finite State Devices for Natural Language Processing*. E.Roche and Y.Schabes, Eds., MIT Press, Cambridge, Massachusetts. 1996.
- [84] Joshi, A.K., and Srinivas, B. Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING'94* (Kyoto, 1994).

- [85] Kameyama, M. A Property-Sharing Constraint in Centering. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics, ACL'86* (1986). pp. 200-206.
- [86] Kameyama, M. Intrasentential Centering: A case Study. In *Centering in Discourse*. E. Prince, A. Joshi, and L. Walker, Eds., Oxford University Press. 1997.
- [87] Kameyama, M. Recognizing referential links: an information extraction perspective. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [88] Karlsson, F., Voutilainen, A., Heikkilä, J., and Anttila, A. Eds. *Constraint Grammar: A language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin & New York, 1995.
- [89] Kay, M. Parsing in Functional Unification Grammar. In *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*. Cambridge, England: Cambridge University Press. 1985.
- [90] Kay, M. *Unification Grammar*. Technical report, Xerox Palo Alto Research Center, Palo Alto, California, 1983.
- [91] Kehler, A. A discourse copying algorithm for ellipsis and anaphora resolution. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, EACL'93* (1993). pp. 203-212.
- [92] Kehler, A. The effect of establishing coherence in ellipsis and anaphora resolution. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, ACL'93* (1993). pp. 62-69.
- [93] Kennedy, C., and Boguraev, B. Anaphora for Everyone: Pronominal Anaphora Resolution without a Parser. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING'96* (Copenhagen, Denmark, 1996). Vol. I, pp. 113-118.
- [94] Kniht, K. Unification: A Multidisciplinary Survey. *ACM Computing Surveys*, 21, 1 (1989).
- [95] Lappin, S., and Leass, H. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20, 4 (1994). pp. 535-561.
- [96] Lappin, S., and McCord, M. A syntactic filter on pronominal anaphora in Slot Grammar. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, ACL'90* (1990). pp. 135-142.
- [97] Lappin, S., and McCord, M. Anaphora resolution in Slot Grammar. *Computational Linguistics*, 16, 4 (December 1990).
- [98] Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. Cyc: Toward programs with common sense. *Communications of the ACM*, 33, 8 (1990). pp. 30-49.

- [99] Llopis, F.; Muñoz, R.; Suárez, A.; Montoyo, A.; Palomar, M.; Ferrández, A.; Peral, J.; Martínez-Barco, P.; Romero, R.; Saiz, M. *Sistema EXIT*. Report Interno del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante, 1998.
- [100] Luperfoy, S. *Discourse Pegs: A Computational Analysis of Context-Dependent Referring Expressions*. Ph.D. thesis. Department of Linguistics, The University of Texas at Austin, Austin, TX, 1991.
- [101] Martínez-Barco, P., Peral, J., Ferrández, A., Moreno, L., and Palomar, M. Analizador parcial SUPP. In *Proceedings of the 6th Ibero-American Conference on Artificial Intelligence, IBERAMIA '98* (Lisbon, Portugal, 1998).
- [102] McCord, M. Modular Logic Grammars. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, ACL '90* (1985).
- [103] McCord, M.C. Slot Grammar: a system for simpler construction of practical natural language grammars. In *Natural Language and Logic, International Scientific Symposium*. R.Studer, Ed., Lecture Notes in Computer Science, Springer Verlag. 1990.
- [104] McCord, M.C. The Slot Grammar System. In *Unification in Grammar*. J.Wedekind and C. Rohrer, Eds., MIT Press. 1992.
- [105] McCord, M.C. Using Slots and modifiers in logic grammars for natural language. *Artificial Intelligence*, 18 (1982). pp. 327-367.
- [106] McEnery, A., Tanaka, I., and Botley, S. Corpus annotation and reference resolution. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [107] Meya, M., and Hubert, W. *Lingüística computacional. Barcelona, Teide. 1986*.
- [108] Miller, G. *Five papers on WordNet*. Technical Report CSL Report 43, Cognitive Science Laboratory, Princeton University, 1993.
- [109] Mitkov, R. A new approach for tracking center. In *Proceedings of the International Conference New Methods in Language Processing* (Manchester, 1994).
- [110] Mitkov, R. An integrated model for anaphora resolution. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING '94* (Kyoto, Japan, 1994).
- [111] Mitkov, R. An uncertainty reasoning approach to anaphor resolution. In *Proceedings of the Natural Language Pacific Rim Symposium* (Seoul, Korea, 1995).
- [112] Mitkov, R. Anaphor resolution in natural language processing and machine translation. In *Proceedings of the Discourse Anaphora And Resolution Colloquium, DAARC96* (1996).

- [113] Mitkov, R. Anaphor resolution: a combination of linguistic and statistical approaches. In *Proceedings of the Discourse Anaphora And Resolution Colloquium, DAARC96* (1996).
- [114] Mitkov, R. Factors in anaphora resolution: they are not the only things that matter. A case study based on two different approaches. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [115] Mitkov, R. How far are we from (semi-) automatic annotation of anaphoric links in corpora. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [116] Mitkov, R. Pronoun resolution: the practical alternative. In *Corpus-based and Computational Approaches to Discourse Anaphora*. S. Botley and T. McEnery, Eds., Univ. College London Press. 1997.
- [117] Mitkov, R. Two engines are better than one: generating more power and confidence in the search for the antecedent. In *Proceedings of the Recent Advances in Natural Language Resolution, RANLP* (1995).
- [118] Mitkov, R., and Stys, M. Robust reference resolution with limited knowledge: high precision genre-specific approach for English and Polish. In *Proceedings of the Recent Advances in Natural Language Resolution, RANLP* (1997).
- [119] Mollá, D. *On searching the antecedent of an anaphora*. Department of Linguistics. Edinburgh, 1992.
- [120] Moreno, L. *Formalismos Lógicos para el Análisis e Interpretación Oracional del Lenguaje Natural*. Tesis Doctoral. Universidad Politécnica de Valencia, 1993.
- [121] Moreno, L., and Palomar, M. "Semantic Constraints in a Syntactic Parser: Queries-Answering to Database". Database and Expert Systems Applications, Springer-Verlag, 1992.
- [122] Moreno, L., Andrés, F., and Palomar, M. Incorporar Restricciones Semánticas en el Análisis Sintáctico: IRSAS. *Procesamiento del Lenguaje Natural*, 12 (1992).
- [123] Muskens, R. *Combining montague semantics and discourse representation*. Linguistics and Philosophy, 1996.
- [124] Nakaiwa, H., and Shirai, S. Anaphora Resolution of Japanese Zero Pronouns with Deictic Reference. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING'96* (Copenhagen, Denmark, 1996). Vol. II, pp. 812-817.
- [125] Nasukawa, T. Robust method of pronoun resolution using full-text information. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING'94* (1994).

- [126] Okumura Manabu, Tamura Kouji. Zero Pronoun Resolution in Japanese Discourse Based on Centering Theory. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING '96* (Copenhagen, Denmark, 1996). Vol. II, pp. 871-876.
- [127] Palomar, M. *Aportaciones a la resolución de la elipsis en lenguaje natural utilizando técnicas incrementales*. Tesis Doctoral. Universidad Politécnica de Valencia, 1996.
- [128] Palomar, M., Ferrández, A., and Moreno, L. Aportaciones a la Resolución de la elipsis en la coordinación. *Procesamiento del Lenguaje Natural*, 16 (1995).
- [129] Palomar, M., Moreno L., Romero I., and Sebastian, J. Formalización de la coordinación mediante gramáticas de Huecos. *Procesamiento del Lenguaje Natural*, 15 (1994).
- [130] Partee, B. H. Opacity, coreference, and pronouns. In *Semantics of Natural Language*. D. Davidson and G. Harman, Eds., D. Reidel Publishing Company, Dordrecht, Holland. 1972.
- [131] Pereira, F. Extraposition Grammars. *American Journal of Computational Linguistics*, 7, 4, 1981.
- [132] Pereira, F., and Warren, D. Definite Clause Grammars for Language Analysis- A Survey of the Formalism and a Comparison with Augmented Transition Networks. *Artificial Intelligence*, 13, 1980.
- [133] Pereira, L.M. *ORBI-An Expert System for Enviromental Resource Evaluation through Natural Language*. Informe Interno. Universidad Nova de Lisboa, 1982.
- [134] Poesio, M., Vieira, R., and Teufel, S. Resolving bridging references in unrestricted text. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [135] Pollar, C. *Lecture Notes on Head Driven Phrase Structure Grammar*. Unpublished manuscript, Center for the Study of Language and Information, Stanford University, Stanford, Calif. February, 1985.
- [136] Pollar, C., and Sag, I. *Head Driven Phrase Structure Grammar*. Chicago University Press, 1995.
- [137] Pollard, C., and Sag, I.A. Anaphors in English and the Scope of Binding Theory. *Linguistics and Philosophy*, 23, 2 (1992). pp. 261-303.
- [138] Popowich, F. *Reflexives and Tree Unification Grammar*. Ph.D. thesis. University of Edinburgh, 1988.
- [139] Ramshaw, L.A., and Marcus, M.P. Text Chunking using Transformation-Based Learning. In *Proceedings of ACL Third Workshop on Very Large Corpora* (1995). pp.82-94.

- [140] Real Academia Española. *Esbozo de una nueva gramática de la lengua española*. Espasa-Calpe. Madrid, 1973.
- [141] Reinhart, T. *Anaphora and Semantic Interpretation*. Croom Helm Backenham, Kent, 1983.
- [142] Rich, E., and LuperFoy, S. Anaphora architecture for anaphora resolution. In *Proceedings of the Second Conference on Applied NLP* (1988).
- [143] Rico, C. *Aproximación estadístico-algebraica al problema de la resolución de la anáfora en el discurso*. Tesis Doctoral. Universidad de Alicante, 1994.
- [144] Ristad, E.S. *The anaphora problem*. Department of Computer Science, Princeton University, Academic Press, Inc., 1993.
- [145] Rocha, M. Corpus-based study of anaphora in English and Portuguese. In *Corpus-based and computational approaches to discourse anaphora*. S.P. Botley and A.M. McEnery Eds., UCL Press. 1997.
- [146] Rocha, M. Supporting anaphor resolution in dialogues with a corpus-based probabilistic model. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [147] Rosenchein, S.J., and Shieber, S.M. Translating English into Logical Form. In *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics ACL '82* (Toronto, Ontario, Canada, 1982).
- [148] Schwarz, C. Automatic Syntactic Analysis of Free Text. *JASIS*, 41, 6 (1990). pp. 408-417.
- [149] Shalom, L., and Nissim, F. E-Type pronouns, I-sums, and donkey anaphora. *Linguistics and Philosophy*, 17 (1994). pp. 391-428.
- [150] Shieber, S. An Introduction to Unification-Based Approaches to Grammar. *CSLI Lecture Notes*, 4 (1986).
- [151] Shieber, S.M., Uszkoreit, H., Pereira, F., Robinson, J., and Tyson, M. *The formalism and Implementation of PATR-II*. Research on Interactive Acquisition and Use of Knowledge. Artificial Intelligence Center, SRI International, Menlo Park, Calif, 1983.
- [152] Sidner, C. Focusing for interpretation of pronouns. *American Journal of Computational Linguistics*, 7 (1981). pp. 217-231.
- [153] Sidner, C. Focusing in the comprehension of definite anaphora. In *Computational Models of Discourse*. Brady y Berwick, Eds., MIT. 1983.
- [154] Sidner, C. L. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*. Ph.D. thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1979.

- [155] Strube, M. Processing Complex Sentences in the Centering Framework. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, ACL '96* (Santa Cruz, Ca., 1996).
- [156] Strube, M., and Hahn, U. Functional centering. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, ACL '96* (Santa Cruz, Ca., 1996).
- [157] Stuckardt, R. Anaphor Resolution and the Scope of Syntactic Constraints. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING '96* (Copenhagen, Denmark, 1996). Vol. II, pp. 937-942.
- [158] Stuckardt, R. Resolving anaphoric references on deficient syntactic descriptions. In *Proceedings of ACL/ EACL workshop on Operational factors in practical, robust anaphor resolution* (Madrid, 1997).
- [159] Suri, L.Z., and McCoy, K.F. RAFT / RAPR and Centering: A comparison and discussion of problems related to processing complex sentences. *Computational Linguistics*, 20, 2 (1994). pp. 301-317.
- [160] Sutcliffe, R. *A Parallel Distributed Processing Approach to the Representation of Knowledge for Natural Language Understanding*. Tesis doctoral. Universidad de Essex, 1989.
- [161] Vilain, M, and Palmer, D. Transformational-Based Bracketing: Fast Algorithms and Experimentals Result". In *Proceedings of ESSLLI 96 Workshop on Robust Parsing* (1996). pp. 93-102.
- [162] Vouitilainen, A., and Padró, L. Developing a hybrid NP parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing* (Washington D.C., USA, 1997).
- [163] Vouitilainen, A., Heikkilä, J., and Anttila, A. *Constraint Grammar of English. A performance-oriented Introduction*. Publications 21, Department of General Linguistics, University of Helsinki, 1992.
- [164] Walker, M.A., Iida, M., and Cote, S. Japanese Discourse and the Process of Centering. *Computational Linguistics*, 20, 2 (1994). pp. 193-232.
- [165] Webber, B. L. *A Formal Approach to Discourse Anaphora*. Ph.D. thesis. Division of Applied Mathematics, Harvard University, Cambridge, MA, 1978.
- [166] Weizenbaum, J. ELIZA: a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9 (1996). pp. 36-45.
- [167] Wilks, Y. A preferential pattern-seeking semantics for natural language inference. *Artificial Intelligence*, 6 (1975). pp. 53-74.

- [168] Williams, S., Harvey, M., and Preston, K. Rule-based reference resolution for unrestricted text using part-of-speech tagging and noun phrase parsing. In *Proceedings of the Discourse Anaphora And Resolution Colloquium, DAARC96* (1996).
- [169] Winograd, T. A procedural model of language understanding. In *Readings in Natural Language*. Grosz et al, Eds., California: Morgan Kaufman. 1986.
- [170] Winograd, T. *Understanding Natural Language*. Academic Press, Nueva York, 1972.
- [171] Woods, W. Lunar rocks in natural English: Explorations in natural language question answering. In *Linguistic Structures Processing*. A. Zampolli, Ed., Nueva York: Elsevier. 1977.
- [172] Woods, W., Kaplan, M., and Nash-Webber, B. *The Lunar Sciences Information System*. Cambridge, Mass: Bolt, Beranek y Newman Inc., 1972.

Apéndice A: gramática *SUG* restringida

```

%----- Orac. declarativas -----
coordinated(
  orac(Numero,Genero,Persona),
  oracSimple(_,_),
  -                                     % No hay ninguna restricción para la coordinación
).
%----- Frases simples -----
oracSimple(Numero,Genero,Persona) ++>
  << SN:sn(Numero,Genero,Persona,_,suj) >>,
  sv(Numero,Genero,Persona,Tipo),
  { ifthen( (Tipo==impers ; Tipo==impersOD),
    noExiste(SN) ) }.                                     % Orac. impersonal no tiene suj
%----- SINTAGMA NOMINAL -----
sn(Numero,Genero,Persona,Tipo,Funcion) ++>
  SNC:snCoordinado(_,_ ,Tipo,Funcion),
  { ngpResultante(SNC, Numero,Genero,Persona) }.         % Halla Numero,Genero y Persona resultante

coordinated(
  snCoordinado(Numero,Genero,Persona,Tipo,Funcion),
  snSimple(_,_ ,Funcion),
  -                                     % No hay ninguna restricción para la coordinación
).

snSimple(Numero,Genero,Persona,Tipo,Funcion) ++>
  poda(sn,Tipo,Funcion),                                 % Obtener TipoSn, FuncSn o podar
  snSimplePodado(Numero,Genero,Persona,Tipo,Funcion).

/*----- S.N.SUSTANTIVO o ADJETIVO cal. -----
  Sirve tanto para funcion de sujeto como de complemento
  %== TODOS LOS MAS DE DOS MUY GRANDES Y VIEJOS PERROS MIOS PINTADOS DE
  ALICANTE QUE VIENEN A JUGAR
  Estr. sust: Det1,Det2, AdjCal1,sust, pronPos,AdjCal2,   sp,aposis,ger,orRel
  Estr. adj : Det1,Det2, AdjCal,   pronPos,               sp,aposis,ger,orRel
*/

```

```

fusion(
  (snSimplePodado(Numero,Genero,terc,sustAdj,Funcion) ++> % SN tipo sustantivo común o propio
    <<DET1:det(Numero,Genero)>>,
    <<DET2:det2(Numero,Genero,TipoDet2)>>,
    {fifthen(TipoDet2==ord, existe(DET1))},
    %== EL PRIMER libro (obligatorio det.delante del adj.ordinal)

    ( <<adyAdj(Numero,Genero)>>,
      SUST:sust(Numero,Genero,Tipo),
      %== (* el perro de HERMANA juega, PERO SE ACEPTA POR: el perro de COLOR GRIS juega
      {fifthen((Tipo==comun,Funcion==suj), (existe(DET1) ; existe(DET2)))},
      <<pron(Numero,Genero,_,pos,compl) >>,
      <<adyAdj(Numero,Genero)>> ),

    << sp(sn,ISP) >>, %== el gato gris DE TU HERMANA Y DE TU CUÑADO
    << aposicion(Numero,Genero) >>, %== el gato gris, EL HIJO DEL GATO DE TU HERMANA,...
    << gerundioSN >>, %== el perro gris de tu hermana COMIENDO CARNE...
    << oracRelat >> ),

  (snSimplePodado(Numero,Genero,terc,sustAdj,Funcion) ++> % SN tipo adjetivo
    <<DET1:det(Numero,Genero)>>,
    <<DET2:det2(Numero,Genero,TipoDet2)>>,
    {fifthen(TipoDet2==ord, existe(DET1))}, %== EL PRIMER libro (obligatorio det.delante del adj.ordinal)

    ( adyAdj(Numero,Genero),
      _
      {fifthen(Funcion==suj, (existe(DET1) ; existe(DET2)))},
      <<pron(Numero,Genero,_,pos,compl)>>,
      _),

    << sp(sn,ISP) >>, %== el gato gris DE TU HERMANA Y DE TU CUÑADO
    << aposicion(Numero,Genero) >>, %== el gato gris, EL HIJO DEL GATO DE TU HERMANA,...
    << gerundioSN >>, %== el perro gris de tu hermana COMIENDO CARNE...
    << oracRelat >> )
  ).
%----- SN con PRONOMBRE funcion SUJETO -----
snSimplePodado(Numero, Genero, Persona, pron, suj) ++>
  pron(Numero,Genero,Persona,TipoPron,suj), %== ELLOS, ESTOS, ALGUNOS, DOS quieren,
                                             %== QUIEN quiera
  {TipoPron \== rel, TipoPron \== int}, %== * QUE, QUIEN
  << oracRelat >>, %== YO QUE MI HIJA LE GUSTA EL FUTBOL
  << gerundioSN >>, %== yo JUGANDO AL FUTBOL disfruto
%----- SN con PRONOMBRE funcion SP -----
snSimplePodado(Numero, Genero, Persona, pron, sp) ++>
  pron(Numero,Genero,Persona,TipoPron,sp), %== por EL, de ELLOS
  {TipoPron \== rel, TipoPron \== int}, %== * QUE, QUIEN
  << oracRelat >>, %== por ti QUE NO VINISTE
%----- SN con VERBO INFINITIVO -----
fusion(
  (snSimplePodado(sing, masc, terc, verbalInf, IF) ++>
    << art(sing,masc,det) >>, %== fumar, EL fumar
    verbo(sing,I1,infinitivo,I2),
    << sp(_,_) >> ), %== jugar AL FUTBOL, vivir POR JUGAR

```

```

(snSimplePodado(sing, masc, terc, verbalnf, IF) ++>
  << art(sing,masc,det) >>,
  verbo(sing,l1,infinitivo,l2),
  << sn(____,sustAdj,_) >> )
).
%----- SINTAGMAS VERBALES
% SV copul/trans/transIntrans/impers/impersOD
% Los verbos transitivos y copulativos se comprueban en ELIPSIS
sv(Numero,Genero,Persona,TipoSV) ++>
  << adv(af) >>,
  << PRON:pron(____,pers,compl) >>,
  verbo(Numero,Persona,Tiempo,TipoSV),
  { Tiempo != infinitivo, Tiempo != gerundio },
  << CC1:cc(NumCC1,GenCC1,PersCC1) >>,
  { ifthen( ( TipoSV == copul, existe(CC1) ),
    (NumCC1=Numero, GenCC1=Genero, PersCC1=Persona) ) },
  << SP1:sp(oi) >>,
  << SN:sn(NumSn,GenSn,____,FuncSn) >>,
  { ifthen( ( TipoSV == copul, existe(SN) ),
    (NumSn=Numero, GenSn=Genero) ),
    ifthen(nonvar(FuncSn), FuncSn != sp) },
  << SP2:sp(oi) , noExiste(SP1) >>,
  << CC2:cc(NumCC2,GenCC2,PersCC2) >>,
  { ifthen( ( TipoSV == copul, existe(CC2) ),
    (NumCC2=Numero, GenCC2=Genero, PersCC2=Persona) ),
    ifthenelse( TipoSV == impers,
      ( noExiste(PRON), noExiste(SP1), noExiste(SN), noExiste(SP2) ),
      %Sino
      ifthen( TipoSV == impersOD,
        ( noExiste(SP1), noExiste(SP2) )
      ) ) },
  %== fumar, EL fumar
  %== fumar, fumar PUROS
  %== SI/NO sabe fumar
  %== no LE gusta fumar
  %== pepe COME
  %== pepe juega BIEN ; HOY ; MUCHO/POCO al futbol
  %== pepe fue A JUGAR Y A CORRER con un amigo
  %== lo hice YO.
  %== él toca LAS VIEJAS ARMONICAS MARRONES DE TU ABUELA
  %== (*) lo hice MI, TI, lo hice YO --> YO lo hice
  %== pepe canta un bolero A ANTONIO
  %== pepe canta un bolero a Antonio EN ALICANTE
  % llueve en Alicante
  % hay manzanas, las hay
%----- COMPLEMENTO CIRCUNSTANCIAL -----
% Sólo devuelve NUM/GEN/PERS cuando se completa con un adj(cal)
cc(____) ++> sp(____).
cc(____) ++> gerundioSN.
cc(____) ++> adv(sit).
fusion(
  (cc(Numero,Genero,IP) ++>
    ADV1:adv(TipoAdv),
    { (TipoAdv == cant ; TipoAdv == mod ; TipoAdv == comp) },
    ( ADV2:adv(TipoAdv2), {TipoAdv2!=comp, TipoAdv2!=af, !+(ADV1=ADV2)} ),
    << gerundioSN >> ),
  %== Pepe canta EN ALICANTE, DURANTE EL DIA, HASTA LA NOCHE,
  %== CON CUIDADO, POR CELOS, POR TODA LA CALLE
  %== Pepe anda COJEANDO, OCULTANDOSE
  %== Pepe cantó MUY LEJOS, MUY BIEN, MAS CERCA
  %== Él está muy bien CANTANDO. cantó muy bien LLOVIENDO
  (cc(Numero,Genero,IP) ++>
    ADV1:adv(TipoAdv),
    { (TipoAdv == cant ; TipoAdv == mod ; TipoAdv == comp) },
    ( adj(Numero,Genero,cal), _ ),
    << gerundioSN >> ),
  %== Pepe canta MUY CONTENTO
  %== Él está muy bien CANTANDO. cantó muy bien LLOVIENDO

```

```

(cc(Numero,Genero,IP) ++>
  ADV1:adv(TipoAdv),
  { (TipoAdv == cant ; TipoAdv == mod ; TipoAdv == comp) },
  ( {+(ADV1=adv(____,muy)), TipoAdv != comp}, _),          %== *...muy muy *...muy menos
  << gerundioSN >> )          %== Él está muy bien CANTANDO. cantó muy bien LLOVIENDO
).
%----- APOSICION -----
%== el gato gris, EL HIJO DEL GATO DE TU HERMANA, se fue. Viene el tren de Alicante, EL
EXPRESO.
fusion(
  (aposition(Numero,Genero) ++>
    [.,],          % Viene precedido por una coma
    snSimple(Numero,Genero,terc,sustAdj,suj),
    [.,]),          % Termina en coma
  (aposition(Numero,Genero) ++>
    [.,],          % Viene precedido por una coma
    snSimple(Numero,Genero,terc,sustAdj,suj),
    [.]),          % Termina en punto (final de frase)
  ).
%----- ADYACENTE TIPO ADJETIVO -----
%== LOS FRECUENTEMENTE GRISES DIAS, MUY GRISES, MAS GRISES, MUY PINTADO, LOS
GRISES DIAS, MAS DE 5 GRISES, PINTADOS, ...
fusion(
  (adyAdj(Numero,Genero) ++>
    << adv(TipoAdv), (TipoAdv==cant ; TipoAdv==mod ; TipoAdv==comp) >>,
    ( adj(Numero,Genero,cal), _), {true}
  ),
  (adyAdj(Numero,Genero) ++>
    << adv(TipoAdv), (TipoAdv==cant ; TipoAdv==mod ; TipoAdv==comp) >>,
    (verbo(Numero,_,participio,TipoVerbo), { TipoVerbo != copul } ), {true}
  )
  ).
%----- DETERMINANTE 1º -----
det(sing,masc) ++> [todo], art(sing,masc,_).
%== ASOCIACION DE DETERMINANTES: tambien ALGUNOS DE, LOS MISMOS, TODO EL
%== PUBLICO,TODO LO, TODO UN, TODOS LOS, TODA LA, ...
det(pl,masc) ++> [todos], art(pl,masc,_).
det(Numero,Genero) ++>          %== MI, ESTE, BASTANTES, ALGUNO, SOLO
  adjSimple(Numero,Genero,TipoAdj),          % Forma átona del adj.pos.
  { (TipoAdj == dem ; TipoAdj == pos ; TipoAdj == ind) }.
det(Numero,Genero) ++> art(Numero,Genero,_).          %== LOS, EL, UN, UNOS
%----- DETERMINANTE 2º -----
det2(Numero,Genero,TipoAdj) ++>
  adjSimple(Numero,Genero,TipoAdj),
  { (TipoAdj == num ; TipoAdj == ord) }.          %== DOS,los DOS amigos,estos DOS,
%== mis DOS,unos DOS,sólo DOS
det2(Numero,Genero,comp) ++>          %== MAS DE 5 AMIGOS, LOS MENOS DE 7
  adv(comp), [de], adjSimple(Numero,Genero,num).
%----- ORACION DE RELATIVO -----
oracRelat ++>          %== QUE NO CANTO UNA CANCION A TU HIJO, QUE YO NO LE GUSTO
  pron(____,rel,_),
  << sn(Numero,Genero,Persona,_,suj) >>,
  sv(Numero,Genero,Persona,_).

```

```

/*----- SINTAGMA PREPOSICIONAL -----*/
Tipo1 puede ser "sn" o "noSn"
Tipo2 puede ser "cc" o "oi"
Esto es así para resolver la situación de que un CC tipo SP puede tener como preposición las típicas
de un SN (de/del)
%== AL/DEL NIÑO, DE TODOS LOS NIÑOS QUE JUEGAN, POR MI/TI, POR JUGAR
*/
coordinated(
  sp(Tipo1,Tipo2),
  spSimple(Tipo1,Tipo2),
  -
  )
% No hay ninguna restricción para la coordinación
spSimple(Tipo1,Tipo2) ++>
  prep(Tipo1,Tipo2),
  sn(____,sp).
%----- COORDINACION DE PREPOSICIONES -----
coordinated(
  prep(Tipo1,Tipo2),
  prepSimple(Tipo1,Tipo2),
  (Tipo2 == cc)
  )
%== María trabaja POR Y PARA el dinero
%----- GERUNDIO COMO MODIFICADOR DEL SN -----
%== el perro gris de tu hermana COMIENDO LAS COSTILLAS DE UN CORDERO es muy peligroso
%== un perro SIENDO BUENO es maravilloso
fusion(
  (gerundioSN ++>
    verbo(I1,I2,gerundio,I3),
    << sp(____) >>,
    {TipoSn \== verboInf, TipoSn \== pron, FuncSn \== sp} ),
  (gerundioSN ++>
    verbo(I1,I2,gerundio,I3),
    << sn(____,TipoSn,FuncSn)>>,
    {TipoSn \== verboInf, TipoSn \== pron, FuncSn \== sp} )
  )
%----- ADJETIVOS -----
coordinated(
  adj(Numero,Genero,Tipo),
  adjYuxt(Numero,Genero,Tipo),
  (Tipo == cal)
  )
% Sólo se coordinarán los calificativos
juxtaposition(
  adjYuxt(Numero,Genero,Tipo),
  adjSimple(Numero,Genero,Tipo),
  (Tipo == cal)
  )
% Sólo se coordinarán los calificativos
%----- SÍMBOLOS TERMINALES -----
basicWord(adv(Tipo)).
basicWord(adjSimple(Numero,Genero,Tipo)).
basicWord(art(Numero,Genero,Tipo)).
basicWord(conj).
basicWord(prepareSimple(Tipo1, Tipo2)).
basicWord(pron(Numero,Genero,Persona,Tipo1,Funcion)).
basicWord(sust(Numero,Genero,Tipo)).

```

basicWord(verbo(Numero, Persona, Tiempo, Tipo)).

Universitat d'Alacant
Universidad de Alicante

Apéndice B: corpus de frases declarativas

Corpus de Oraciones Correctas

Oración Correcta Nº 1 → *orac.imperativa sing*

[tocad,un,bolero,a,Antonio,.]

*** decl(existeUn(_34C0,bolero(_34C0),tocad(noHaySuj,_34C0,Antonio)))

Oración Correcta Nº 2 → *orac.imperativa pl*

[toca,dos,boleros,a,Antonio,.]

*** decl(existe(dos,_24EC,boleros(_24EC),toca(noHaySuj,_24EC,Antonio)))

Oración Correcta Nº 3 → *orac.imperativa*

[no,sabe,fumar,puros,.]

*** decl(no(sabe(noHaySuj,fumar(_1F20) & puros(_1F20),noHayOI)))

Oración Correcta Nº 4 → *orac.imperativa*

[le,gusta,fumar,puros,.]

*** decl(gusta(noHaySuj,fumar(_1604) & puros(_1604),noHayOI))

Oración Correcta Nº 10 → *orac.declarativa minima*

[fuma,.]

*** decl(fuma(noHaySuj,noHayOD,noHayOI))

Oración Correcta Nº 11 → *orac.declarativa minima*

[Pepe,fuma,puros,.]

*** decl(fuma(Pepe,puros(_4204),noHayOI))

Oración Correcta Nº 20 → *coord.orac*

[canta,y,fuma,.]

*** decl(canta(noHaySuj,noHayOD,noHayOI) & fuma(noHaySuj,noHayOD,noHayOI))

Oración Correcta Nº 21 → *coord.orac*

[Marga,canta,,yo,juego,y,tú,juegas,.]

*** decl(canta(Marga,noHayOD,noHayOI) & juego(yo(_07B8),noHayOD,noHayOI) & juegas(tú(_097C),noHayOD,noHayOI))

Oración Correcta Nº 22 → *coord.orac*

[el,jugar,al,futbol,me,gusta,y,Marga,no,le,gusta,.]

*** decl(def(_0788,futbol(_0788),def(_0788,jugar(_0788),gusta(_0788,noHayOD,noHayOI))) & no(gusta(Marga,noHayOD,noHayOI)))

Oración Correcta Nº 30 → *suj pron*

[tú,juegas,al,futbol,.]

<p>*** decl(def(_19A0,futbol(_19A0)),juegas(tú(_107C),noHayOD,_19A0)))</p> <p>Oración Correcta Nº 31 → <i>sub pron</i></p> <p>[yo,que,mi,hija,juega,al,futbol,jugando,al,futbol,doy,risa,.]</p> <p>*** decl(doy(yo(_06CC),risa(_0E34),noHayOI))</p> <p>Oración Correcta Nº 32 → <i>sub sust</i></p> <p>[Antonio,juega,.]</p> <p>*** decl(juega(Antonio,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 33 → <i>sub sust</i></p> <p>[todos,los,más,de,cuatro,muy,grandes,y,viejos,perros,mios,pintados,de,color,gris,que,vienen,a,jugar,corren,mucho,.]</p> <p>*** decl(existe(más,cuatro,_064C,((muy(grandes(_064C) & viejos(_064C)) & perros(_064C)) & pintados(_064C)) & de(color(_09F4) & gris(_09F4),_064C),mucho(corren(_064C,noHayOD,noHayOI))))))</p> <p>Oración Correcta Nº 34 → <i>sub verbo Infinitivo</i></p> <p>[ser,bueno,no,gusta,.]</p> <p>*** decl(no(gusta(ser(_05B4) & bueno(_05B4),noHayOD,noHayOI)))</p> <p>Oración Correcta Nº 35 → <i>sub verbo Infinitivo</i></p> <p>[el,fumar,los,puros,de,tu,abuelo,no,me,gusta,.]</p> <p>*** decl(def(_08DC,abuelo(_08DC),def(_05EC,puros(_05EC) & de(_08DC,_05EC),def(_05EC,fumar(_05EC),no(gusta(_05EC,noHayOD,noHayOI))))))</p> <p>Oración Correcta Nº 36 → <i>sub adj</i></p> <p>[el,más,gris,me,gusta,.]</p> <p>*** decl(def(_05C0,más(gris(_05C0)),gusta(_05C0),noHayOD,noHayOI)))</p> <p>Oración Correcta Nº 37 → <i>sub adj</i></p> <p>[todos,los,muy,grises,gatos,mios,,los,niños,que,vienen,a,jugar,,me,gustan,.]</p> <p>*** decl(def(_0780,niños(_0780),todo(_0780,muy(grises(_0780)) & gatos(_0780),gustan(_0780,noHayOD,noHayOI))))</p> <p>Oración Correcta Nº 38 → <i>SN adj</i></p> <p>[Antonio,de,azul,me,gusta,.]</p> <p>*** decl(gusta(Antonio & de(azul(_075C),Antonio),noHayOD,noHayOI))</p> <p>Oración Correcta Nº 40 → <i>SN sust sing</i></p> <p>[el,gato,juega,.]</p> <p>*** decl(def(_6DC8,gato(_6DC8)),juega(_6DC8,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 41 → <i>SN sust sing</i></p> <p>[mi,gato,juega,.]</p> <p>*** decl(def(_1DEC,gato(_1DEC)),juega(_1DEC,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 42 → <i>SN sust sing</i></p> <p>[este,gato,de,tu,hermana,juega,.]</p> <p>*** decl(def(_1C68,hermana(_1C68),def(_10AC,gato(_10AC) & de(_1C68,_10AC)),juega(_10AC,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 43 → <i>SN sust sing</i></p> <p>[un,gato,de,tu,hermana,juega,.]</p> <p>*** decl(def(_1C00,hermana(_1C00),existeUn(_108C,gato(_108C) & de(_1C00,_108C)),juega(_108C,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 44 → <i>SN sust sing</i></p> <p>[todo,el,público,de,Alicante,fuma,.]</p> <p>*** decl(todo(_10AC,público(_10AC) & de(Alicante,_10AC)),fuma(_10AC,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 45 → <i>SN sust sing</i></p> <p>[todo,un,niño,de,Alicante,hizo,aquello,.]</p> <p>*** decl(todo(_1164,niño(_1164) & de(Alicante,_1164)),hizo(_1164,aquello(_5E54),noHayOI))</p> <p>Oración Correcta Nº 46 → <i>SN sust sing</i></p> <p>[todo,perro,comiendo,carne,da,miedo,.]</p> <p>*** decl(todo(_0654,perro(_0654) & comiendo(carne(_07EC),_0654),da(_0654,miedo(_0968),noHayOI))</p> <p>Oración Correcta Nº 50 → <i>SN sust pl</i></p>
--

[los,gatos,juegan,.]
 *** decl(def_41CC,gatos_41CC),juegan(_41CC,noHayOD,noHayOI)))
 Oración Correcta Nº 51 → SN sust pl

[mis,gatos,juegan,.]
 *** decl(def_0FB8,gatos_0FB8),juegan(_0FB8,noHayOD,noHayOI)))
 Oración Correcta Nº 52 → SN sust pl

[estos,gatos,de,tu,hermana,juegan,.]
 *** decl(def_1C68,hermana_1C68),def_10AC,gatos_10AC) &
 de(_1C68,_10AC),juegan(_10AC,noHayOD,noHayOI)))
 Oración Correcta Nº 53 → SN sust pl

[unos,gatos,de,tu,hermana,juegan,.]
 *** decl(def_1C20,hermana_1C20),existeVarios_10AC,gatos_10AC) &
 de(_1C20,_10AC),juegan(_10AC,noHayOD,noHayOI)))
 Oración Correcta Nº 54 → SN sust pl

[todos,los,gatos,de,Alicante,fuman,.]
 *** decl(todo_10CC,gatos_10CC) & de(Alicante,_10CC),fuman(_10CC,noHayOD,noHayOI)))
 Oración Correcta Nº 55 → SN sust pl

[todos,unos,niños,de,Alicante,hicieron,aquello,.]
 *** decl(todo_1184,niños_1184) & de(Alicante,_1184),hicieron(_1184,aquello_53D8),noHayOI)))
 Oración Correcta Nº 60 → SN sust pl

[dos,gatos,juegan,.]
 *** decl(existe(dos,_10F0,gatos_10F0),juegan(_10F0,noHayOD,noHayOI)))
 Oración Correcta Nº 61 → SN sust pl

[más,de,dos,gatos,juegan,.]
 *** decl(existe(más,dos,_0718,gatos_0718),juegan(_0718,noHayOD,noHayOI)))
 Oración Correcta Nº 62 → SN sust pl

[el,primer,gato,juega,.]
 *** decl(existe(primer,_653C,gato_653C),juega(_653C,noHayOD,noHayOI)))
 Oración Correcta Nº 70 → SN sust sing

[un,antiguo,amigo,americano,fuma,.]
 *** decl(existeUn_242C,(antiguo_242C) & amigo_242C)) &
 americano_242C),fuma(_242C,noHayOD,noHayOI)))
 Oración Correcta Nº 71 → SN sust pl

[dos,antiguos,amigos,americanos,fuman,.]
 *** decl(existe(dos,_1068,(antiguos_1068) & amigos_1068)) &
 americanos_1068),fuman(_1068,noHayOD,noHayOI)))
 Oración Correcta Nº 72 → SN sust pl

[los,gatos,pintados,de,color,gris,me,gustan,.]
 *** decl(def_1148,(gatos_1148) & pintados_1148)) & de(color_1EA8) &
 gris_1EA8),_1148),gustan(_1148,noHayOD,noHayOI)))
 Oración Correcta Nº 73 → SN sust pl

[los,días,muy,grises,me,gustan,.]
 *** decl(def_10B0,días_10B0) & muy(grises_10B0),gustan(_10B0,noHayOD,noHayOI)))
 Oración Correcta Nº 74 → SN sust pl

[los,frecuentemente,grises,días,de,Alicante,me,gustan,.]
 *** decl(def_113C,(frecuentemente(grises_113C)) & días_113C)) &
 de(Alicante,_113C),gustan(_113C,noHayOD,noHayOI)))
 Oración Correcta Nº 75 → SN sust pl

[los,muy,grises,días,de,Alicante,me,gustan,.]
 *** decl(def_118C,(muy(grises_118C)) & días_118C)) &
 de(Alicante,_118C),gustan(_118C,noHayOD,noHayOI)))
 Oración Correcta Nº 76 → SN sust sing

[el,gato,más,azul,juega,bien,.]
 *** decl(def_10D0,gato_10D0) & más(azul_10D0),bien(juega_10D0,noHayOD,noHayOI)))

<p>Oración Correcta Nº 77 -> SN sust sing [el,gato,gris,azulado,juega,bien,.] *** decl(def(_10D8,gato(_10D8) & gris(_10D8) & azulado(_10D8),bien(juega(_10D8,noHayOD,noHayOI))))</p> <p>Oración Correcta Nº 80 -> SN sust sing [un,antiguo,amigo,mio,americano,fuma,.] *** decl(existeUn(_07E4,(antiguo(_07E4) & amigo(_07E4)) & americano(_07E4),fuma(_07E4,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 81 -> SN sust pl [unas,amigas,mias,fuman,.] *** decl(existeVarios(_7644,amigas(_7644),fuman(_7644,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 90 -> SN sust sing [Pepe,,el,niño,americano,,fuma,.] *** decl(def(Pepe,niño(Pepe) & americano(Pepe),fuma(Pepe & Pepe,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 91 -> SN sust pl [los,Pepes,,los,niños,americanos,,fuman,.] *** decl(def(Pepes,niños(Pepes) & americanos(Pepes),fuman(Pepes & Pepes,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 92 -> SN sust sing [el,tren,de,Alicante,,el,expreso,,viene,deprisa,.] *** decl(def(Alicante,expreso(Alicante),def(_05E4,tren(_05E4) & de(Alicante & Alicante,_05E4),deprisa(viene(_05E4,noHayOD,noHayOI))))</p> <p>Oración Correcta Nº 93 -> SN sust sing [viene,el,tren,de,Alicante,,el,expreso,.] *** decl(def(Alicante,expreso(Alicante),def(_064C,tren(_064C) & de(Alicante & Alicante,_064C),viene(noHaySuj,_064C,noHayOI))</p> <p>Oración Correcta Nº 94 -> SN sust sing [el,tren,de,Alicante,,el,gris,,viene,deprisa,.] *** decl(def(Alicante,gris(Alicante),def(_1184,tren(_1184) & de(Alicante & Alicante,_1184),deprisa(viene(_1184,noHayOD,noHayOI))))</p> <p>Oración Correcta Nº 100 -> coord.SN [Antonio,,los,Pepes,,ella,,el,jugar,al,futbol,,el,fumar,,el,azul,y,yo,cansamos,.] *** decl(def(_08D4,futbol(_08D4),def(_0B94,fumar(_0B94),def(_0C58,azul(_0C58),def(_08D4,jugar(_08D4) & _08D4 & _0B94 & _0C58,cansamos(Antonio & Pepes & ella(_0820) & _08D4 & yo(_0DA8),noHayOD,noHayOI))))</p> <p>Oración Correcta Nº 101 -> coord.SN [juegan,los,amigos,y,amigas,.] *** decl(def(_1320,amigos(_1320),juegan(noHaySuj,_1320 & amigas(_1AB8),noHayOI))</p> <p>Oración Correcta Nº 102 -> coord.SN [tú,y,yo,jugamos,.] *** decl(jugamos(tú(_0A94) & yo(_0B38),noHayOD,noHayOI))</p> <p>Oración Correcta Nº 103 -> coord.SN [él,y,yo,jugamos,.] *** decl(jugamos(él(_71F0) & yo(_8EA8),noHayOD,noHayOI))</p> <p>Oración Correcta Nº 104 -> coord.SN [yo,y,tú,jugamos,.] *** decl(jugamos(yo(_2484) & tú(_413C),noHayOD,noHayOI))</p> <p>Oración Correcta Nº 105 -> coord.SN [yo,y,él,jugamos,.] *** decl(jugamos(yo(_1070) & él(_1290),noHayOD,noHayOI))</p> <p>Oración Correcta Nº 106 -> coord.SN [él,y,ellas,juegan,.] *** decl(juegan(él(_063C) & ellas(_06E0),noHayOD,noHayOI))</p> <p>Oración Correcta Nº 107 -> coord.SN [nosotros,y,vosotros,jugamos,.]</p>
--

*** decl(jugamos(nosotros(_5180) & vosotros(_6E38),noHayOD,noHayOI))
 Oración Correcta Nº 108 -> coord.SN
 [vosotros,y,tú,jugais,.]
 *** decl(jugais(vosotros(_15D4) & tú(_328C),noHayOD,noHayOI))
 Oración Correcta Nº 109 -> coord.SN
 [vosotros,y,ella,jugais,.]
 *** decl(jugais(vosotros(_1090) & ella(_12B0),noHayOD,noHayOI))
 Oración Correcta Nº 110 -> SP sing
 [el,gato,del,niño,juega,.]
 *** decl(def(_0814,niño(_0814),def(_063C,gato(_063C) & de(_0814,_063C),juega(_063C,noHayOD,noHayOI))))
 Oración Correcta Nº 111 -> SP pl
 [el,gato,de,las,niñas,juega,.]
 *** decl(def(_0824,niñas(_0824),def(_064C,gato(_064C) & de(_0824,_064C),juega(_064C,noHayOD,noHayOI))))
 Oración Correcta Nº 112 -> SP sing
 [Pepe,de,tí,jugaba,.]
 *** decl(jugaba(Pepe & de(tí(_07C8),Pepe),noHayOD,noHayOI))
 Oración Correcta Nº 113 -> SP pl
 [Pepe,de,vosotros,que,vuestro,hiho,juega,al,futbol,jugaba,.]
 *** decl(jugaba(Pepe & de(vosotros(_09CC),Pepe),noHayOD,noHayOI))
 Oración Correcta Nº 114 -> SP pl
 [Pepe,de,vosotros,jugaba,.]
 *** decl(jugaba(Pepe & de(vosotros(_07C4),Pepe),noHayOD,noHayOI))
 Oración Correcta Nº 115 -> coord.de SP
 [el,gato,de,Antonio,y,de,los,músicos,no,juega,.]
 *** decl(def(_09F4,músicos(_09F4),def(_068C,gato(_068C) & de(Antonio,_068C) & de(_09F4,_068C),no(juega(_068C,noHayOD,noHayOI))))
 Oración Correcta Nº 116 -> coord.de SP
 [el,gato,de,él,,de,ellos,y,de,aquellos,no,juega,.]
 *** decl(def(_12CC,gato(_12CC) & de(él(_1E40),_12CC) & de(ellos(_2290),_12CC) & de(aquellos(_3D30),_12CC),no(juega(_12CC,noHayOD,noHayOI))))
 Oración Correcta Nº 117 -> coord.de Prep
 [María,trabaja,por,y,para,ganar,dinero,.]
 *** decl(por(ganar(_519C) & dinero(_519C),trabaja(María,noHayOD,noHayOI)) & para(ganar(_519C) & dinero(_519C),trabaja(María,noHayOD,noHayOI)))
 Oración Correcta Nº 120 -> Sucesion adj CAL sing
 [el,gato,gris,azulado,oscuro,gris,oscuro,juega,.]
 *** decl(def(_2AAC,gato(_2AAC) & gris(_2AAC) & azulado(_2AAC) & oscuro(_2AAC) & gris(_2AAC) & oscuro(_2AAC),juega(_2AAC,noHayOD,noHayOI))
 Oración Correcta Nº 121 -> Sucesion adj CAL pl
 [los,gatos,grises,azulados,oscuros,juegan,.]
 *** decl(def(_1100,gatos(_1100) & grises(_1100) & azulados(_1100) & oscuros(_1100),juegan(_1100,noHayOD,noHayOI))
 Oración Correcta Nº 122 -> Coordinacion adj CAL sing
 [el,gato,gris,azulado,y,oscuro,juega,.]
 *** decl(def(_1118,gato(_1118) & (gris(_1118) & azulado(_1118)) & oscuro(_1118),juega(_1118,noHayOD,noHayOI))
 Oración Correcta Nº 123 -> Coordinacion adj CAL pl
 [todos,los,gatos,grises,juegan,.]
 *** decl(todo(_07D4,gatos(_07D4) & grises(_07D4),juegan(_07D4,noHayOD,noHayOI))
 Oración Correcta Nº 130 -> Gerundio como parte un SN
 [yo,comiendo,la,carne,de,tu,abuela,doy,risa,.]
 *** decl(def(_22A8,abuela(_22A8),def(_1734,carne(_1734) & de(_22A8,_1734),doy(yo(_11E0) & comiendo(_1734,_11E0),risa(_2F74),noHayOI))

<p>Oración Correcta Nº 131 → Gerundio como parte un SN [yo, jugando, al, futbol, doy, risa, .] *** decl(def(_26C4, futbol(_26C4), doy(yo(_12C4) & jugando(_26C4, _12C4), risa(_5528), noHayOI)))</p> <p>Oración Correcta Nº 132 → Gerundio como parte un SN [yo, comiendo, de, pie, doy, risa, .] *** decl(doy(yo(_2424) & comiendo(de(pie(_3824), _2424), _2424), risa(_64E8), noHayOI))</p> <p>Oración Correcta Nº 133 → Gerundio como parte un SN [yo, comiendo, doy, risa, .] *** decl(doy(yo(_2490) & comiendo(_84C8, _2490), risa(_436C), noHayOI))</p> <p>Oración Correcta Nº 134 → Gerundio como parte un SN [yo, siendo, bueno, doy, risa, .] *** decl(doy(yo(_10A4) & siendo(bueno(_15F8), _10A4), risa(_4124), noHayOI))</p> <p>Oración Correcta Nº 140 → SV min [canta, .] *** decl(canta(noHaySuj, noHayOD, noHayOI))</p> <p>Oración Correcta Nº 141 → SV max [ella, no, me, canta, muy, bien, un, bolero, de, Machín, en, Alicante, .] *** decl(existeUn(_22C8, bolero(_22C8) & de(Machín, _22C8), no(muy(bien(en(Alicante, canta(ella(_1250), _22C8, noHayOI))))))</p> <p>Oración Correcta Nº 150 → SV OI:(sp) [ella, canta, a, Pepe, .] *** decl(canta(ella(_1A7C), noHayOD, Pepe))</p> <p>Oración Correcta Nº 151 → SV OD:SUST [ella, canta, dos, boleros, .] *** decl(existe(dos, _17C4, boleros(_17C4), canta(ella(_1098), _17C4, noHayOI)))</p> <p>Oración Correcta Nº 152 → SV OD:VERBOINF [ella, sabe, fumar, los, puros, de, tu, abuelo, .] *** decl(def(_2CC0, abuelo(_2CC0), def(_18C0, puros(_18C0) & de(_2CC0, _18C0), sabe(ella(_11B8), fumar(_18C0), noHayOI)))</p> <p>Oración Correcta Nº 153 → SV OD:PRON(suj) [lo, hice, yo, .] *** decl(hice(noHaySuj, yo(_1EF0), noHayOI))</p> <p>Oración Correcta Nº 154 → SV OD:PRON(suj) [yo, hice, aquello, .] *** decl(hice(yo(_0620), aquello(_0784), noHayOI))</p> <p>Oración Correcta Nº 155 → SV OD y OI [ella, canta, dos, boleros, a, Pepe, .] *** decl(existe(dos, _07B8, boleros(_07B8), canta(ella(_0650), _07B8, Pepe)))</p> <p>Oración Correcta Nº 156 → SV OI y OD [ella, canta, a, Pepe, dos, boleros, .] *** decl(existe(dos, _094C, boleros(_094C), canta(ella(_0650), _094C, Pepe)))</p> <p>Oración Correcta Nº 157 → SV OD, OI y CC [ella, canta, dos, boleros, a, Pepe, en, Alicante, .] *** decl(existe(dos, _093C, boleros(_093C), en(Alicante, canta(ella(_07C8), _093C, Pepe)))</p> <p>Oración Correcta Nº 158 → SV OI, OD y CC [ella, canta, a, Pepe, dos, boleros, en, Alicante, .] *** decl(existe(dos, _2214, boleros(_2214), en(Alicante, canta(ella(_11B0), _2214, Pepe)))</p> <p>Oración Correcta Nº 159 → SV CC, OD, OI y CC [ella, canta, muy, bien, dos, boleros, a, Pepe, en, Alicante, .] *** decl(existe(dos, _0874, boleros(_0874), muy(bien(en(Alicante, canta(ella(_0690), _0874, Pepe))))))</p> <p>Oración Correcta Nº 160 → SV CC, OI, OD y CC [ella, canta, hoy, a, Pepe, dos, boleros, en, Alicante, .] *** decl(existe(dos, _24EC, boleros(_24EC), hoy(en(Alicante, canta(ella(_11F0), _24EC, Pepe))))))</p>
--

Oración Correcta Nº 161 → SV CC,OI,OD y CC
 [ella,canta,por,mí,a,Pepe,dos,boleros,en,Alicante,.]
 *** decl(existe(dos,_0B44,boleros(_0B44),por(mí(_08CC),en(Alicante,canta(ella(_0694),_0B44,Pepe))))))

Oración Correcta Nº 170 → SV CC:sp
 [ella,canta,a,Pepe,dos,boleros,por,la,calle,.]

 decl(existe(dos,_222C,boleros(_222C),def(_32D0,calle(_32D0),por(_32D0,canta(ella(_11C8),_222C,Pepe))))))

Oración Correcta Nº 171 → SV CC:VerboGerundio
 [ella,canta,comiendo,.]
 *** decl(comiendo(_640C,canta(ella(_14DC),noHayOD,noHayOI)))

Oración Correcta Nº 172 → SV CC:AdvTiempo
 [ella,canta,hoy,.]
 *** decl(hoy(canta(ella(_958C),noHayOD,noHayOI)))

Oración Correcta Nº 173 → SV CC:AdvLugar
 [ella,canta,allí,.]
 *** decl(allí(canta(ella(_2188),noHayOD,noHayOI)))

Oración Correcta Nº 174 → SV CC:AdvCantidad
 [ella,canta,bastante,.]
 *** decl(bastante(canta(ella(_05AC),noHayOD,noHayOI)))

Oración Correcta Nº 175 → SV CC:AdvModo
 [ella,canta,acaloradamente,.]
 *** decl(acaloradamente(canta(ella(_3C98),noHayOD,noHayOI)))

Oración Correcta Nº 176 → SV CC:AdvCantidad+Adv
 [ella,canta,bastante,deprisa,.]
 *** decl(bastante(deprisa(canta(ella(_0784),noHayOD,noHayOI))))

Oración Correcta Nº 177 → SV CC:AdvComp+Adv
 [ella,canta,más,lejos,.]
 *** decl(más(lejos(canta(ella(_57F0),noHayOD,noHayOI))))

Oración Correcta Nº 178 → SV CC:AdvModo+AdjCal
 [ella,canta,bien,fuerte,.]
 *** decl(bien(fuerte(canta(ella(_1070),noHayOD,noHayOI))))

Oración Correcta Nº 179 → SV CC:AdvCantidad+AdjCal
 [ella,canta,bastante,fuerte,.]
 *** decl(bastante(fuerte(canta(ella(_8D0C),noHayOD,noHayOI))))

Oración Correcta Nº 180 → SV CC:AdvComp+AdjCal
 [ella,canta,más,fuerte,.]
 *** decl(más(fuerte(canta(ella(_23B8),noHayOD,noHayOI))))

Oración Correcta Nº 181 → SV concord.adj
 [ella,salió,muy,contenta,.]
 *** decl(muy(contenta(salió(ella(_062C),noHayOD,noHayOI))))

Oración Correcta Nº 182 → SV concord.adj
 [ella,salió,contenta,.]
 *** decl(salió(ella(_4A24),contenta(_6688),noHayOI))

Oración Correcta Nº 200 → SV Cop-CC,SN,SP,CC
 [el,fumar,no,les,es,afortunadamente,muy,malo,a,ellas,hoy,.]
 *** decl(def(_3294,fumar(_3294),no(afortunadamente(muy(hoy(malo(_3294))))))))

Oración Correcta Nº 201 → SV Cop-SP,SN,CC
 [el,fumar,no,le,es,a,Marga,muy,malo,hoy,.]
 *** decl(def(_068C,fumar(_068C),no(hoy(muy(malo(_068C))))))

Oración Correcta Nº 202 → SV Cop-SN
 [Pepe,es,bueno,.]
 *** decl(bueno(Pepe))

Oración Correcta Nº 203 --> SV Cop-SP
[Pepe, se, parece, a, ella, .]
*** decl(ella(Pepe))
Oración Correcta Nº 204 --> SV Cop-CC(adv+adj)
[Pepe, es, muy, bueno, .]
*** decl(muy(bueno(Pepe)))
Oración Correcta Nº 205 --> SV Cop-CC(adv+adv)
[Pepe, está, muy, bien, .]
*** decl(muy(bien(Pepe)))
Oración Correcta Nº 206 --> SV Cop-CC(sp)
[Antonio, está, de, mal, humor, .]
*** decl(de(mal_0868) & humor_0868), Antonio))
Oración Correcta Nº 207 --> SV Cop-CC(adv)
[ella, está, allí, .]
*** decl(allí(ella_50DC))
Oración Correcta Nº 208 --> SV Cop-CC, CC
[Pepe, es, muy, bueno, para, ellas, .]
*** decl(muy(bueno(para(ellas_36D4), Pepe)))
Oración Correcta Nº 209 --> SV Cop-CC, SN
[Pepe, es, para, ellas, muy, bueno, .]
*** decl(para(ellas_1E18), muy(bueno(Pepe)))
Oración Correcta Nº 210 --> SV Cop-SN, CC
[Pepe, es, bueno, para, todos, .]
*** decl(para(todos_524C), bueno(Pepe))
Oración Correcta Nº 211 --> SV Cop-SN, CC
[Pepe, es, un, amigo, para, todos, .]
*** decl(existeUn(Pepe, amigo(Pepe), para(todos_62AC), Pepe))
Oración Correcta Nº 220 --> SV trans-OI
[ella, mira, a, Pepe, .]
*** decl(mira(ella_1BE4), noHayOD, Pepe))
Oración Correcta Nº 221 --> SV trans-OD
[ella, mira, un, gato, .]
*** decl(existeUn_1754, gato_1754), mira(ella_1028), _1754, noHayOI))
Oración Correcta Nº 222 --> SV trans-Pron
[ella, la, mira, .]
*** decl(mira(ella_05AC), noHayOD, noHayOI))
Oración Correcta Nº 223 --> SV trans-Pron
[ella, se, mira, .]
*** decl(mira(ella_3ED8), noHayOD, noHayOI))
Oración Correcta Nº 230 --> SV impers
[llueve, .]
*** decl(llueve(noHaySuj, noHayOD, noHayOI))
Oración Correcta Nº 231 --> SV impers
[no, llueve, mucho, en, Alicante, .]
*** decl(no(mucho(en(Alicante, llueve(noHaySuj, noHayOD, noHayOI))))
Oración Correcta Nº 232 --> SV impers
[llueve, muy, poco, en, Alicante, .]
*** decl(muy(poco(en(Alicante, llueve(noHaySuj, noHayOD, noHayOI))))
Oración Correcta Nº 233 --> SV impers
[hay, manzanas, en, Alicante, .]
*** decl(en(Alicante, hay(noHaySuj, manzanas_12C4), noHayOI))
Oración Correcta Nº 240 --> NO TIENE SENTIDO
[el, músico, come, y, rie, un, chiste, .]

*** decl(def_05CC,músico_05CC),existeUn_0860,chiste_0860),come_05CC,_0860,noHayOl))) & def_05CC,músico_05CC),existeUn_0860,chiste_0860),rie_05CC,_0860,noHayOl)))

Oración Correcta Nº 241 → NO Elip. Obj

[el,músico,come,una,manzana,y,rie,.]

*** decl(def_0724,músico_0724),existeUn_08E0,manzana_08E0),come_0724,_08E0,noHayOl))) & def_0724,músico_0724),rie_0724,noHayOD,noHayOl)))

Oración Correcta Nº 242 → NO Elip. Obj, SI Elip. Suj/CC

[el,músico,cuenta,un,chiste,a,Pepe,y,rie,en,Alicante,.]

decl(def_05EC,músico_05EC),existeUn_07AC,chiste_07AC),en(Alicante,cuenta_05EC,_07AC,Pepe))) & def_05EC,músico_05EC),en(Alicante,rie_05EC,noHayOD,noHayOl)))

Oración Correcta Nº 243 → Verb.trans:Elip. Suj/Obj

[el,músico,cuenta,y,rie,un,chiste,a,Pepe,.]

*** decl(def_05DC,músico_05DC),existeUn_0874,chiste_0874),cuenta_05DC,_0874,Pepe))) & def_05DC,músico_05DC),existeUn_0874,chiste_0874),rie_05DC,_0874,Pepe)))

Oración Correcta Nº 244 → Verb.trans:Elip. Obj

[el,músico,rie,y,Marga,cuenta,un,chiste,.]

*** decl(def_05D4,músico_05D4),existeUn_096C,chiste_096C),rie_05D4,_096C,noHayOl))) & existeUn_096C,chiste_096C),cuenta(Marga,_096C,noHayOl)))

Oración Correcta Nº 245 → Verb.trans:Multioracion:Elip. Suj/Obj

[el,músico,cuenta,,,no,rie,,,disfruta,y,canta,un,chiste,en,Alicante,.]

decl(def_0604,músico_0604),existeUn_0A6C,chiste_0A6C),en(Alicante,cuenta_0604,_0A6C,noHayOl))) & def_0604,músico_0604),existeUn_0A6C,chiste_0A6C),no(en(Alicante,rie_0604,_0A6C,noHayOl)))) & def_0604,músico_0604),existeUn_0A6C,chiste_0A6C),en(Alicante,disfruta_0604,_0A6C,noHayOl)))) & def_0604,músico_0604),existeUn_0A6C,chiste_0A6C),en(Alicante,canta_0604,_0A6C,noHayOl))))

Oración Correcta Nº 246 → Orac. Impers:Elip. Suj/Obj

[llueve,,,graniza,y,nieva,mucho,en,Madrid,.]

*** decl(mucho(en(Madrid,llueve(noHaySuj,noHayOD,noHayOl))) & mucho(en(Madrid,graniza(noHaySuj,noHayOD,noHayOl))) & mucho(en(Madrid,nieva(noHaySuj,noHayOD,noHayOl))))

Oración Correcta Nº 247 → Orac. Impers:NO Elip. Suj/Obj

[llueve,en,Alicante,y,Pepe,no,gusta,de,jugar,lloviendo,.]

*** decl(en(Alicante,llueve(noHaySuj,noHayOD,noHayOl)) & no(de(jugar_32E0),lloviendo_CCFc,gusta(Pepe,noHayOD,noHayOl))))

Oración Correcta Nº 248 → Orac. Impers:NO Elip. Suj/Obj

[llueve,y,Pepe,no,está,contento,.]

*** decl(llueve(noHaySuj,noHayOD,noHayOl)) & no(contento(Pepe)))

Oración Correcta Nº 250 → ELIP. V. EC-SN

[yo,canto,un,bolero,y,una,copla,.]

*** decl(existeUn_0744,bolero_0744),existeUn_0840,copla_0840),canto(yo_05D0,_0744 & _0840,noHayOl)))

Oración Correcta Nº 251 → ELIP. V. EC-SP

[yo,canto,a,Pepe,y,a,ella,.]

*** decl(canto(yo_05D0),noHayOD,Pepe & ella_0948)))

Oración Correcta Nº 252 → ELIP. V. EC-CC

[yo,canto,en,Alicante,y,en,Madrid,.]

*** decl(en(Alicante,canto(yo_0728),noHayOD,noHayOl)) & en(Madrid,canto(yo_0728),noHayOD,noHayOl)))

Oración Correcta Nº 253 → ELIP. V. EC-CC

[yo,canto,por,ti,y,por,mí,.]

*** decl(por(ti_07FC),canto(yo_05D8),noHayOD,noHayOl)) & por(mí_0960),canto(yo_05D8),noHayOD,noHayOl)))

Oración Correcta Nº 254 → ELIP. V. EC-SP, SN, CC

[yo,canto,a,Pepe,y,a,ella,un,bolero,y,una,copla,en,Alicante,y,en,Madrid,.]

 decl(existeUn(_0AAC,bolero(_0AAC),existeUn(_0BA8,copla(_0BA8),en(Alicante,canto(yo(_0620),_0AAC & _0BA8,Pepe & ella(_09B8))) & en(Madrid,canto(yo(_0620),_0AAC & _0BA8,Pepe & ella(_09B8))))))
 Oración Correcta Nº 255 -> ELIP.V.EC-SN,SP,CC
 [yo,canto,un,bolero,y,una,copla,a,Pepe,y,a,ella,en,Alicante,.]

 decl(existeUn(_078C,bolero(_078C),existeUn(_0888,copla(_0888),en(Alicante,canto(yo(_0608),_078C & _0888,Pepe & ella(_0BEC))))))
 Oración Correcta Nº 256 -> ELIP.V.EC-SN,CC/
 [yo,canto,un,bolero,en,Alicante,y,en,Madrid,.]

*** decl(existeUn(_0750,bolero(_0750),en(Alicante,canto(yo(_05E0),_0750,noHayOl)) & en(Madrid,canto(yo(_05E0),_0750,noHayOl))))
 Oración Correcta Nº 257 -> ELIP.V.EC-CC,SN/
 [yo,canto,en,Alicante,y,en,Madrid,un,bolero,y,una,copla,.]

 decl(existeUn(_2AD0,bolero(_2AD0),existeUn(_3268,copla(_3268),en(Alicante,canto(yo(_1220),_2AD0 & _3268,noHayOl)) & en(Madrid,canto(yo(_1220),_2AD0 & _3268,noHayOl))))
 Oración Correcta Nº 258 -> ELIP.V.EC-SN,CC/
 [yo,canto,un,bolero,y,una,copla,en,Alicante,y,en,Madrid,.]

 decl(existeUn(_0910,bolero(_0910),existeUn(_0A0C,copla(_0A0C),en(Alicante,canto(yo(_0790),_0910 & _0A0C,noHayOl)) & en(Madrid,canto(yo(_0790),_0910 & _0A0C,noHayOl))))
 Oración Correcta Nº 265 -> ELIP.V.EC-CC,SN,CC
 [yo,canto,en,Alicante,un,bolero,y,una,copla,en,Madrid,.]

 decl(existeUn(_0934,bolero(_0934),existeUn(_0A30,copla(_0A30),en(Alicante,en(Madrid,canto(yo(_05F0),_0934 & _0A30,noHayOl))))))
 Oración Correcta Nº 270 -> ELIP.V.ENC-SN,CC
 [yo,canto,un,bolero,en,Alicante,y,una,copla,en,Madrid,.]

*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(existeUn(_0760,bolero(_0760),en(Alicante,canto(yo(_05F0),_0760,noHayOl))) & existeUn(_0AC0,copla(_0AC0),existeUn(_0760,bolero(_0760),en(Madrid,canto(_0AC0,_0760,noHayOl))))))
 Oración Correcta Nº 271 -> ELIP.V.ENC-CC,SN
 [yo,canto,en,Alicante,un,bolero,y,en,Madrid,una,copla,.]

*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(existeUn(_0918,bolero(_0918),en(Alicante,canto(yo(_05F0),_0918,noHayOl))) & existeUn(_0CF0,copla(_0CF0),en(Madrid,canto(yo(_05F0),_0CF0,noHayOl))))
 Oración Correcta Nº 272 -> ELIP.V.ENC-SP,CC
 [yo,canto,a,Pepe,en,Alicante,y,a,ella,en,Madrid,.]

*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(en(Alicante,canto(yo(_05F0),noHayOD,Pepe)) & en(Madrid,canto(yo(_05F0),noHayOD,ella(_0C00))))
 Oración Correcta Nº 273 -> ELIP.V.ENC-CC,SP
 [yo,canto,en,Alicante,a,Pepe,y,en,Madrid,a,ella,.]

*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(en(Alicante,canto(yo(_05F0),noHayOD,Pepe)) & en(Madrid,canto(yo(_05F0),noHayOD,ella(_0DB8))))
 Oración Correcta Nº 274 -> ELIP.V.ENC-SP,SN
 [yo,canto,a,Pepe,un,bolero,y,a,ella,una,copla,.]

*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(existeUn(_08EC,bolero(_08EC),canto(yo(_05F0),_08EC,Pepe)) & existeUn(_0C7C,copla(_0C7C),canto(yo(_05F0),_0C7C,ella(_0B98))))
 Oración Correcta Nº 275 -> ELIP.V.ENC-SN,SP
 [yo,canto,una,copla,a,Pepe,y,un,bolero,a,ella,.]

*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(existeUn(_0758,copla(_0758),canto(yo(_05F0),_0758,Pepe)) & existeUn(_0A94,bolero(_0A94),existeUn(_0758,copla(_0758),canto(_0A94,_0758,ella(_0CD4))))))

<p>Oración Correcta Nº 280 -> ELIP.V.ENC-SN [yo,canto,un,bolero,y,tú,una,copla,.] *** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada) *** decl(canto(yo(_05B8),noHayOD,noHayOI) & existeUn(_09C4,copla(_09C4),canto(noHaySuj,_09C4,noHayOI)))</p> <p>Oración Correcta Nº 281 -> ELIP.V.ENC-CC [yo,canto,en,Alicante,y,tú,en,Madrid,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(en(Alicante,canto(yo(_1118),noHayOD,noHayOI)) & en(Madrid,canto(tú(_2774),noHayOD,noHayOI)))</p> <p>Oración Correcta Nº 282 -> ELIP.V.ENC-SP [yo,canto,a,él,y,tú,a,ella,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(canto(yo(_05DC),noHayOD,él(_07C4)) & canto(tú(_0910),noHayOD,ella(_0AF8)))</p> <p>Oración Correcta Nº 283 -> ELIP.V.ENC-SN,SP [yo,canto,un,bolero,a,él,y,tú,una,copla,a,ella,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(existeUn(_0764,bolero(_0764),canto(yo(_05FC),_0764,él(_0938))) & existeUn(_0BEC,copla(_0BEC),canto(tú(_0A84),_0BEC,ella(_0DC0)))</p> <p>Oración Correcta Nº 284 -> ELIP.V.ENC-SN,SP [yo,canto,un,bolero,a,él,y,tú,a,ella,una,copla,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(existeUn(_0764,bolero(_0764),canto(yo(_05FC),_0764,él(_0938))) & existeUn(_0D64,copla(_0D64),canto(tú(_0A84),_0D64,ella(_0C80)))</p> <p>Oración Correcta Nº 285 -> ELIP.V.ENC-SN,CC [yo,canto,un,bolero,en,Alicante,y,tú,una,copla,en,Madrid,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(existeUn(_076C,bolero(_076C),en(Alicante,canto(yo(_05FC),_076C,noHayOI))) & existeUn(_0C3C,copla(_0C3C),en(Madrid,canto(tú(_0ACC),_0C3C,noHayOI)))</p> <p>Oración Correcta Nº 286 -> ELIP.V.ENC-CC,SN [yo,canto,en,Alicante,un,bolero,y,tú,una,copla,en,Madrid,.] *** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada) *** decl(en(Alicante,canto(yo(_0AE0),noHayOD,noHayOI)) & existeUn(_10B8,copla(_10B8),en(Madrid,canto(noHaySuj,_10B8,noHayOI)))</p> <p>Oración Correcta Nº 300 -> ELIP.V.ENC-ADV [yo,canto,muy,bien,un,bolero,a,él,en,Alicante,y,tú,muy,mal,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(existeUn(_0988,bolero(_0988),muy(bien(en(Alicante,canto(yo(_07A4),_0988,él(_0B5C)))))) & existeUn(_0988,bolero(_0988),muy(mal(en(Alicante,canto(tú(_0E60),_0988,él(_0B5C))))))</p> <p>Oración Correcta Nº 301 -> ELIP.V.ENC-CC [yo,canto,muy;bien,un,bolero,a,él,en,Alicante,y,tú,en,Madrid,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(existeUn(_07F0,bolero(_07F0),muy(bien(en(Alicante,canto(yo(_060C),_07F0,él(_09C4)))))) & existeUn(_07F0,bolero(_07F0),en(Madrid,muy(bien(canto(tú(_0CC8),_07F0,él(_09C4))))))</p> <p>Oración Correcta Nº 302 -> ELIP.V.ENC-ADV,CC [yo,canto,muy,bien,un,bolero,a,él,en,Alicante,y,tú,muy,mal,en,Madrid,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(existeUn(_0800,bolero(_0800),muy(bien(en(Alicante,canto(yo(_061C),_0800,él(_09D4)))))) & existeUn(_0800,bolero(_0800),muy(mal(en(Madrid,canto(tú(_0CD8),_0800,él(_09D4))))))</p> <p>Oración Correcta Nº 303 -> ELIP.V.ENC-SP [yo,canto,muy,bien,un,bolero,a,él,en,Alicante,y,tú,a,ella,.] *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada) *** decl(existeUn(_0988,bolero(_0988),muy(bien(en(Alicante,canto(yo(_07A4),_0988,él(_0B5C)))))) & existeUn(_0988,bolero(_0988),muy(bien(en(Alicante,canto(tú(_0E60),_0988,ella(_1048))))))</p> <p>Oración Correcta Nº 304 -> ELIP.V.ENC-SP</p>
--

<p>[yo,canto,muy,bien,un,bolero,a,él,en,Alicante,y,tú,una,copla,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(existeUn(_0D08,bolero(_0D08),muy(bien(en(Alicante,canto(yo(_0B24),_0D08,él(_0EDC)))))) & existeUn(_1344,copla(_1344),muy(bien(en(Alicante,canto(tú(_11E0),_1344,él(_0EDC))))))</p> <p>Oración Correcta Nº 310 → ELIP.V.ENC-SN+CC</p> <p>[yo,canto,un,bolero,y,tú,una,copla,en,Alicante,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)</p> <p>*** decl(en(Alicante,canto(yo(_05C8),noHayOD,noHayOI)) & existeUn(_09E0,copla(_09E0),en(Alicante,canto(noHaySuj,_09E0,noHayOI))))</p> <p>Oración Correcta Nº 311 → ELIP.V.ENC-SN+CC</p> <p>[yo,canto,en,Alicante,un,bolero,y,tú,una,copla,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)</p> <p>*** decl(en(Alicante,canto(yo(_05C8),noHayOD,noHayOI)) & existeUn(_0B94,copla(_0B94),en(Alicante,canto(noHaySuj,_0B94,noHayOI))))</p> <p>Oración Correcta Nº 312 → ELIP.V.ENC-SN+CC</p> <p>[yo,canto,muy,bien,un,bolero,en,Alicante,y,tú,en,Madrid,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(existeUn(_07DC,bolero(_07DC),muy(bien(en(Alicante,canto(yo(_05FC),_07DC,noHayOI)))) & existeUn(_07DC,bolero(_07DC),en(Madrid,muy(bien(canto(tú(_0B3C),_07DC,noHayOI))))))</p> <p>Oración Correcta Nº 313 → ELIP.V.ENC-SP+CC</p> <p>[yo,canto,en,Alicante,a,él,y,tú,a,ella,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(en(Alicante,canto(yo(_05EC),noHayOD,él(_0998)) & en(Alicante,canto(tú(_0AE4),noHayOD,ella(_0CCC))))</p> <p>Oración Correcta Nº 314 → ELIP.V.ENC-SP+CC</p> <p>[yo,canto,en,Alicante,a,él,y,tú,en,Madrid,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(en(Alicante,canto(yo(_05EC),noHayOD,él(_0998)) & en(Madrid,canto(tú(_0AE4),noHayOD,él(_0998))))</p> <p>Oración Correcta Nº 315 → ELIP.V.ENC-SN+CC</p> <p>[yo,canto,en,Alicante,un,bolero,y,tú,en,Madrid,.]</p> <p>*** decl(existeUn(_22F4,bolero(_22F4),en(Alicante,en(Madrid,canto(yo(_11A8),_22F4 & tú(_2A8C),noHayOI))))</p> <p>Oración Correcta Nº 320 → ELIP.V.ENC-ADV</p> <p>[él,canta,en,Alicante,a,él,y,tú,no,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(en(Alicante,canta(él(_05E0),noHayOD,él(_098C))) & no(en(Alicante,canta(tú(_0AD4),noHayOD,él(_098C))))</p> <p>Oración Correcta Nº 321 → ELIP.V.ENC-ADV</p> <p>[yo,canto,muy,bien,un,bolero,a,él,en,Alicante,y,tú,no,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(existeUn(_07E8,bolero(_07E8),muy(bien(en(Alicante,canto(yo(_0604),_07E8,él(_09BC)))) & existeUn(_07E8,bolero(_07E8),no(muy(bien(en(Alicante,canto(tú(_0CBC),_07E8,él(_09BC))))))</p> <p>Oración Correcta Nº 322 → ELIP.V.ENC-ADV</p> <p>[yo,no,canto,muy,bien,un,bolero,a,él,en,Alicante,y,tú,si,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(existeUn(_0808,bolero(_0808),no(muy(bien(en(Alicante,canto(yo(_060C),_0808,él(_09DC)))))) & existeUn(_0808,bolero(_0808),muy(bien(en(Alicante,canto(tú(_0CDC),_0808,él(_09DC))))))</p> <p>Oración Correcta Nº 330 → ELIP.V.ENC-SV Copul</p> <p>[yo,soy,muy,malo,jugando,al,futbol,en,Alicante,y,tú,en,Madrid,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(muy(malo(en(Alicante,yo(_07A4)))) & en(Madrid,muy(malo(tú(_0DE8))))</p> <p>Oración Correcta Nº 331 → ELIP.V.ENC-SV Copul</p> <p>[yo,soy,muy,malo,jugando,al,futbol,en,Alicante,y,tú,no,.]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p>

*** decl(muy(malo(en(Alicante,yo_0604)))) & no(muy(malo(en(Alicante,tú_0C44))))
 Oración Correcta Nº 332 --> ELIP.V.ENC-SV Copul
 [Pepe,se,parece,a,ella,y,tú,a,él,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(ella(Pepe) & él(tú_0968))
 Oración Correcta Nº 333 --> ELIP.V.ENC-SV Copul
 [yo,soy,muy,malo,jugando,al,futbol,en,Alicante,y,tú,muy,bueno,jugando,al,tenis,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(muy(malo(en(Alicante,yo_062C)))) & muy(bueno(en(Alicante,tú_0C70))))
 Oración Correcta Nº 334 --> ELIP.V.ENC-SV Copul
 [yo,soy,muy,malo,jugando,al,futbol,en,Alicante,y,tú,muy,bueno,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(muy(malo(en(Alicante,yo_060C)))) & muy(bueno(en(Alicante,tú_0C50))))
 Oración Correcta Nº 335 --> ELIP.V.ENC-SV Copul
 [yo,soy,muy,malo,jugando,al,futbol,en,Alicante,y,tú,jugando,al,tenis,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(muy(malo(en(Alicante,yo_061C)))) & def_0E28,tenis_0E28,muy(malo(en(Alicante,tú_0C60) & jugando_0E28_0C60))))
 Oración Correcta Nº 340 --> AUS-Conj.Or.decl
 [Antonio,juega,y,Pepe,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(juega(Antonio,noHayOD,noHayOI) & juega(Pepe,noHayOD,noHayOI))
 Oración Correcta Nº 341 --> ELIP.V.ENC-ADV
 [Pepe,canta,muy,bien,boleros,,,Antonio,no,y,Marga,canta,muy,mal,todos,los,mambos,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)
 *** decl(muy(bien(canta(Pepe,noHayOD,noHayOI))) & no(muy(bien(canta(noHaySuj,noHayOD,noHayOI))) & todo_0CB8,mambos_0CB8,muy(mal(canta(Marga_0CB8,noHayOI))))
 Oración Correcta Nº 342 --> ELIP.V.ENC-ADV
 [Pepe,canta,muy,bien,los,boleros,a,Marga,en,Alicante,,,Antonio,no,y,él,sí,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(def_0810,boleros_0810,muy(bien(en(Alicante,canta(Pepe_0810,Marga)))) & def_0810,boleros_0810,no(muy(bien(en(Alicante,canta(Antonio_0810,Marga)))) & def_0810,boleros_0810,muy(bien(en(Alicante,canta(él_0EE4,_0810,Marga))))
 Oración Correcta Nº 343 --> ELIP.V.ENC-ADV
 [Pepe,canta,muy,bien,los,boleros,,,Antonio,no,y,él,sí,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)
 *** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)
 *** decl(muy(bien(canta(Pepe,noHayOD,noHayOI))) & no(muy(bien(canta(noHaySuj,noHayOD,noHayOI))) & muy(bien(canta(él_0ADC),noHayOD,noHayOI)))
 Oración Correcta Nº 344 --> ELIP.V.ENC-ADV
 [yo,canto,muy,bien,los,boleros,,,ella,no,y,Antonio,canta,algunos,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)
 *** decl(muy(bien(canto(yo_05DC),noHayOD,noHayOI))) & no(muy(bien(canto(noHaySuj,noHayOD,noHayOI))) & muy(bien(canta(Antonio,algunos_0C44),noHayOI)))
 Oración Correcta Nº 360 --> FL Sólo verbo
 [fuma,.]
 *** decl(fuma(noHaySuj,noHayOD,noHayOI))
 Oración Correcta Nº 361 --> FL Sin Suj
 [fuma,puros,.]
 *** decl(fuma(noHaySuj,puros_2D58),noHayOI))
 Oración Correcta Nº 362 --> FL Sin Pred
 [Pepe,grande,fuma,.]

*** decl(fuma(Pepe & grande(Pepe),noHayOD,noHayOI))
Oración Correcta Nº 363 -> FL OD
[Pepe,canta,boleros,.]
*** decl(canta(Pepe,boleros(_0728),noHayOI))
Oración Correcta Nº 364 -> FL OI
[Pepe,canta,a,Marga,.]
*** decl(canta(Pepe,noHayOD,Marga))
Oración Correcta Nº 365 -> FL OD-OI
[Pepe,canta,boleros,a,Marga,.]
*** decl(canta(Pepe,boleros(_073C),Marga))
Oración Correcta Nº 366 -> FL OI-OD
[Pepe,canta,a,Marga,boleros,.]
*** decl(canta(Pepe,boleros(_08D0),Marga))
Oración Correcta Nº 367 -> FL Negación
[Pepe,no,canta,a,Marga,boleros,.]
*** decl(no(canta(Pepe,boleros(_0A48),Marga)))
Oración Correcta Nº 368 -> FL Afirmación
[Pepe,sí,canta,a,Marga,boleros,.]
*** decl(canta(Pepe,boleros(_2734),Marga))
Oración Correcta Nº 369 -> FL VerboCopul-OD-Sust
[Pepe,es,amigo,mio,.]
*** decl(amigo(Pepe))
Oración Correcta Nº 370 -> FL VerboCopul-OD-Sust
[Pepe,es,hoy,un,gran,amigo,mio,.]
*** decl(existeUn(Pepe,gran(Pepe) & amigo(Pepe),hoy(Pepe)))
Oración Correcta Nº 371 -> FL VerboCopul-OD-Adj
[tu,perro,es,gris,azulado,.]
*** decl(def(_1048,perro(_1048),gris(_1048) & azulado(_1048)))
Oración Correcta Nº 372 -> FL VerboCopul-CC
[un,amigo,mio,es,de,Alicante,.]
*** decl(existeUn(_1094,amigo(_1094),de(Alicante,_1094)))
Oración Correcta Nº 373 -> FL SN-Suj-Adj
[el,gris,azulado,me,gusta,.]
*** decl(def(_1050,gris(_1050) & azulado(_1050),gusta(_1050),noHayOD,noHayOI))
Oración Correcta Nº 374 -> FL SN-OD-Adj
[me,gusta,gris,azulado,.]
*** decl(gusta(noHaySuj,gris(_066C) & azulado(_066C),noHayOI))
Oración Correcta Nº 375 -> FL CC-Adv-Adv
[Pepe,canta,muy,fuerte,y,claro,las,coplas,hoy,.]
*** decl(def(_085C,coplas(_085C),muy(fuerte(hoy(canta(Pepe,_085C,noHayOI))) & claro(hoy(canta(Pepe,_085C,noHayOI))))))
Oración Correcta Nº 376 -> FL CC-AdvCant-Adv
[Pepe,se,come,las,manzanas,muy,deprisa,.]
*** decl(def(_0790,manzanas(_0790),muy(deprisa(come(Pepe,_0790,noHayOI))))))
Oración Correcta Nº 377 -> FL CC-Adv-SP
[un,amigo,mio,juega,mañana,a,futbol,en,Alicante,.]
*** decl(existeUn(_0734,amigo(_0734),mañana(en(Alicante,juega(_0734,noHayOD,futbol(_09F0))))))
Oración Correcta Nº 378 -> FL CC-SP-Adv
[un,amigo,mio,juega,en,Alicante,a,futbol,mañana,.]
*** decl(existeUn(_1160,amigo(_1160),en(Alicante,mañana(juega(_1160,noHayOD,futbol(_2F40))))))
Oración Correcta Nº 379 -> FL CC-SP-SP
[un,amigo,mio,juega,durante,el,dia,a,futbol,en,una,calle,.]

 decl(existeUn(_05F4,amigo(_05F4),def(_089C,dia(_089C),existeUn(_0C28,calle(_0C28),durante(_089C,en
 (_0C28,juega(_05F4,noHayOD,futbol(_0A70)))))))))
 Oración Correcta Nº 380 --> FL SN-DI
 [un,amigo,mio,de,un,pueblo,de,Alicante,juega,al,futbol,en,Madrid,.]
 *** decl(existeUn(_0800,pueblo(_0800) & de(Alicante,_0800),existeUn(_05FC,amigo(_05FC) &
 de(_0800,_05FC),def(_0BE4,futbol(_0BE4),en(Madrid,juega(_05FC,noHayOD,_0BE4)))))))))
 Oración Correcta Nº 381 --> FL SN-GSN
 [Pepe,contando,un,chiste,da,risa,.]
 *** decl(existeUn(_18E4,chiste(_18E4),da(Pepe & contando(_18E4,Pepe),risa(_2588),noHayOI)))
 Oración Correcta Nº 382 --> FL APOSIC
 [el,tren,azul,,el,expreso,,viene,deprisa,.]
 *** decl(def(_1140,expreso(_1140),def(_1140,tren(_1140) &
 azul(_1140),deprisa(viene(_1140,noHayOD,noHayOI)))))))))
 Oración Correcta Nº 383 --> FL APOSIC
 [viene,aquel,hombre,,Pepe,.]
 *** decl(def(Pepe,hombre(Pepe) & Pepe,viene(noHaySuj,Pepe,noHayOI)))
 Oración Correcta Nº 384 --> FL APOSIC
 [aquellos,niños,rubios,,los,hijos,de,tu,hermano,y,tu,hermana,,están,muy,delgados,.]
 *** decl(def(_098C,hermano(_098C),def(_0A88,hermana(_0A88),def(_061C,hijos(_061C) & de(_098C &
 _0A88,_061C),def(_061C,niños(_061C) & rubios(_061C),muy(delgados(_061C)))))))))
 Oración Correcta Nº 385 --> FL APOSIC
 [aquellos,niños,rubios,de,tu,pueblo,,los,hijos,de,tu,hermana,,juegan,muy,bien,al,futbol,.]
 *** decl(def(_087C,pueblo(_087C),def(_0B64,hermana(_0B64),def(_062C,hijos(_062C) &
 de(_0B64,_062C),def(_062C,(niños(_062C) & rubios(_062C)) &
 de(_087C,_062C),def(_0E14,futbol(_0E14),muy(bien(juegan(_062C,noHayOD,_0E14)))))))))
 Oración Correcta Nº 386 --> FL SN-Verbolnf
 [me,gusta,fumar,puros,en,mi,casa,.]
 *** decl(def(_3D0C,casa(_3D0C),en(_3D0C,gusta(noHaySuj,fumar(_16D8) & puros(_16D8),noHayOI))))
 Oración Correcta Nº 387 --> FL SN-Verbolnf
 [me,gusta,fumar,un,puro,en,mi,casa,.]

 decl(existeUn(_2638,puro(_2638),def(_59D0,casa(_59D0),en(_59D0,gusta(noHaySuj,fumar(_2638),noHay
 OI)))))))))
 Oración Correcta Nº 388 --> FL SN-Verbolnf
 [me,gusta,jugar,al,futbol,.]
 *** decl(def(_3464,futbol(_3464),gusta(noHaySuj,jugar(_3464),noHayOI)))
 Oración Correcta Nº 389 --> FL SN-Verbolnf
 [me,gusta,comer,de,pie,.]
 *** decl(gusta(noHaySuj,comer(_166C) & de(pie(_166C),_166C),noHayOI))
 Oración Correcta Nº 390 --> FL Cuant.Suj-OD-OI
 [todos,los,hombres,cantan,un,bolero,a,sus,mujeres,.]

 decl(todo(_11A4,hombres(_11A4),existeUn(_26FC,bolero(_26FC),def(_4C2C,mujeres(_4C2C),cantan(_11
 A4,_26FC,_4C2C)))))))))
 Oración Correcta Nº 391 --> FL Cuant.Suj
 [esta,mujer,fuma,puros,.]
 *** decl(def(_2564,mujer(_2564),fuma(_2564,puros(_5260),noHayOI)))
 Oración Correcta Nº 392 --> FL Cuant.Suj-OD
 [unas,mujeres,fuman,un,puro,.]

 decl(existeVarios(_1454,mujeres(_1454),existeUn(_40F0,puro(_40F0),fuman(_1454,_40F0),noHayOI)))
 Oración Correcta Nº 393 --> FL Cuant.Suj-OI
 [el,hermano,canta,a,su,hermana,.]
 *** decl(def(_108C,hermano(_108C),def(_3F4C,hermana(_3F4C),canta(_108C,noHayOD,_3F4C))))
 Oración Correcta Nº 394 --> FL Cuant.Adj.Numeral

<p>[cuatro,hombres,fuman,un,puro,.]</p> <p>***</p> <p>decl(existe(cuatro,_1068,hombres(_1068),existeUn(_3D48,puro(_3D48),fuman(_1068,_3D48,noHayOI)))</p> <p>Oración Correcta Nº 395 --> FL Cuant.Adv.Comp+Adj.Numeral</p> <p>[más,de,cuatro,hombres,fuman,un,puro,.]</p> <p>***</p> <p>decl(existe(más,cuatro,_10EC,hombres(_10EC),existeUn(_3A74,puro(_3A74),fuman(_10EC,_3A74,noHayOI)))</p> <p>Oración Correcta Nº 400 --> FL Coord.SN-Suj</p> <p>[Pepe,,Antonio,y,Marga,fuman,puros,.]</p> <p>*** decl(fuman(Pepe & Antonio & Marga,puros(_08E0),noHayOI))</p> <p>Oración Correcta Nº 401 --> FL Coord.SN-Suj</p> <p>[tu,hermana,,Pepe,y,su,amigo,fuman,puros,.]</p> <p>*** decl(def(_1148,hermana(_1148),def(_1E38,amigo(_1E38),fuman(_1148 & Pepe & _1E38,puros(_390C),noHayOI)))</p> <p>Oración Correcta Nº 402 --> FL Coord.SN-OI</p> <p>[Pepe,canta,a,Juan,,un,amigo,suyo,y,su,hermana,.]</p> <p>*** decl(existeUn(_08CC,amigo(_08CC),def(_09F4,hermana(_09F4),canta(Pepe,noHayOD,Juan & _08CC & _09F4)))</p> <p>Oración Correcta Nº 403 --> FL Coord.SP-OI</p> <p>[Pepe,canta,a,un,amigo,,a,Juan,y,a,su,hermana,.]</p> <p>*** decl(existeUn(_0808,amigo(_0808),def(_0B58,hermana(_0B58),canta(Pepe,noHayOD,_0808 & Juan & _0B58)))</p> <p>Oración Correcta Nº 404 --> FL Coord.Adj cal</p> <p>[Pepe,come,una,manzana,verde,oscura,,roja,y,azulada,.]</p> <p>*** decl(existeUn(_0C58,manzana(_0C58) & (verde(_0C58) & oscura(_0C58)) & roja(_0C58) & azulada(_0C58),come(Pepe,_0C58,noHayOI))</p> <p>Oración Correcta Nº 405 --> FL Coord.Adj cal</p> <p>[Pepe,es,rico,y,famoso,.]</p> <p>*** decl(rico(Pepe) & famoso(Pepe))</p> <p>Oración Correcta Nº 406 --> FL Coord.Adj cal</p> <p>[su,manzana,es,verde,o,roja,.]</p> <p>*** decl(def(_1080,manzana(_1080),verde(_1080) & roja(_1080)))</p> <p>Oración Correcta Nº 407 --> FL Coord.Adj cal</p> <p>[la,verde,y,roja,me,gusta,.]</p> <p>*** decl(def(_10A8,verde(_10A8) & roja(_10A8),gusta(_10A8,noHayOD,noHayOI))</p> <p>Oración Correcta Nº 408 --> FL Coord.Copul-OD-Adj</p> <p>[las,manzanas,y,las,peras,son,verdes,o,rojas,.]</p> <p>*** decl(def(_05F0,manzanas(_05F0),def(_06EC,peras(_06EC),verdes(_05F0 & _06EC) & rojas(_05F0 & _06EC)))</p> <p>Oración Correcta Nº 409 --> FL Coord.Copul-OD-Adj</p> <p>[Juan,y,Pepe,son,famosos,y,ricos,.]</p> <p>*** decl(famosos(Juan & Pepe) & ricos(Juan & Pepe))</p> <p>Oración Correcta Nº 410 --> FL Coord.Copul-OD-Sust</p> <p>[Juan,y,Pepe,son,unos,amigos,y,compañeros,en,Alicante,.]</p> <p>*** decl(existeVarios(Juan & Pepe,amigos(Juan & Pepe),en(Alicante,(Juan & Pepe) & compañeros(_0944)))</p> <p>Oración Correcta Nº 411 --> FL Coord.Copul-OD-Sust</p> <p>[los,perros,y,los,gatos,son,enemigos,o,adversarios,.]</p> <p>*** decl(def(_05EC,perros(_05EC),def(_06E8,gatos(_06E8),enemigos(_05EC & _06E8) & adversarios(_0970)))</p> <p>Oración Correcta Nº 412 --> FL Coord.Copul-CC</p> <p>[un,amigo,mio,es,de,Alicante,o,de,Madrid,.]</p> <p>*** decl(existeUn(_05DC,amigo(_05DC),de(Alicante,_05DC) & de(Madrid,_05DC)))</p> <p>Oración Correcta Nº 413 --> FL Coord.Copul-CC</p>

[un, amigo, mio, y, su, mujer, son, de, Alicante, o, Madrid, .]

*** decl(existeUn(_0600, amigo(_0600), def(_0728, mujer(_0728), de(Alicante & Madrid, _0600 & _0728)))

Oración Correcta Nº 414 → FL Coord. Copul-CC

[un, amigo, mio, y, su, mujer, son, de, un, pueblo, o, una, ciudad, .]

decl(existeUn(_0B20, amigo(_0B20), def(_0C48, mujer(_0C48), existeUn(_0EC4, pueblo(_0EC4), existeUn(_0FC0, ciudad(_0FC0), de(_0EC4 & _0FC0, _0B20 & _0C48))))))

Oración Correcta Nº 415 → FL Coord. Orac

[Pepe, mira, , come, y, pela, una, manzana, verde, .]

*** decl(existeUn(_0908, manzana(_0908) & verde(_0908), mira(Pepe, _0908, noHayOl)) & existeUn(_0908, manzana(_0908) & verde(_0908), come(Pepe, _0908, noHayOl)) & existeUn(_0908, manzana(_0908) & verde(_0908), pela(Pepe, _0908, noHayOl)))

Oración Correcta Nº 416 → FL Coord. CC-SP

[un, amigo, mio, canta, en, Alicante, , en, Madrid, y, en, Sevilla, .]

*** decl(existeUn(_05F4, amigo(_05F4), en(Alicante, canta(_05F4, noHayOD, noHayOl)) & en(Madrid, canta(_05F4, noHayOD, noHayOl)) & en(Sevilla, canta(_05F4, noHayOD, noHayOl)))

Oración Correcta Nº 417 → FL Coord. CC-SP-Adv

[un, amigo, mio, canta, en, Alicante, o, en, Madrid, mañana, .]

*** decl(existeUn(_1198, amigo(_1198), en(Alicante, mañana(canta(_1198, noHayOD, noHayOl))) & en(Madrid, mañana(canta(_1198, noHayOD, noHayOl))))))

Oración Correcta Nº 418 → FL Coord. CC-Adv-SP

[un, amigo, mio, juega, mañana, a, futbol, en, una, calle, o, en, una, ciudad, .]

decl(existeUn(_0604, amigo(_0604), existeUn(_0A88, calle(_0A88), existeUn(_0C48, ciudad(_0C48), mañana(en(_0A88, juega(_0604, noHayOD, futbol(_08C0))) & en(_0C48, juega(_0604, noHayOD, futbol(_08C0))))))

Oración Correcta Nº 419 → FL Coord. CC-SP-SP

[un, amigo, mio, juega, durante, el, día, y, durante, la, noche, a, futbol, en, una, calle, o, en, otra, .]

decl(existeUn(_07C4, amigo(_07C4), def(_0A7C, día(_0A7C), def(_0C3C, noche(_0C3C), existeUn(_0FD8, calle(_0FD8), durante(_0A7C, en(_0FD8, juega(_07C4, noHayOD, futbol(_0E10))) & en(otra(_1198), juega(_07C4, noHayOD, futbol(_0E10)))) & durante(_0C3C, en(_0FD8, juega(_07C4, noHayOD, futbol(_0E10))) & en(otra(_1198), juega(_07C4, noHayOD, futbol(_0E10))))))

Oración Correcta Nº 420 → FL Coord. SN-Suj/SN-OD

[Pepe, y, un, amigo, suyo, cantan, una, copla, y, un, bolero, .]

decl(existeUn(_06B8, amigo(_06B8), existeUn(_08B0, copla(_08B0), existeUn(_09AC, bolero(_09AC), cantan(Pepe & _06B8, _08B0 & _09AC, noHayOl))))))

Oración Correcta Nº 421 → FL Coord. SN-DI

[un, amigo, mio, de, Alicante, o, de, Madrid, juega, .]

*** decl(existeUn(_1158, amigo(_1158) & de(Alicante, _1158) & de(Madrid, _1158), juega(_1158, noHayOD, noHayOl)))

Oración Correcta Nº 422 → FL Coord. SN-DI

[un, amigo, mio, de, Alicante, o, de, un, pueblo, de, Alicante, juega, .]

*** decl(existeUn(_0988, pueblo(_0988) & de(Alicante, _0988), existeUn(_05F4, amigo(_05F4) & de(Alicante, _05F4) & de(_0988, _05F4), juega(_05F4, noHayOD, noHayOl))))))

Oración Correcta Nº 423 → FL Coord. SN-GSN

[un, amigo, mio, contando, un, chiste, y, una, anécdota, da, risa, .]

decl(existeUn(_07C8, chiste(_07C8), existeUn(_08C4, anécdota(_08C4), existeUn(_05F4, amigo(_05F4) & contando(_07C8 & _08C4, _05F4), da(_05F4, risa(_0A80), noHayOl))))))

Oración Correcta Nº 430 → FL Elip. Suj-Obj

[tu, mujer, y, su, amiga, cuentan, y, rien, un, chiste, muy, gracioso, a, Pepe, en, tu, casa, .]

*** decl(def(_0638, mujer(_0638), def(_0734, amiga(_0734), existeUn(_09D8, chiste(_09D8) & muy(gracioso(_09D8), def(_0DF8, casa(_0DF8), en(_0DF8, cuentan(_0638 & _0734, _09D8, Pepe)))))) & def(_0638, mujer(_0638), def(_0734, amiga(_0734), existeUn(_09D8, chiste(_09D8) & muy(gracioso(_09D8), def(_0DF8, casa(_0DF8), en(_0DF8, rien(_0638 & _0734, _09D8, Pepe))))))

Oración Correcta Nº 431 → FL Elip. Suj-Obj

<p>[Pepe, cuenta, y, rie, un, chiste, .]</p> <p>*** decl(existeUn(_09B0, chiste(_09B0), cuenta(Pepe, _09B0, noHayOI)) & existeUn(_09B0, chiste(_09B0), rie(Pepe, _09B0, noHayOI)))</p> <p>Oración Correcta Nº 432 → FL Elip. Verbal</p> <p>[el, ingeniero, conectó, la, impresora, y, Juan, el, servidor, .]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)</p> <p>*** decl(def(_0710, ingeniero(_0710), conectó(_0710, noHayOD, noHayOI)) & def(_0B8C, servidor(_0B8C), conectó(noHaySuj, _0B8C, noHayOI)))</p> <p>Oración Correcta Nº 433 → FL ELIP. V. ENC-SP+CC</p> <p>[Pepe, canta, una, copla, a, su, hermana, y, Antonio, a, Ana, .]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(existeUn(_0770, copla(_0770), def(_0944, hermana(_0944), canta(Pepe, _0770, _0944))) & existeUn(_0770, copla(_0770), canta(Antonio, _0770, Ana)))</p> <p>Oración Correcta Nº 434 → FL ELIP. V. ENC-ADV</p> <p>[Pepe, canta, muy, bien, un, bolero, a, él, en, Alicante, pero, Juan, no, .]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(existeUn(_0990, bolero(_0990), muy(bien(en(Alicante, canta(Pepe, _0990, él(_0B64)))))) & existeUn(_0990, bolero(_0990), no(muy(bien(en(Alicante, canta(Juan, _0990, él(_0B64)))))))))</p> <p>Oración Correcta Nº 435 → FL ELIP. V. ENC-ADV</p> <p>[Pepe, canta, muy, bien, los, boleros, , Antonio, no, y, Ana, sí, .]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(muy(bien(canta(Pepe, noHayOD, noHayOI))) & no(muy(bien(canta(noHaySuj, noHayOD, noHayOI))) & muy(bien(canta(Ana, noHayOD, noHayOI))))))</p> <p>Oración Correcta Nº 436 → FL APOSIC</p> <p>[aquellos, niños, rubios, de, tu, pueblo, , los, hijos, de, tu, hermana, , corren, y, juegan, al, futbol, .]</p> <p>*** decl(def(_087C, pueblo(_087C), def(_0B64, hermana(_0B64), def(_062C, hijos(_062C) & de(_0B64, _062C), def(_062C, (niños(_062C) & rubios(_062C))) & de(_087C, _062C), def(_0E7C, futbol(_0E7C), corren(_062C, noHayOD, _0E7C)))))) & def(_087C, pueblo(_087C), def(_0B64, hermana(_0B64), def(_062C, hijos(_062C) & de(_0B64, _062C), def(_062C, (niños(_062C) & rubios(_062C))) & de(_087C, _062C), def(_0E7C, futbol(_0E7C), juegan(_062C, noHayOD, _0E7C))))))</p> <p>Oración Correcta Nº 450 → 1.(a)</p> <p>[María, edita, e, imprime, el, informe, .]</p> <p>*** decl(def(_0818, informe(_0818), edita(María, _0818, noHayOI)) & def(_0818, informe(_0818), imprime(María, _0818, noHayOI)))</p> <p>Oración Correcta Nº 451 → 1.(b)</p> <p>[Juan, y, Pedro, van, a, la, catedral, .]</p> <p>*** decl(def(_0898, catedral(_0898), van(Juan & Pedro, noHayOD, _0898)))</p> <p>Oración Correcta Nº 452 → 1.(c)</p> <p>[Juan, lo, pidió, a, Pedro, y, María, .]</p> <p>*** decl(pidió(Juan, noHayOD, Pedro & María))</p> <p>Oración Correcta Nº 453 → 1.(d)</p> <p>[Juan, es, rico, y, famoso, .]</p> <p>*** decl(rico(Juan) & famoso(Juan))</p> <p>Oración Correcta Nº 454 → 1.(e)</p> <p>[María, trabaja, por, y, para, ganar, dinero, .]</p> <p>*** decl(por(ganar(_1F74) & dinero(_1F74), trabaja(María, noHayOD, noHayOI)) & para(ganar(_1F74) & dinero(_1F74), trabaja(María, noHayOD, noHayOI)))</p> <p>Oración Correcta Nº 455 → 1.(f)</p> <p>[Juan, camina, lenta, y, cuidadosamente, .]</p> <p>*** POSIBLE ELIPSIS VERBAL: Tipo 1 (Conjunción no analizada)</p> <p>*** decl(lenta(camina(Juan, noHayOD, noHayOI)) & cuidadosamente(camina(Juan, noHayOD, noHayOI)))</p> <p>Oración Correcta Nº 456 → 1.(g)</p> <p>[uno, y, otro, van, a, Valencia, .]</p> <p>*** decl(van(uno(_2228) & otro(_3F74), noHayOD, Valencia))</p>
--

Oración Correcta Nº 457 → 1.(h)
 [Pedro,y,Juan,son,parecidos,.]
 *** decl(parecidos(Pedro & Juan))

Oración Correcta Nº 458 → 1.(h)
 [la,gramática,categorial,y,la,transformacional,divergen,en,muchos,aspectos,.]
 *** decl(def_074C,gramática_074C) & categorial_074C,def_08C0,transformacional_08C0,def_0B80,aspectos_0B80,en_0B80,divergen_074C & _08C0,noHayOD,noHayOI))))))

Oración Correcta Nº 460 → 1.(j)
 [la,economía,está,determinada,por,las,limitaciones,físicas,y,psíquicas,de,los,seres,humanos,.]
 *** decl(def_0604,economía_0604,def_0C4C,seres_0C4C) & humanos_0C4C,def_09B0,(limitaciones_09B0) & físicas_09B0) & psíquicas_09B0) & de_0C4C,_09B0),por_09B0,determinada_0604))))))

Oración Correcta Nº 461 → 2.(a)
 [el,ingeniero,conectó,la,impresora,y,cambió,el,servidor,.]
 *** decl(def_05DC,ingeniero_05DC,def_0798,impresora_0798,conectó_05DC,_0798,noHayOI))) & def_05DC,ingeniero_05DC,def_09C4,servidor_09C4,cambió_05DC,_09C4,noHayOI))))

Oración Correcta Nº 462 → 2.(b)
 [el,mecánico,cambió,el,carburador,del,coche,y,Juan,la,batería,.]
 *** POSIBLE ELIPSIS VERBAL: Tipo 2 (Conjunción ya analizada)
 *** decl(def_05C8,mecánico_05C8,cambió_05C8,noHayOD,noHayOI)) & def_0C1C,batería_0C1C,cambió(noHaySuj,_0C1C,noHayOI))

Oración Correcta Nº 464 → 2.(d)
 [Juan,vio,y,María,oyó,el,tren,.]
 *** decl(def_0924,tren_0924,vio(Juan,_0924,noHayOI)) & def_0924,tren_0924,oyó(María,_0924,noHayOI))

Oraciones Incorrectas

Oración Incorrecta Nº 1 → AUS-Or.declarativa minima
 [Antonio,.]
 Oración Incorrecta Nº 2 → AUS-Or.declarativa minima
 [Antonio,,un,bolero,.]
 Oración Incorrecta Nº 3 → AUS-Conj.Or.decl
 [Antonio,juega,y,.]
 Oración Incorrecta Nº 10 → CONC.NUM-Suj pron
 [tú,juegan,al,futbol,.]
 Oración Incorrecta Nº 11 → CONC.PERS-Suj pron
 [tú,juego,al,futbol,.]
 Oración Incorrecta Nº 12 → CONC.FUNC-Suj pron
 [mio,juega,al,futbol,.]
 Oración Incorrecta Nº 13 → CONC.FUNC-Suj pron
 [ti,juegas,al,futbol,.]
 Oración Incorrecta Nº 20 → CONC.NUM-Suj sust
 [Antonio,jugamos,.]
 Oración Incorrecta Nº 21 → CONC.PERS-Suj sust
 [Antonio,juego,.]
 Oración Incorrecta Nº 22 → AUS-Suj sust
 [de,Alicante,juega,.]
 Oración Incorrecta Nº 30 → CONC.NUM-Suj verbo Infinitivo
 [el,fumar,gustan,.]
 Oración Incorrecta Nº 31 → CONC.PERS-Suj verbo Infinitivo
 [el,fumar,gusto,.]

- Oración Incorrecta Nº 35 --> CONC.NUM-Suj adj
[el,gris,juegan,.]
- Oración Incorrecta Nº 36 --> CONC.PERS-Suj adj
[el,gris,juego,.]
- Oración Incorrecta Nº 37 --> AUS-SN adj
[gris,juega,.]
- Oración Incorrecta Nº 38 --> AUS-SN adj
[el,gris,mio,azul,juega,.]
- Oración Incorrecta Nº 40 --> CONC.NUM-SN sust
[los,gato,juega,.]
- Oración Incorrecta Nº 41 --> CONC.GEN-SN sust
[la,gato,juega,.]
- Oración Incorrecta Nº 42 --> CONC.NUM-SN sust
[mis,gato,juega,.]
- Oración Incorrecta Nº 43 --> CONC.GEN-SN sust
[nuestra,gato,juega,.]
- Oración Incorrecta Nº 44 --> CONC.NUM-SN sust
[estos,gato,de,tu,hermana,juega,.]
- Oración Incorrecta Nº 45 --> CONC.GEN-SN sust
[esta,gato,de,tu,hermana,juega,.]
- Oración Incorrecta Nº 46 --> CONC.NUM-SN sust
[unos,gato,de,tu,hermana,juega,.]
- Oración Incorrecta Nº 47 --> CONC.GEN-SN sust
[una,gato,de,tu,hermana,juega,.]
- Oración Incorrecta Nº 48 --> CONC.NUM-SN sust
[todo,los,publico,de,Alicante,fuma,.]
- Oración Incorrecta Nº 49 --> CONC.GEN-SN sust
[todo,la,publico,de,Alicante,fuma,.]
- Oración Incorrecta Nº 50 --> CONC.NUM-SN sust
[todo,unos,niño,hizo,aquello,.]
- Oración Incorrecta Nº 51 --> CONC.GEN-SN sust
[todo,una,niño,hizo,aquello,.]
- Oración Incorrecta Nº 52 --> ORDEN-SN sust
[unos,todos,niños,hicieron,aquello,.]
- Oración Incorrecta Nº 60 --> CONC.NUM-SN sust
[dos,gato,juegan,.]
- Oración Incorrecta Nº 61 --> CONC.NUM-SN sust
[más,de,dos,gato,juegan,.]
- Oración Incorrecta Nº 62 --> ORDEN-SN sust
[más,dos,de,gatos,juegan,.]
- Oración Incorrecta Nº 63 --> ORDEN-SN sust
[dos,más,de,gatos,juegan,.]
- Oración Incorrecta Nº 64 --> AUS-SN sust
[primer,gato,juega,.]
- Oración Incorrecta Nº 65 --> CONC.NUM-SN sust
[los,primer,gato,juega,.]
- Oración Incorrecta Nº 66 --> CONC.GEN-SN sust
[el,primera,gato,juega,.]
- Oración Incorrecta Nº 69 --> AUS-SN sust
[gatos,juegan,.]
- Oración Incorrecta Nº 70 --> CONC.NUM-SN sust
[un,antiguos,amigo,americano,fuma,.]
- Oración Incorrecta Nº 71 --> CONC.NUM-SN sust

[un, antiguo, amigo, americanos, fuma,.]
 Oración Incorrecta Nº 72 → CONC.NUM-SN sust
 [el, gato, más, azules, juega, bien,.]
 Oración Incorrecta Nº 73 → CONC.GEN-SN sust
 [un, antigua, amigo, americano, fuma,.]
 Oración Incorrecta Nº 74 → CONC.GEN-SN sust
 [un, antiguo, amigo, americana, fuma,.]
 Oración Incorrecta Nº 75 → AUS-SN sust-Adj verbo participio COPULATIVO
 [un, amigo, sido, fuma,.]
 Oración Incorrecta Nº 80 → CONC.NUM-SN sust
 [un, amigo, míos, fuma,.]
 Oración Incorrecta Nº 81 → CONC.PERS-SN sust
 [un, amigo, mía, fuma,.]
 Oración Incorrecta Nº 90 → CONC.NUM-SN sust
 [Pepe,,, niños, americanos,,, fuma,.]
 Oración Incorrecta Nº 91 → CONC.GEN-SN sust
 [Pepe,,, la, niña, americana,,, fuma,.]
 Oración Incorrecta Nº 92 → CONC.NUM-SN sust
 [Pepe,,, el, niños, americano,,, fuma,.]
 Oración Incorrecta Nº 93 → CONC.GEN-SN sust
 [Pepe,,, la, niño, americano,,, fuma,.]
 Oración Incorrecta Nº 94 → AUS (de la coma)-SN sust
 [el, perro, el, gato, juega,.]
 Oración Incorrecta Nº 95 → AUS-SN sust
 [Pepe, el, niño, americano,,, fuma,.]
 Oración Incorrecta Nº 96 → AUS-SN sust
 [Pepe,,, el, niño, americano, fuma,.]
 Oración Incorrecta Nº 97 → AUS-SN sust
 [Pepe,,, niño,,, fuma,.]
 Oración Incorrecta Nº 98 → AUS-SN sust
 [Pepe,,, negro,,, fuma,.]
 Oración Incorrecta Nº 100 → ORD-Suj sust
 [viejos, los, cuatro, perros, juegan,.]
 Oración Incorrecta Nº 101 → ORD-Suj sust
 [un, los, cuatro, perros, juegan,.]
 Oración Incorrecta Nº 102 → ORD-Suj sust
 [los, dos, cuatro, perros, juegan,.]
 Oración Incorrecta Nº 103 → ORD-Suj sust
 [los, perros, dos, juegan,.]
 Oración Incorrecta Nº 104 → ORD-Suj sust
 [de, Antonio, los, perros, juegan,.]
 Oración Incorrecta Nº 105 → ORD-Suj sust
 [los, perros, míos, tuyos, juegan,.]
 Oración Incorrecta Nº 110 → CONC.NUM-Conj.de SN
 [Antonio, y, Pepe, juega,.]
 Oración Incorrecta Nº 111 → CONC.PERS-Conj.de SN
 [Antonio, y, Pepe, jugamos,.]
 Oración Incorrecta Nº 112 → CONC.PERS-Conj.de SN
 [tú, y, yo, jugais,.]
 Oración Incorrecta Nº 113 → CONC.PERS-Conj.de SN
 [él, y, yo, jugais,.]
 Oración Incorrecta Nº 114 → CONC.PERS-Conj.de SN
 [él, y, ellas, jugamos,.]

Oración Incorrecta Nº 115 → CONC.PERS-Conj.de SN [nosotros,y,vosotros,jugais,.]
Oración Incorrecta Nº 116 → CONC.PERS-Conj.de SN [vosotros,y,tú,jugamos,.]
Oración Incorrecta Nº 117 → CONC.PERS-Conj.de SN [vosotros,y,ella,jugamos,.]
Oración Incorrecta Nº 118 → AUS-Conj.de SN [Antonio,,,Pepe,y,juegan,.]
Oración Incorrecta Nº 119 → AUS-Conj.de SN [juegan,Antonio,y,.]
Oración Incorrecta Nº 130 → CONC.NUM-SP [el,gato,del,niños,juega,.]
Oración Incorrecta Nº 131 → CONC.NUM-SP [juega,al,pies,.]
Oración Incorrecta Nº 132 → CONC.NUM-SP [el,gato,de,los,niño,juega,.]
Oración Incorrecta Nº 133 → CONC.FUNC-SP [Pepe,de,yo,jugaba,.]
Oración Incorrecta Nº 134 → CONC.FUNC-SP [Pepe,de,mio,jugaba,.]
Oración Incorrecta Nº 135 → AUS-Conj.de SP [el,gato,de,Antonio,y,de,juega,.]
Oración Incorrecta Nº 136 → CONC-Conj.de SP [el,gato,por,Antonio,juega,.]
Oración Incorrecta Nº 137 → CONC-Conj.de SP [el,gato,juega,por,y,a,jugar,.]
Oración Incorrecta Nº 138 → CONC-Conj.de SP [el,gato,juega,a,y,por,jugar,.]
Oración Incorrecta Nº 139 → CONC-Conj.de SP [el,gato,de,y,de,Antonio,juega,a,y,al,futbol,.]
Oración Incorrecta Nº 140 → CONC.NUM-Suces.ADJ.CAL [el,gato,grises,azulado,juega,.]
Oración Incorrecta Nº 141 → CONC.NUM-Suces.ADJ.CAL [el,gris,azulados,gato,juega,.]
Oración Incorrecta Nº 142 → CONC.GEN-Suces.ADJ.CAL [el,gato,gris,azulada,juega,.]
Oración Incorrecta Nº 143 → CONC.FUNC-Suces.ADJ.NO CAL [el,gato,gris,dos,juega,.]
Oración Incorrecta Nº 144 → AUS-coord.ADJ.CAL [el,gato,gris,y,juega,.]
Oración Incorrecta Nº 145 → CONC.NUM-coord.ADJ.CAL [los,gatos,grises,y,azulado,juegan,.]
Oración Incorrecta Nº 146 → CONC.NUM-coord.ADJ.CAL [los,gatos,gris,y,azulados,juegan,.]
Oración Incorrecta Nº 147 → CONC.GEN-coord.ADJ.CAL [los,gatos,bonitos,y,azuladas,juegan,.]
Oración Incorrecta Nº 160 → CONC.FUNC-Gerundio SN [yo,comiendo,tú,doy,risa,.]
Oración Incorrecta Nº 161 → CONC.FUNC-Gerundio SN [yo,comiendo,fumar,doy,risa,.]
Oración Incorrecta Nº 170 → AUS-Pred.SV [.]
Oración Incorrecta Nº 171 → AUS-Pred.SV-Verbolnf

[ella,gustar,.]
Oración Incorrecta Nº 172 --> CONC.NUM-Pred.SV

[ella,cantan,.]
Oración Incorrecta Nº 173 --> CONC.PERS-Pred.SV

[yo,canta,.]
Oración Incorrecta Nº 174 --> AUS-Pred.SV con OI

[ella,canta,a,.]
Oración Incorrecta Nº 175 --> AUS-Pred.SV con OI

[ella,canta,de,.]
Oración Incorrecta Nº 176 --> AUS-Pred.SV con OD:SUST

[ella,canta,un,.]
Oración Incorrecta Nº 177 --> CONC.FUNC-Pred.SV con OD:PRON(sp)

[lo,hice,mí,.]
Oración Incorrecta Nº 178 --> CONC.FUNC-Pred.SV con OD:PRON(comp)

[lo,hice,me,.]
Oración Incorrecta Nº 190 --> REP-Pred.SV CC

[ella,canta,a,Pepe,dos,boleros,por,la,calle,hoy,.]
Oración Incorrecta Nº 191 --> REP-Pred.SV CC

[ella,canta,a,Pepe,dos,boleros,por,la,calle,por,Alicante,.]
Oración Incorrecta Nº 192 --> AUS-Pred.SV CC

[ella,canta,muy,.]
Oración Incorrecta Nº 193 --> AUS-Pred.SV CC

[ella,canta,no,.]
Oración Incorrecta Nº 194 --> AUS-Pred.SV CC

[ella,canta,muy,muy,.]
Oración Incorrecta Nº 195 --> AUS-Pred.SV CC

[ella,canta,menos,menos,.]
Oración Incorrecta Nº 196 --> AUS-Pred.SV CC

[ella,canta,muy,no,.]
Oración Incorrecta Nº 210 --> CONC.NUM-Pred.SV Cop

[el,fumar,es,muy,malos,.]
Oración Incorrecta Nº 211 --> CONC.GEN-Pred.SV Cop

[el,fumar,es,muy,mala,.]
Oración Incorrecta Nº 212 --> CONC.NUM-Pred.SV Cop

[Pepe,es,un,amigos,para,todos,.]
Oración Incorrecta Nº 213 --> CONC.GEN-Pred.SV Cop

[Pepe,es,un,amiga,para,todos,.]
Oración Incorrecta Nº 214 --> CONC.NUM-Pred.SV Cop

[el,fumar,no,son,muy,malos,.]
Oración Incorrecta Nº 215 --> CONC.NUM-Pred.SV Cop

[el,fumar,no,son,muy,malo,.]
Oración Incorrecta Nº 216 --> AUS-Pred.SV Cop

[ella,está,.]
*** ERROR: Faltan objetos de oración: : copul
Oración Incorrecta Nº 230 --> AUS-Pred.SV trans-OI

[ella,mira,.]
*** ERROR: Faltan objetos de oración: : trans
Oración Incorrecta Nº 231 --> AUS-Pred.SV trans-OI

[ella,no,mira,.]
*** ERROR: Faltan objetos de oración: : trans
Oración Incorrecta Nº 232 --> AUS-Pred.SV trans-OI

[ella,mira,en,Alicante,.]
*** ERROR: Faltan objetos de oración: : trans

Oración Incorrecta Nº 233 → AUS-Pred.SV trans-OI
 [ella,mira,y,no,come,.]
 *** ERROR: Faltan objetos de oración: : trans

Oración Incorrecta Nº 240 → AUS-Pred.SV impers
 [Pepe,llueve,.]
 Oración Incorrecta Nº 241 → AUS-Pred.SV impers
 [le,llueve,.]
 Oración Incorrecta Nº 242 → AUS-Pred.SV impers
 [llueve,a,Alicante,.]
 Oración Incorrecta Nº 243 → AUS-Pred.SV impers
 [llueve,Antonio,.]
 Oración Incorrecta Nº 244 → AUS-Pred.SV impers
 [hay,.]
 *** ERROR: Faltan objetos de oración: : impersOD

Oración Incorrecta Nº 245 → AUS-Pred.SV impers
 [hay,en,Alicante,.]
 *** ERROR: Faltan objetos de oración: : impersOD

Oración Incorrecta Nº 246 → AUS-Pred.SV impers
 [hay,a,Antonio,.]
 Oración Incorrecta Nº 247 → AUS-Pred.SV impers
 [Antonio,hay,manzanas,.]
 Oración Incorrecta Nº 248 → AUS-Pred.SV impers
 [hay,muy,poco,.]
 *** ERROR: Faltan objetos de oración: : impersOD

Oración Incorrecta Nº 249 → AUS-Pred.SV impers
 [hay,y,sobra,.]
 *** ERROR: Faltan objetos de oración: : impersOD

Apéndice C: corpus de frases con anáfora

<i>%----- ANAFORA PRONOMINAL -----</i>
<i>%----- C-command constraint sobre pronombres no reflexivos</i>
<i>o(1010,'Anaf.pron.c-c.Sec.con sigte',['Ana','eligió','esta','.']).</i>
<i>o(1011,'Anaf.pron.c-c',['Marga','lo','compró','para,ella','.']).</i>
<i>o(1012,'Anaf.pron.c-c',['Pedro','y','Ana','entraron','y','él','compró','.']).</i>
<i>o(1013,'Anaf.pron.c-c',['Pepe','y','él','entraron','y','él','compró','.']).</i>
<i>o(1014,'Anáf.pron.c-c',['Pepe','y','María','entraron','con','Marga','y,con','Luisa','y','Ana','al,supermercado,ayer,y','ella','compró','.']).</i>
<i>o(1015,'Anaf.pron.c-c.Sec.sigte',['Ana','eligió','a','Pedro','.']).</i>
<i>o(1016,'Anaf.pron.c-c',['él','también','eligió','a,ella','.']).</i>
<i>o(1017,'Anaf.pron.c-c.Sec.Sigte',['Juan','y','Ana','entraron','.']).</i>
<i>o(1018,'Anáf.pron.c-c',['compré','con','Antonio','y','él','y,con','Pedro','y,ellos','.']).</i>
<i>o(1019,'Anáf.pron.c-c',['compré','con','Antonio','y,con','él','.']).</i>
<i>o(1020,'Anáf.pron.c-c.Sec.sigte',['Juan','estuvo','aquí','.']).</i>
<i>o(1021,'Anáf.pron.c-c',['Antonio','dijo,que','él','compró','un,libro','.']).</i>
<i>o(1022,'c-c Dom.SN',[el,hijo,de,'él','compró','un,libro','.']).</i>
<i>o(1023,'c-c Dom.SN.Sec.Sigte',['Juan','tiene,un,retrato','.']).</i>
<i>o(1024,'c-c Dom.SN',[el,retrato,de,'Pedro','de,'él','es,bonito','.']).</i>
<i>o(1025,'c-c roto por SN dentro de otro SN', ['Antonio','compró','un,libro,para','Ana','y,la,hermana,de,ella','.']).</i>
<i>o(1026,'c-c roto por SN dentro de otro SN', ['Marga','compró','un,libro,para','Ana','y,la,hermana,de,ella','.']).</i>
<i>o(1027,'c-c roto por SN dentro de otro SN', ['Antonio','tiene,un,libro,de,el,hermano,de,'él','.']).</i>
<i>%----- constraint sobre pronombres no reflexivos por persona</i>
<i>o(1030,'anáf.pron.', ['Pedro','y','Ana','entraron','y','tú','compraste','.']).</i>
<i>%----- constraint sobre pronombres no reflexivos por número</i>
<i>o(1040,'anáf.pron.', ['Antonio','y','Luisa','entraron,y,ellos,compraron','.']).</i>
<i>o(1041,'anáf.pron.', ['Antonio','entró','y,ellos,compraron','.']).</i>
<i>%----- constraint sobre pronombres no reflexivos por información semántica mediante IRSAS</i>
<i>o(1045,'anáf.pron.', [el,'ratón','paró','cerca,del,coche,ayer,y,'éste','comió','queso','.']).</i>
<i>o(1046,'anáf.pron.', [en,el,coche,'había','un,'ratón','ayer,y,'éste','comió','queso','.']).</i>
<i>o(1047,'anáf.pron.', ['Juan','compró','queso,y,zumo,ayer,y,'comió','éste','.']).</i>

- o(1048, 'anáf.pron.', ['Juan', 'compró', 'queso, y, zumo, ayer, y, 'bebió', 'éste', '.']).
 o(1049, 'anáf.pron.', ['Juan', 'compró', 'queso, y, zumo, ayer, y, 'él', 'comió', 'éste', '.']).

%—— preferencias sobre pronombres no reflexivos

- o(1050, 'Pref.por Juan', ['Pedro', 'entró', 'con, 'Juan', 'y, 'compraron', 'para, 'él', '.']).
 o(1051, 'Sin Pref.', ['Pedro', 'y, 'Pepe', 'entraron', 'y, 'compraron', 'para, 'él', '.']).
 o(1052, 'Sec. Sigte', ['Pedro', 'entró', 'con, 'Juan', '.']).
 o(1053, 'Pref.por Juan', ['compró', 'para, 'él', '.']).
 o(1054, 'Sec. Sigte', ['Pedro', 'entró', 'con, 'Juan', '.']).
 o(1055, 'Pref.por Pedro', ['él', 'compró', '.']).
 o(1056, 'Sec. Sigte', ['Pedro', 'entró', 'con, 'Juan', '.']).
 o(1057, 'Sec. Sigte', ['Pepe', 'también', 'entró', 'con, 'Luis', '.']).
 o(1058, 'Pref.por Pepe', ['él', 'compró', '.']).
 o(1059, 'Sec. Sigte', ['mañana', 'es, la, fiesta, de, 'Luis', '.']).
 o(1060, 'Pref.por Luis', ['Pedro', 'entró', 'con, 'Juan', 'y, 'compraron', 'un, regalo, para, 'él', '.']).
 o(1061, 'Pref.por la mujer', ['la, mujer, dijo, que, ella, era, graciosa', '.']).
 o(1062, 'Sec. Sigte', ['Luisa', 'entró', '.']).
 o(1063, 'Pref.por Luisa', ['la, mujer, dijo, que, ella, era, graciosa', '.']).
 o(1064, 'Sec. Sigte', ['Luisa', 'entró', 'en, la, tienda', '.']).
 o(1065, 'Pref.por Luisa', ['la, mujer, dijo, que, ella, era, graciosa', '.']).

%———— ANAFORA TIPO ADJETIVO —————

%—— Referencia al mismo antecedente

- o(1100, 'anáf.adj.', ['Luisa', 'y, 'Ana', 'compraron', 'una, pera, verde, y, una, manzana, roja, ayer, y, 'Pedro', 'eligió', 'la, verde', '.']).
 o(1101, 'anáf.adj.', ['Luisa', 'y, 'Ana', 'compraron', 'una, pera, verde, oscura, y, una, manzana, roja, hoy, y, 'Pedro', 'eligió', 'la, verde', '.']).
 o(1102, 'anáf.adj.', ['Luisa', 'y, 'Ana', 'compraron', 'una, pera, verde, oscura, y, una, manzana, roja, hoy, y, 'Pedro', 'eligió', 'la, oscura', '.']).
 o(1103, 'anáf.adj.', ['Luisa', 'y, 'Ana', 'compraron', 'una, pera, verde, oscura, y, una, manzana, roja, hoy, y, 'Pedro', 'eligió', 'la, roja', '.']).
 o(1104, 'Sec. sigte', ['Luisa', 'y, 'Ana', 'compraron', 'una, pera, verde, y, una, manzana, roja', '.']).
 o(1105, 'anáf.adj.', ['Pedro', 'eligió', 'la, verde', '.']).
 o(1106, 'Sec. sigte', ['Luisa', 'compró', 'una, pera, verde, de, 'Alicante', 'ayer, y, otra, pera, verde, de, 'Madrid', 'hoy', '.']).
 o(1107, 'anáf.adj.', ['Pedro', 'eligió', 'la, verde, de, 'Alicante', '.']).
 o(1108, 'anáf.adj.', ['Luisa', 'compró', 'una, pera, verde, y, un, libro, verde, hoy, y, 'Pedro', 'eligió', 'la, verde', '.']).

%—— Referencia a una nueva entidad del discurso

- o(1120, 'Nueva Ent.por difte.adj.', ['Luisa', 'y, 'Ana', 'compraron', 'una, pera, verde, hoy, y, 'Pedro', 'eligió', 'otra, roja', '.']).
 o(1121, 'Nueva Ent.por difte.adj.', ['Luisa', 'compró', 'una, pera, verde, y, otra, roja', '.']).
 o(1122, 'Nueva Ent.por difte.adj.', ['Luisa', 'compró', 'una, pera, verde, oscura, y, otra, roja', '.']).
 o(1123, 'Nueva Ent.por difte.nº', ['Luisa', 'compró', 'una, pera, verde, ayer, y, 'Ana', 'compró', 'dos, verdes', '.']).
 o(1124, 'Nueva Ent.por diftes.modif.', ['Luisa', 'compró', 'una, pera, verde, hoy, y, 'Ana', 'compró', 'otra, verde, de, 'Alicante', '.']).
 o(1125, 'Nueva Ent.por diftes.modif.', ['Luisa', 'compró', 'una, pera, verde, de, 'Madrid', 'hoy, y, una, manzana, roja, ayer, y, 'Ana', 'compró', 'otra, verde, de, 'Alicante', '.']).

Apéndice D: resultados obtenidos del tratamiento de la anáfora pronominal en el corpus del apéndice C

Oración Correcta Nº 1012 → Anaf.pron.c-c. [*Pedro,y,Ana,entraron,y,él,compró.,*]

Anáfora tipo pronSuj:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pedro

--- Sol.Anáf.

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pedro

Oración Correcta Nº 1013 → Anaf.pron.c-c [*Pepe,y,él,entraron,y,él,compró.,*]

Anáfora tipo pronSuj:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados: — OJO: Anáfora no resuelta o exófora

Anáfora tipo pronSuj:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pepe

— Sol.Anáf.**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pepe

Oración Correcta Nº 1014 → Anáf.pron.c-c

[Pepe,y,María,entraron,con,Marga,y,con,Luisa,y,Ana,al,supermercado,ayer,y,ella,compró,.]**Anáfora tipo pronSuj:****** SINT.NOMINAL SIMPLE:****** PRONOMBRE:**

ella

Lista de antecedentes seleccionados:**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Ana

**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Luisa

**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Marga

**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

María

— Sol.Anáf.**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

María

Oración Correcta Nº 1015 → Anaf.pron.c-c. Sec.sigte **[Ana,eligió,a,Pedro,.]**Oración Correcta Nº 1016 → Anaf.pron.c-c. **[él,también,eligió,a,ella,.]****Anáfora tipo pronSuj:****** SINT.NOMINAL SIMPLE:****** PRONOMBRE:**

él

Lista de antecedentes seleccionados:**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pedro

— Sol.Anáf.**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pedro

Anáfora tipo pronSP:**** SINT.NOMINAL SIMPLE:****** PRONOMBRE:**

ella

Lista de antecedentes seleccionados:**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Ana

— Sol. Anáf.

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Ana

Oración Correcta Nº 1017 → Anáf.pron.c-c.Sec.Sigte [Juan,y,Ana,entraron,.]

Oración Correcta Nº 1018 → Anáf.pron.c-c [compré,con,Antonio,y,él,y,con,Pedro,y,ellos,.]

Anáfora tipo pronSP:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

— Sol. Anáf.

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

Anáfora tipo pronSP:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

ellos

Lista de antecedentes seleccionados:

** SINT.NOMINAL:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

** CONJUNCION:

y

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Ana

— Sol. Anáf.

** SINT.NOMINAL:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

** CONJUNCION:

y

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Ana

Oración Correcta Nº 1019 → Anáf.pron.c-c [compré,con,Antonio,y,con,él,.]

Anáfora tipo pronSP:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados: — OJO: Anáfora no resuelta o exófora

Oración Correcta Nº 1020 → Anáf.pron.c-c.Sec.sigte [Juan,estuvo,aquí,.]

Oración Correcta Nº 1021 → Anáf.pron.c-c [Antonio,dijo,que,él,compró,un,libro,.]

Anáfora tipo pronSuj:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

<p>él</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Antonio</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Juan</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Antonio</p> <p>Oración Correcta Nº 1022 → c-c Dom.SN [el,hijo,de,él,compró,un,libro,,]</p> <p>Anáfora tipo pronSP:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>él</p> <p>Lista de antecedentes seleccionados: — OJO: Anáfora no resuelta o exófora</p> <p>Oración Correcta Nº 1023 → c-c Dom.SN.Sec.Sigte. [Juan,tiene,un,retrato,,]</p> <p>Oración Correcta Nº 1024 → c-c Dom.SN [el,retrato,de,Pedro,de,él,es,bonito,,]</p> <p>Anáfora tipo pronSP:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>él</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>un</p> <p>** SUSTANTIVO:</p> <p>retrato</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Juan</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Juan</p> <p>Oración Correcta Nº 1025 → c-c roto por SN dentro de otro SN</p> <p>[Antonio,compró,un,libro,para,Ana,y,la,hermana,de,ella,,]</p> <p>Anáfora tipo pronSP:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>ella</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Ana</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p>

<p>Ana</p> <p>Oración Correcta Nº 1026 → c-c roto por SN dentro de otro SN [Marga, compró, un, libro, para, Ana, y, la, hermana, de, ella,.]</p> <p>Anáfora tipo pronSP:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>ella</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Ana</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Marga</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Ana</p> <p>Oración Correcta Nº 1027 → c-c roto por SN dentro de otro SN [Antonio, tiene, un, libro, de, el, hermano, de, él,.]</p> <p>Anáfora tipo pronSP:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>él</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Antonio</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>un</p> <p>** SUSTANTIVO:</p> <p>libro</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Antonio</p> <p>Oración Correcta Nº 1030 → anáf.pron. [Pedro, y, Ana, entraron, y, tú, compraste,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>tú</p> <p>Lista de antecedentes seleccionados: --- OJO: Anáfora no resuelta o exófora</p> <p>Oración Correcta Nº 1040 → anáf.pron. [Antonio, y, Luisa, entraron, y, ellos, compraron,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>ellos</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL:</p> <p>** SINT.NOMINAL SIMPLE:</p>

<p>** SUSTANTIVO: Antonio</p> <p>** CONJUNCION: y</p> <p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: Luisa</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL: ** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: Antonio</p> <p>** CONJUNCION: y</p> <p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: Luisa</p> <p>Oración Correcta Nº 1041 --> anáf.pron. [Antonio,entró,y,ellos,compraron,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE: ** PRONOMBRE: ellos</p> <p>Lista de antecedentes seleccionados: — OJO: Anáfora no resuelta o exófora</p> <p>Oración Correcta Nº 1045 --> anáf.pron. [el,ratón,paró,cerca,del,coche,ayer,y,éste,comió,queso,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE: ** PRONOMBRE: éste</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: el</p> <p>** SUSTANTIVO: coche</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: el</p> <p>** SUSTANTIVO: ratón</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: el</p> <p>** SUSTANTIVO: ratón</p> <p>Oración Correcta Nº 1046 --> anáf.pron. [en,el,coche,había,un,ratón,ayer,y,éste,comió,queso,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE: ** PRONOMBRE:</p>
--

<p>éste</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p> ** ARTICULO:</p> <p> un</p> <p>** SUSTANTIVO:</p> <p> ratón</p> <p> ** SINT.NOMINAL SIMPLE:</p> <p> ** DETERMINANTE 1:</p> <p> ** ARTICULO:</p> <p> el</p> <p> ** SUSTANTIVO:</p> <p> coche</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p> ** ARTICULO:</p> <p> el</p> <p> ** SUSTANTIVO:</p> <p> ratón</p> <p>Oración Correcta Nº 1047 → anáf.pron. [Juan,compró,queso,y,zumo,ayer,y,comió,éste,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p> éste</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p> zumo</p> <p> ** SINT.NOMINAL SIMPLE:</p> <p> ** SUSTANTIVO:</p> <p> queso</p> <p> ** SINT.NOMINAL SIMPLE:</p> <p> ** SUSTANTIVO:</p> <p> Juan</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p> queso</p> <p>Oración Correcta Nº 1048 → anáf.pron. [Juan,compró,queso,y,zumo,ayer,y,bebió,éste,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p> éste</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p> zumo</p>

<p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: queso</p> <p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: Juan</p> <p>-- Sol.Anáf. ** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: zumo</p> <p>Oración Correcta Nº 1049 → anáf.pron. [Juan,compró,queso,y,zumo,ayer,y,él,comió,éste,]</p> <p>Anáfora tipo pronSuj: ** SINT.NOMINAL SIMPLE: ** PRONOMBRE: él</p> <p>Lista de antecedentes seleccionados: ** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: zumo</p> <p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: queso</p> <p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: Juan</p> <p>-- Sol.Anáf. ** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: Juan</p> <p>Anáfora tipo pronSuj: ** SINT.NOMINAL SIMPLE: ** PRONOMBRE: éste</p> <p>Lista de antecedentes seleccionados: ** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: zumo</p> <p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: queso</p> <p>** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: Juan</p> <p>-- Sol.Anáf. ** SINT.NOMINAL SIMPLE: ** SUSTANTIVO: queso</p> <p>Oración Correcta Nº 1050 → Pref.por Juan [Pedro,entró,con,Juan,y,compraron,para,él,.]</p>
--

Anáfora tipo pronSP:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pedro

--- Sol.Anáf.

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

Oración Correcta Nº 1051 --> Sin Pref. [Pedro,y,Pepe,entraron,y,compraron,para,él,.]

Anáfora tipo pronSP:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pepe

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pedro

--- Sol.Anáf.

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pedro

Oración Correcta Nº 1052 --> Sec. Sigte [Pedro,entró,con,Juan,.]

Oración Correcta Nº 1053 --> Pref.por Juan [compró,para,él,.]

Anáfora tipo pronSP:

** SINT.NOMINAL SIMPLE:

** PRONOMBRE:

él

Lista de antecedentes seleccionados:

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Pedro

--- Sol.Anáf.

** SINT.NOMINAL SIMPLE:

** SUSTANTIVO:

Juan

Oración Correcta Nº 1054 --> Sec. Sigte [Pedro,entró,con,Juan,.]

Oración Correcta Nº 1055 --> Pref.por Pedro [él,compró,.]

Anáfora tipo pronSuj:**** SINT.NOMINAL SIMPLE:****** PRONOMBRE:**

él

Lista de antecedentes seleccionados:**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Juan

**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pedro

— Sol.Anáf.**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pedro

Oración Correcta Nº 1054 --> Sec.Sigte [**Pedro,entró,con,Juan,.**]Oración Correcta Nº 1055 --> Pref.por Pedro [**él,compró,.**]**Anáfora tipo pronSuj:****** SINT.NOMINAL SIMPLE:****** PRONOMBRE:**

él

Lista de antecedentes seleccionados:**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Juan

**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pedro

— Sol.Anáf.**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pedro

Oración Correcta Nº 1056 --> Sec.Sigte [**Pedro,entró,con,Juan,.**]Oración Correcta Nº 1057 --> Sec.Sigte [**Pepe,también,entró,con,Luis,.**]Oración Correcta Nº 1058 --> Pref.por Pepe [**él,compró,.**]**Anáfora tipo pronSuj:****** SINT.NOMINAL SIMPLE:****** PRONOMBRE:**

él

Lista de antecedentes seleccionados:**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Luis

**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pepe

— Sol.Anáf.**** SINT.NOMINAL SIMPLE:****** SUSTANTIVO:**

Pepe

<p>Oración Correcta Nº 1059 → Sec. Sigte [mañana,es,la,fiesta,de,Luis,.]</p> <p>Oración Correcta Nº 1060 → Pref.por Juan [Pedro,entró,con,Juan,y,compraron,un,regalo,para,él,.]</p> <p>Anáfora tipo pronSP:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>él</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Juan</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Pedro</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Luis</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Juan</p> <p>Oración Correcta Nº 1061 → Pref.por la mujer [la,mujer,dijo,que,ella,era,graciosa,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>ella</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>la</p> <p>** SUSTANTIVO:</p> <p>mujer</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>la</p> <p>** SUSTANTIVO:</p> <p>mujer</p> <p>Oración Correcta Nº 1062 → Sec. Sigte [Luisa,entró,.]</p> <p>Oración Correcta Nº 1063 → Pref.por Luisa [la,mujer,dijo,que,ella,era,graciosa,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>ella</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>la</p> <p>** SUSTANTIVO:</p>

<p>mujer</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Luisa</p> <p>--- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>la</p> <p>** SUSTANTIVO:</p> <p>mujer</p> <p>Oración Correcta Nº 1064 → Sec. Sigte [Luisa,entró,en,la,tienda,.]</p> <p>Oración Correcta Nº 1065 → Pref.por Luisa [la,mujer,dijo,que,ella,era,graciosa,.]</p> <p>Anáfora tipo pronSuj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** PRONOMBRE:</p> <p>ella</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>la</p> <p>** SUSTANTIVO:</p> <p>mujer</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>la</p> <p>** SUSTANTIVO:</p> <p>tienda</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Luisa</p> <p>--- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>la</p> <p>** SUSTANTIVO:</p> <p>mujer</p>

Apéndice E: resultados obtenidos del tratamiento de la anáfora de tipo adjetivo en el corpus del apéndice C

Oración Correcta Nº 1100 --> anáf.adj

[Luisa,y,Ana,compraron,una,pera,verde,y,una,manzana,roja,hoy,y,Pedro,eligió,la,verde,.]

Anáfora tipo adj:

** SINT.NOMINAL SIMPLE:

** DETERMINANTE 1:

** ARTICULO:

la

** ADYACENTE ADJETIVO:

** ADJETIVO SIMPLE:

verde

Lista de antecedentes seleccionados:

** SINT.NOMINAL SIMPLE:

** DETERMINANTE 1:

** ARTICULO:

una

** SUSTANTIVO:

manzana

** ADYACENTE ADJETIVO:

** ADJETIVO SIMPLE:

roja

** SINT.NOMINAL SIMPLE:

** DETERMINANTE 1:

** ARTICULO:

una

** SUSTANTIVO:

pera

<p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>Oración Correcta Nº 1101 → anáf.adj [Luisa,y,Ana,compraron,una,pera,verde,oscura,y,una,manzana,roja,hoy,y,Pedro,eligió,la,verde,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: la</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: manzana</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: roja</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>** ADJETIVO SIMPLE: oscura</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO:</p>
--

<p>** ADJETIVO SIMPLE: verde</p> <p>** ADJETIVO SIMPLE: oscura</p> <p>Oración Correcta Nº 1102 → anáf. adj [Luisa,y,Ana,compraron,una,pera,verde,oscura,y,una,manzana,roja,hoy,y,Pedro,eligió,la,oscura,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO: la</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE: oscura</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO: una</p> <p>** SUSTANTIVO: manzana</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE: roja</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE: verde</p> <p>** ADJETIVO SIMPLE: oscura</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE: verde</p> <p>** ADJETIVO SIMPLE: oscura</p> <p>Oración Correcta Nº 1103 → anáf. adj [Luisa,y,Ana,compraron,una,pera,verde,oscura,y,una,manzana,roja,hoy,y,Pedro,eligió,la,roja,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p>

<p>** ARTICULO: la</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: roja</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: manzana</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: roja</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>** ADJETIVO SIMPLE: oscura</p> <p>— Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: manzana</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: roja</p> <p>Oración Correcta Nº 1104 --> Sec.sigte [Luisa,y,Ana,compraron,una,pera,verde,y,una,manzana,roja,.]</p> <p>Oración Correcta Nº 1105 --> anáf.adj. [Pedro,eligió,la,verde,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: la</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO:</p>

<p>manzana</p> <p>** ADYACENTE ADJETIVO:</p> <p> ** ADJETIVO SIMPLE:</p> <p> roja</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p> ** ARTICULO:</p> <p> una</p> <p>** SUSTANTIVO:</p> <p> pera</p> <p>** ADYACENTE ADJETIVO:</p> <p> ** ADJETIVO SIMPLE:</p> <p> verde</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p> ** ARTICULO:</p> <p> una</p> <p>** SUSTANTIVO:</p> <p> pera</p> <p>** ADYACENTE ADJETIVO:</p> <p> ** ADJETIVO SIMPLE:</p> <p> verde</p> <p>Oración Correcta Nº 1106 → Sec.sigte [Luisa,compró,una,pera,verde,de,Alicante,ayer,y,otra,pera,verde,de,Madrid,hoy,.]</p> <p>Oración Correcta Nº 1107 → anáf.adj. [Pedro,eligió,la,verde,de,Alicante,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p> ** ARTICULO:</p> <p> la</p> <p>** ADYACENTE ADJETIVO:</p> <p> ** ADJETIVO SIMPLE:</p> <p> verde</p> <p>** SINT.PREPOSICIONAL:</p> <p> ** SINT.PREPOSICIONAL SIMPLE:</p> <p> ** PREPOSICION:</p> <p> ** PREPOSICION SIMPLE:</p> <p> de</p> <p> ** SINT.NOMINAL:</p> <p> ** SINT.NOMINAL SIMPLE:</p> <p> ** SUSTANTIVO:</p> <p> Alicante</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p> ** ADJETIVO SIMPLE:</p> <p> otra</p> <p>** SUSTANTIVO:</p> <p> pera</p> <p>** ADYACENTE ADJETIVO:</p> <p> ** ADJETIVO SIMPLE:</p>

<p>verde</p> <p>** SINT.PREPOSICIONAL:</p> <p>** SINT.PREPOSICIONAL SIMPLE:</p> <p>** PREPOSICION:</p> <p>** PREPOSICION SIMPLE:</p> <p>de</p> <p>** SINT.NOMINAL:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Madrid</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>una</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verde</p> <p>** SINT.PREPOSICIONAL:</p> <p>** SINT.PREPOSICIONAL SIMPLE:</p> <p>** PREPOSICION:</p> <p>** PREPOSICION SIMPLE:</p> <p>de</p> <p>** SINT.NOMINAL:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Alicante</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>una</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verde</p> <p>** SINT.PREPOSICIONAL:</p> <p>** SINT.PREPOSICIONAL SIMPLE:</p> <p>** PREPOSICION:</p> <p>** PREPOSICION SIMPLE:</p> <p>de</p> <p>** SINT.NOMINAL:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** SUSTANTIVO:</p> <p>Alicante</p> <p>Oración Correcta Nº 1108 --> anáf.adj [Luisa,compró,una,pera,verde,y,un,libro,verde,hoy,y,Pedro,eligió,la,verde,.] Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p>

*Apéndice E: resultados obtenidos del tratamiento
de la anáfora de tipo adjetivo en el corpus del apéndice C*

261

<p>** ARTICULO: la</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>Oración Correcta Nº 1120 -> Nueva Ent.por difte.adj [Luisa,y,Ana,compraron,una,pera,verde,hoy,y,Pedro,eligió,otra,roja,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ADJETIVO SIMPLE: otra</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: roja</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ARTICULO: una</p> <p>** SUSTANTIVO: pera</p> <p>** ADYACENTE ADJETIVO: ** ADJETIVO SIMPLE: verde</p> <p>Antec. y anaf. no co-refieren</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE: ** DETERMINANTE 1: ** ADJETIVO SIMPLE: otra</p> <p>** SUSTANTIVO: pera</p>
--

<p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>roja</p> <p>Oración Correcta Nº 1121 --> Nueva Ent.por difte.adj [Luisa,compró,una,pera,verde,y,otra,roja,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ADJETIVO SIMPLE:</p> <p>otra</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>roja</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>una</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verde</p> <p>Antec. y anaf. no co-refieren</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ADJETIVO SIMPLE:</p> <p>otra</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>roja</p> <p>Oración Correcta Nº 1122 --> Nueva Ent.por difte.adj [Luisa,compró,una,pera,verde,oscura,y,otra,roja,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ADJETIVO SIMPLE:</p> <p>otra</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>roja</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>una</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verde</p> <p>** ADJETIVO SIMPLE:</p>

<p>oscura</p> <p>Antec. y anaf. no co-refieren</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ADJETIVO SIMPLE:</p> <p>otra</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>roja</p> <p>Oración Correcta Nº 1123 -> Nueva Ent.por difte.nº [Luisa,compró,una,pera,verde,hoy,y,Ana,compró,dos,verdes,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 2:</p> <p>** ADJETIVO SIMPLE:</p> <p>dos</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verdes</p> <p>Lista de antecedentes seleccionados:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ARTICULO:</p> <p>una</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verde</p> <p>Antec. y anaf. no co-refieren</p> <p>-- Sol.Anáf.</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 2:</p> <p>** ADJETIVO SIMPLE:</p> <p>dos</p> <p>** SUSTANTIVO:</p> <p>pera</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verdes</p> <p>Oración Correcta Nº 1124 -> Nueva Ent.por diftes.modif. [Luisa,compró,una,pera,verde,hoy,y,Ana,compró,otra,verde,de,Alicante,.]</p> <p>Anáfora tipo adj:</p> <p>** SINT.NOMINAL SIMPLE:</p> <p>** DETERMINANTE 1:</p> <p>** ADJETIVO SIMPLE:</p> <p>otra</p> <p>** ADYACENTE ADJETIVO:</p> <p>** ADJETIVO SIMPLE:</p> <p>verde</p>
--

**** SINT.PREPOSICIONAL:**

**** SINT.PREPOSICIONAL SIMPLE:**

**** PREPOSICION:**

**** PREPOSICION SIMPLE:**

de

**** SINT.NOMINAL:**

**** SINT.NOMINAL SIMPLE:**

**** SUSTANTIVO:**

Alicante

Lista de antecedentes seleccionados:

**** SINT.NOMINAL SIMPLE:**

**** DETERMINANTE 1:**

**** ARTICULO:**

una

**** SUSTANTIVO:**

pera

**** ADYACENTE ADJETIVO:**

**** ADJETIVO SIMPLE:**

verde

Antec. y anaf. no co-refieren

-- Sol.Anáf.

**** SINT.NOMINAL SIMPLE:**

**** DETERMINANTE 1:**

**** ADJETIVO SIMPLE:**

otra

**** SUSTANTIVO:**

pera

**** ADYACENTE ADJETIVO:**

**** ADJETIVO SIMPLE:**

verde

**** SINT.PREPOSICIONAL:**

**** SINT.PREPOSICIONAL SIMPLE:**

**** PREPOSICION:**

**** PREPOSICION SIMPLE:**

de

**** SINT.NOMINAL:**

**** SINT.NOMINAL SIMPLE:**

**** SUSTANTIVO:**

Alicante

Oración Correcta Nº 1125 → Nueva Ent.por diftes.modif.

[Luisa,compró,una,pera,verde,de,Madrid,hoy,y,una,manzana,roja,ayer,y,Ana,compró,otra,verde,de,Alicante,.]

Anáfora tipo adj:

**** SINT.NOMINAL SIMPLE:**

**** DETERMINANTE 1:**

**** ADJETIVO SIMPLE:**

otra

**** ADYACENTE ADJETIVO:**

**** ADJETIVO SIMPLE:**

verde

**** SINT.PREPOSICIONAL:**

**** SINT.PREPOSICIONAL SIMPLE:**

**** PREPOSICION:**

*Apéndice E: resultados obtenidos del tratamiento
de la anáfora de tipo adjetivo en el corpus del apéndice C*

265

**** PREPOSICION SIMPLE:**

de

**** SINT.NOMINAL:**

**** SINT.NOMINAL SIMPLE:**

**** SUSTANTIVO:**

Alicante

Lista de antecedentes seleccionados:

**** SINT.NOMINAL SIMPLE:**

**** DETERMINANTE 1:**

**** ARTICULO:**

una

**** SUSTANTIVO:**

manzana

**** ADYACENTE ADJETIVO:**

**** ADJETIVO SIMPLE:**

roja

**** SINT.NOMINAL SIMPLE:**

**** DETERMINANTE 1:**

**** ARTICULO:**

una

**** SUSTANTIVO:**

pera

**** ADYACENTE ADJETIVO:**

**** ADJETIVO SIMPLE:**

verde

**** SINT.PREPOSICIONAL:**

**** SINT.PREPOSICIONAL SIMPLE:**

**** PREPOSICION:**

**** PREPOSICION SIMPLE:**

de

**** SINT.NOMINAL:**

**** SINT.NOMINAL SIMPLE:**

**** SUSTANTIVO:**

Madrid

Antec. y anaf. no co-refieren

-- Sol.Anáf.

**** SINT.NOMINAL SIMPLE:**

**** DETERMINANTE 1:**

**** ADJETIVO SIMPLE:**

otra

**** SUSTANTIVO:**

pera

**** ADYACENTE ADJETIVO:**

**** ADJETIVO SIMPLE:**

verde

**** SINT.PREPOSICIONAL:**

**** SINT.PREPOSICIONAL SIMPLE:**

**** PREPOSICION:**

**** PREPOSICION SIMPLE:**

de

**** SINT.NOMINAL:**

**** SINT.NOMINAL SIMPLE:**

Universitat d'Alacant
Universidad de Alicante

**** SUSTANTIVO:**

Alicante

Universitat d'Alacant
Universidad de Alicante

Apéndice F: resultados obtenidos del tratamiento de la anáfora en textos no restringidos

Oración Correcta Nº 1 ->

Recomendación A.21 COLABORACIÓN CON OTRAS ORGANIZACIONES INTERNACIONALES EN RELACIÓN CON LOS SERVICIOS DE TELEMÁTICA DEFINIDOS POR EL CCITT (Ginebra , 1980 ; modificada en Málaga - Torremolinos , 1984) El CCITT , considerando (a) que en el artículo 1 del acuerdo entre la Organización de las Naciones Unidas y la Unión Internacional de Telecomunicaciones , las Naciones Unidas reconocen a la Unión Internacional de Telecomunicaciones como el organismo especializado encargado de adoptar , de conformidad con su Acta constitutiva , las medidas necesarias para el cumplimiento de las funciones señaladas en la misma ; (b) que el artículo 4 del Convenio Internacional de Telecomunicaciones ^ (Nairobi , 1982) declara que la Unión tiene por objeto : a) mantener y ampliar la cooperación internacional entre todos los Miembros de la Unión para el mejoramiento y el empleo racional de toda clase de telecomunicación , así como promover y proporcionar asistencia técnica a los países en desarrollo en el campo de las telecomunicaciones ; b) favorecer el desarrollo de los medios técnicos y su más eficaz explotación , a fin de aumentar el rendimiento de los servicios de telecomunicación , acrecentar su empleo y generalizar lo más posible su utilización por el público ; c) armonizar los esfuerzos de las naciones para la consecución de estos fines .

Anáfora tipo adj:

la misma

Con paralelismo: para(_6897,3,dv,25,1)

Lista de antecedentes seleccionados:

su Acta constitutiva

para(_6897,3,dv,25,1)

la Unión Internacional de Telecomunicaciones

para(_6897,3,dv,21,1)

Telecomunicaciones

para(_6897,3,dv,21,1)

la Organización de las Naciones Unidas y la Unión Internacional de Telecomunicaciones , las Naciones Unidas

boca

para(1951,1,av,7,1)

referencia

para(1951,1,av,6,1)

43.20

para(1951,1,av,4,1)

— Sol.Anáf.

una boca artificial

<----->

Oración Correcta Nº 94 -->

\ 43.24 @ trayecto de efecto local @ E : sidetone path F : trajet d " effet local \ Cualquier trayecto acústico , mecánico o eléctrico por el cual un usuario telefónico oye su propia voz y / o el ruido ambiente (en el PRO) .

Anáfora tipo pronRelativo:

pronRelativo el cual

Con paralelismo: para(70484,3,av,22,1)

Lista de antecedentes seleccionados:

Cualquier trayecto acústico , mecánico o eléctrico

para(70484,3,av,20,1)

trayecto de efecto local

para(70484,1,av,6,1)

efecto local

para(70484,1,av,6,1)

43.24

para(70484,1,av,4,1)

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

— Sol.Anáf.

Cualquier trayecto acústico , mecánico o eléctrico

<----->

Oración Correcta Nº 117 -->

2.10 Eco para el oyente Por eco para el oyente se entiende aquella condición de transmisión en la cual la señal vocal principal llega al extremo de la conexión del oyente acompañada por una o más versiones retardadas (ecos) de la señal .

Anáfora tipo pronRelativo:

pronRelativo la cual

Con paralelismo: para(102137,1,dv,8,1)

Lista de antecedentes seleccionados:

aquella condición de transmisión

para(102137,1,dv,6,1)

transmisión

para(102137,1,dv,6,1)

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

— Sol.Anáf.

aquella condición de transmisión

<----->

Oración Correcta Nº 119 -->

Una fuente simple , pero corriente , de eco para el oyente es un trayecto de transmisión a cuatro hilos de baja pérdida que interconecta dos líneas de abonado a dos hilos , conexión en la cual pueden producirse reflexiones como consecuencia de una desadaptación de impedancias en las bobinas híbridas de cada extremo de la sección a cuatro hilos .

Anáfora tipo pronRelativo:

pronRelativo la cual

Con paralelismo: para(_106535,5,dv,16,1)

Lista de antecedentes seleccionados:

conexión

para(_106535,5,dv,14,2)

pérdida

para(_106535,4,dv,11,1)

transmisión a cuatro hilos de baja

para(_106535,4,dv,10,1)

corriente

para(_106535,3,av,6,1)

Una fuente simple

para(_106535,1,av,3,1)

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

— Sol.Anáf.

conexión

<----->

Oración Correcta Nº 129 -->

2.11 Distorsión no lineal Se produce distorsión no lineal en un sistema , en el sentido más general , cuando la salida de éste no guarda una relación lineal con su entrada .

Anáfora tipo pronSP:

éste

Con paralelismo: para(_112785,3,av,13,1)

Lista de antecedentes seleccionados:

el sentido más general

para(_112785,1,dv,10,1)

un sistema

para(_112785,1,dv,10,1)

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

— Sol.Anáf.

el sentido más general

<----->

Oración Correcta Nº 135 -->

De ordinario , la distorsión no lineal es una degradación más importante para la transmisión de datos que para la transmisión de la palabra , pero puede también ser importante en esta última .

Anáfora tipo adj:

esta última

Con paralelismo: *para*(_19196,4,dv,15,1)

Lista de antecedentes seleccionados:

una degradación más importante para la transmisión de datos que para la transmisión de la palabra
para(_19196,2,dv,9,1)

la transmisión de datos que para la transmisión de la palabra
para(_19196,2,dv,9,1)

la transmisión de la palabra
para(_19196,2,dv,9,1)

la palabra
para(_19196,2,dv,9,1)

la distorsión
para(_19196,2,av,5,1)

Antec. y anaf. no co-refieren

— Sol.Anáf.

esta degradación última para la transmisión de datos que para la transmisión de la palabra

<----->

Oración Correcta Nº 144 -->

Para muchos algoritmos de codificación digital , como los utilizados en la MIC de ley A o de ley , que tienen una función de expansión casi logarítmica , es posible reproducir aproximadamente el efecto subjetivo de la distorsión de cuantificación añadiendo un ruido correlacionado con la señal (ruido blanco modulado por la señal vocal) .

Anáfora tipo adj:

los utilizados

Con paralelismo: *para*(_149313,2,av,6,1)

Lista de antecedentes seleccionados:

muchos algoritmos de codificación digital
para(_149313,1,av,3,1)

Antec. y anaf. no co-refieren

— Sol.Anáf.

los algoritmos utilizados de codificación digital

<----->

Oración Correcta Nº 166 -->

2.2.1 Opiniones de los usuarios La opinión del usuario en función de la pérdida de sonoridad puede variar con el grupo de participantes en una prueba y con la estructura de ésta .

— Sol.Anáf.

los resultados obtenidos en pruebas de conversación en laboratorio

<----->

Oración Correcta Nº 199 -->

En las conexiones largas o de longitud media , es probable que la fuente principal de ruido de circuito estribe en los sistemas de transmisión analógica , ya que en ellos la potencia de ruido suele ser proporcional a la longitud del circuito .

Anáfora tipo pronSP:

ellos

Con paralelismo: *para*(_1870,4,av,11,1)

Lista de antecedentes seleccionados:

los sistemas de transmisión analógica

para(_1870,2,dv,8,1)

probable que la fuente principal de ruido de circuito

para(_1870,2,dv,6,0)

ERROR: no halla correctamente el tipo de SN

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

— Sol.Anáf.

los sistemas de transmisión analógica

<----->

Oración Correcta Nº 207 -->

Los límites aplicables a un tono único o a bandas estrechas de ruido deben ser más estrictos que los aplicables al ruido de banda ancha , a fin de evitar molestias al usuario .

Anáfora tipo adj:

los aplicables

Con paralelismo: *para*(_201015,1,dv,6,2)

Lista de antecedentes seleccionados:

Los límites aplicables a un tono único o a bandas estrechas de ruido

para(_1472,1,av,3,1)

ruido

para(_1472,1,av,3,1)

un tono único

para(_1472,1,av,3,1)

— Sol.Anáf.

Los límites aplicables a un tono único o a bandas estrechas de ruido

<----->

Oración Correcta Nº 217 -->

Por lo general , para cualquier tipo de ruido , la satisfacción disminuye en forma monótona a medida que aumenta la potencia de aquél .

Anáfora tipo pronSP:

aquél

Con paralelismo: *para*(_143386,3,dv,10,1)

Lista de antecedentes seleccionados:

cualquier tipo de ruido , la satisfacción

para(_143386,2,av,5,1)

ruido

para(_143386,2,av,5,1)

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

— Sol.Anáf.

cualquier tipo de ruido , la satisfacción

<----->

Oración Correcta Nº 226 -->

En general esto no será siempre posible pues para la mayoría de los aparatos telefónicos que tienen micrófonos lineales y circuitos de conversación , el IELO está estrechamente relacionado con el IEEL y por lo general es de 1, 5 a 4 dB mayor que éste .

Anáfora tipo pronSuj:

esto

Con paralelismo: *para*(_102143,1,av,4,1)

Lista de antecedentes seleccionados:

— OJO: Anáfora no resuelta o exófora

Anáfora tipo pronSuj:

éste

Con paralelismo: *para*(_102143,4,dv,22,2)

Lista de antecedentes seleccionados:

relacionado con el IEEL

para(_102143,3,dv,16,1)

el IEEL

para(_102143,3,dv,16,1)

el IELO

para(_102143,3,dv,13,2)

posible

para(_102143,1,dv,8,1)

— Sol.Anáf.

el IEEL

<----->

Oración Correcta Nº 231 -->

\ Es decir , que el trayecto de efecto local del aparato telefónico es uno de los trayectos a través de los cuales el hablante se oye a sí mismo .

Anáfora tipo pronSuj:

uno

Con paralelismo: *para*(_62137,3,dv,9,1)

Lista de antecedentes seleccionados:

-- OJO: Anáfora no resuelta o exófora

Anáfora tipo *pronRelativo*:

pronRelativo los cuales

Con paralelismo: *para*(_62137,3,dv,12,1)

Lista de antecedentes seleccionados:

los trayectos

para(_62137,3,dv,10,1)

-- Sol.Anáf.

los trayectos

Anáfora tipo *pronSP*:

sí

Con paralelismo: *para*(_62137,3,dv,16,1)

Lista de antecedentes seleccionados:

-- OJO: Anáfora no resuelta o exófora

----->

Oración Correcta Nº 233 -->

La presencia de estos otros trayectos influye en la percepción que tiene el usuario del efecto local , y determina por consiguiente su reacción a éste .

Anáfora tipo *pronSP*:

éste

Con paralelismo: *para*(_206187,4,dv,16,1)

Lista de antecedentes seleccionados:

el usuario de el efecto local

para(_206187,2,dv,11,1)

el efecto local

para(_1091,2,dv,11,1)

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

-- Sol.Anáf.

el usuario de el efecto local

----->

Oración Correcta Nº 240 -->

Por eso existe una gama general de valores preferidos de esta pérdida .

Anáfora tipo *pronSP*:

eso

Con paralelismo: *para*(_63679,1,av,3,1)

Lista de antecedentes seleccionados:

-- OJO: Anáfora no resuelta o exófora

←—————→

Oración Correcta Nº 241 →

Cuando ésta es excesiva , se puede tener la impresión de que el aparato telefónico está muerto cuando uno habla , y en muchas conexiones la ausencia de efecto local no sería la condición preferible .

Anáfora tipo pronSuj:

ésta

Con paralelismo: *para*(_209086,2,av,4,1)

Lista de antecedentes seleccionados:

una gama general de valores preferidos de esta pérdida

para(_63679,1,dv,5,1)

esta pérdida

para(_63679,1,dv,5,1)

No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE

— Sol.Anáf.

una gama general de valores preferidos de esta pérdida

Anáfora tipo pronSuj:

uno

Con paralelismo: *para*(_1698,4,dv,16,2)

Lista de antecedentes seleccionados:

— OJO: Anáfora no resuelta o exófora

←—————→

Oración Correcta Nº 247 →

La atenuación del efecto local viene determinada por el diseño del aparato telefónico y la adaptación de impedancias entre éste y la línea del abonado .

Anáfora tipo pronSP:

éste

Con paralelismo: *para*(_136492,1,dv,5,1)

Lista de antecedentes seleccionados:

el aparato telefónico

para(_136492,1,dv,5,1)

el efecto local

para(_136492,1,av,3,1)

— Sol.Anáf.

el aparato telefónico

Apéndice G: traductor de *SUG* a Prolog

```

%----- Realiza la consulta o reconsulta de Fichero. Leer explicación en "quitarPredicadoMemoria"
sugConsult(Fichero) :- sugConsAux(Fichero,consult).
sugReConsult(Fichero) :- sugConsAux(Fichero,reconsult).

%----- Operadores SUG -----
:-op(1000,xfy,'++>').      % Regla SUG
:-op(600,xfy,':').        % Acceso a la estructura sintáctica
:-op(600,fy,'<<').        % Componente opcional
:-op(600,xf,'>>').        % Componente opcional
:-op(600,fy,'<#').        % Componente alternativo
:-op(600,xf,'#>').        % Componente alternativo
:-op(600,fy,'<##').        % Componente alternativo y opcional
:-op(600,xf,'##>').        % Componente alternativo y opcional

sugConsAux(Fichero,Consult) :-
  see(Fichero),
  repeat,
  read(Regla),
  sugTraduce(Regla,Err,Consult),
  ifthenelse(nonvar(Err),
  seen,
  % sino
  (Regla == end_of_file,
  !, seen)) .

sugConsAux(Fichero,_) :- seen, mensaje2('Consulta fallida del fichero',Fichero).

/*****
  Quita el predicado 'Cab :- Cuer' de memoria en caso que ya existiese
  Si Operacion==consult, significa que se lee la regla por primera vez, por lo que sólo se eliminaría si existiese una
  exactamente igual
  Si no, significa que estamos reconsultando, por lo que se eliminará la primera regla que tenga la misma
  cabecera, independientemente del cuerpo que tenga.
*****/

```

Esto se hace así para evitar que al hacer una modificación del cuerpo de una regla, no se cree en memoria una nueva copia de la regla (y se elimine la anterior)

NOTA: La comprobación de igualdad es siguiendo los criterios de la unificación (=) (A=a, a!=b, A=B,...)

OJO: si en una regla SUG se añade un subpredicado en su cuerpo, hará que se modifique el cuerpo de la regla, por lo que se considerará diferentes (y almacenará otra nueva regla)

sin embargo, si ese subpredicado se añade entre llaves, entonces sí que almacena una sólo regla

```

*****/
quitarPredicadoMemoria(Cab,Cuer,Operacion) :-
  ifthen(Operacion==consult, CuerF=Cuer),
  ifthen(retract((Cab :- CuerF)), true),      % Quita la primera regla de memoria
  assertz((Cab :- Cuer)).                    % Almacenar en memoria

```

%----- Predicado para traducir reglas -----

```
sugTraduce(end_of_file,_).                % Caracter EOF. No se hace nada.
```

```
sugTraduce(Regla, Err, Consult) :-
```

```
  functor(Regla, Functor, _),
```

```
  sugTraduceAux(Functor,Regla, Err, Consult), !.
```

```
sugTraduce(_,_).
```

%----- Reglas SUG -----

```
sugTraduceAux('++>', Regla, Err, Consult) :-
```

```
  arg(1,Regla,Cabeza), arg(2,Regla,Cuerpo),
```

```
  Cuerpo=..L,                                % Lista de SubReglas
```

```
  Cabeza=..C,
```

```
  %--- construccion del 'cuerpo'
```

```
  L=[P1_],
```

```
  ifthenelse(P1==',',
```

```
  cambiar_lista(L,Nuevalista), % varios componentes
```

```
  % sino
```

```
  Nuevalista=[Cuerpo]), % un solo componente
```

```
  lista_subreglas(S,Nuevalista,L0,LF,LRESULT,Err), % resultado en LRESULT
```

```
  %--- construccion de la 'cabeza'
```

```
  ifthenelse(nonvar(Err),
```

```
  mensaje('Revise',Regla),
```

```
  % sino
```

```
(
```

```
  C=[_ARG],
```

```
  noeL(ARG,Num),
```

```
  ifthen( Num>0 ,
```

```
(
```

```
  insercionL(conc,ARG,ARG1),
```

```
  ARGF=..ARG1
```

```
)),
```

```
  C=[NOM_],
```

```
  append([NOM,ARGF,_],S,S1),
```

```
  SyntStr=..S1,
```

```
  append(C,[SyntStr,L0,LF],C3),
```

```
  Cab=..C3,
```

```
%--- tengo una lista y la transformo para dejarla en un solo componente
```

```
%--- que sera el 'cuerpo'
```

```
  cuerpo_lista(LRESULT, Cuer),
```

```
  %--- Preveer situación de tener dos copias de la misma regla
```

```
  quitarPredicadoMemoria(Cab,Cuer,Consult)
```

```

))

/*****
cambiar_lista : Cambia el formato de una lista.
  L: Lista a cambiar y que tiene la forma [' ', pred1, (pred2, pred3)]
  Nuevalista: Lista resultado con la forma [pred1, pred2, pred3]
*****/

cambiar_lista(L, Nuevalista) :-
  L = [_ | R],
  R = [C1, C2],
  ifthenelse(nonvar(C2),
  (
  C2 = ..E1,
  E1 = [P1 | _],
  ifthenelse(P1 == ' ',
  cambiar_lista(E1, NL),
  % sino
  NL = [C2]
  ),
  % sino
  NL = [C2] ),
  insercionL(C1, NL, Nuevalista) .

/*****
lista_subreglas(S1, L, L0, LF, M, Err) : Construye una lista con la
traducción de los predicados que forman parte
del cuerpo de la regla.
S1: Lista, donde se van guardando todos los R1 que forman
parte de la estructura sintáctica.
L: Lista, que contiene el cuerpo de la regla a procesar.
L0: La primera vez es el penultimo componente que hay en
la 'cabeza', las siguientes veces es el ultimo componente
del predicado anterior traducido. Siempre se coloca como
penultimo componente en la traducción de predicados.
LF: Es el ultimo componente que hay en la 'cabeza'. Se coloca
como ultimo componente del ultimo predicado.
M: Lista, en la cual se devuelve todos los predicados
traducidos.
Err: Indicador de error.
*****/

lista_subreglas(S1, [], L0, LF, M, _) :- S1 = [], M = [], L0 = LF .
lista_subreglas(S1, L, L0, LF, M, Err) :-
  L = [EL1 | _],
  functor(EL1, Functor, _),

  %--- traducción de los predicados que engloban los simbolos
  simbolos(Functor, SRESULT, L, MRESULT, F, L0, LU, Err),
  ifthen(var(Err),
  (
  lista_subreglas(S2, F, LU, LF, N, Err),
  %--- Si la lista que contiene los componentes de la Estruct. Sint.
  %--- no esta instanciada significa que se ha hecho una traducción
  %--- de simbolos como '[]', a los cuales no se les inserta argumentos

```

```

ifthenelse(var(SRESULT), % si no esta instanciada
  S1=S2,
% sino
  append(SRESULT,S2,S1)),
  append(MRESULT,N,M)
)).

```

```

/*****
simbolos(Functor,SRESULT,L,MRESULT,F,L0,L1,Err) : Dependiendo
del Functor('<<','>>',etc) entrara en la funcion correspondiente.

```

Procesa toda la traduccion de los predicados que engloben esos simbolos.

Functor: Puede ser cualquiera de los simbolos.

SRESULT: Lista donde se va a ir guardando los componentes R1 que se inserten en los Predicados, y que van a formar parte de la estructura sintactica.

L: Lista que contiene todos los predicados que quedan por procesar.

MRESULT: Lista donde se va guardando los predicados traducidos.

F: Lista que se contiene los predicados que quedan por traducir. Seria la lista L pero sin los predicados que se han traducido en la funcion 'simbolos'.

L0: Componente que se ha de enlazar. Es el ultimo del predicado anterior, y se coloca como penultimo en el siguiente.

L1: En L1, se devuelve el ultimo componete del ultimo predicado traducido, para enlazarlo en los siguientes predicados a traducir.

Err: Indicador de error.

```

*****/
simbolos('<<',SRESULT,L,MRESULT,F,L0,L1,Err) :-

```

/* Comprueba si se trata de <<Regla1>> o <<Regla1,...,ReglaN>>, y tambien el caso de que sea la ultima regla de la lista */

```

L=[E1|R],

```

```

E1=..E,

```

```

E=[_P31],

```

```

ifthenelse(P31==[], % mensaje de error si se da el caso '<<, pred1'

```

```

( Err=si,

```

```

mensaje('Error en',E1)

```

```

),

```

```

% sino

```

```

(

```

```

E=[_P3],

```

```

P3=..E2,

```

```

E2=[EL3|EL5],

```

```

ifthenelse(EL3=':',

```

```

(

```

```

  EL5=[A,C1],

```

```

  C1=..C2,

```

```

  C2=[EL6|C3]

```

```

),

```

```

% sino

```



```

ifthen(EL3='>>', EL6='>>') ,
ifthenelse(EL6='>>',
(
  ifthenelse(EL3=':',
  (
    C3=[PR],
    E3=[EL3,A,PR],
    acceso_es(E3,X,L0,L1,R1)
  ),
  % sino
  (
    EL5=[EL4],
    EL4=..E3,
    insertar_argumentos(E3,R1,L0,L1,X)
  )),
  E4=[X],
  SRESULT=[R1],
  insercionL(opcional,E4,RESULT),
  M1=..RESULT,
  F=R,
  MRESULT=[M1]
),
% caso de <<Pred1,Pred2,...,PredN>>
% sino
(
  %— opcionales trata los predicados a partir de Pred2 a PredN
  opcionales(R,N,F,Err),
  ifthen(var(Err), % si no ha habido error se continua
  (
    %— comprueba si Pred1 lleva :
    ifthenelse(EL3=':',
    acceso_es(E2,X,L0,L1,R1),
    % sino
    insertar_argumentos(E2,R1,L0,L1,X) ),
    SRESULT=[R1],
    insercionL(X,N,E4),
    insercionL(opcional,E4,E5),
    reconstruir_cuerpo(E5,LB),
    LB=[_C22,C33],
    RESULT=[C22,C33],
    M1=..RESULT,
    MRESULT=[M1]
  ))
))
simbolos('{}',_L,MRESULT,F,L0,L1,_):-
  L=[E1|F],
  E1=..E,
  E=[_C2],
  C2=..LN,
  llaves(LN,MRESULT), % la funcion 'llaves' deja los Predicados como esta
  L1=L0.

```

```

simbolos(':',SRESULT,L,MRESULT,F,L0,L1,Err) :-
  L=[E1|F],
  noeL(E1,Num),
  ifthenelse(Num\==1, % solamente puede englobar un predicado
  ( mensaje('Error en', E1),
    Err=si
  ),
  % sino
  (
    E1=[Pred],
    MRESULT=[(L0=[w(Pred,_)|L1])],
    SRESULT=[w(Pred)] % Estructura sintactica
  )).

```

```

simbolos(':',SRESULT,L,MRESULT,F,L0,L1,_) :-
  L=[E1|F],
  E1=..P1,
  acceso_es(P1,M3,L0,L1,R1),
  MRESULT=[M3],
  SRESULT=[R1].

```

```

simbolos('poda',_L,MRESULT,F,L0,L1,Err) :-
  L=[E1|F],
  ifthenelse(F==[],
  ( Err=si,
    mensaje('Error en "poda", necesita un predicado debajo',E1)
  ),
  % sino
  (
    E1=..E,
    E=[P|R],
    insercionL(L0,R,LAUX),
    insercionL(P,LAUX,LAUX1),
    M1=..LAUX1,
    MRESULT=[M1],
    L1=L0
  )).

```

```

simbolos('<#',SRESULT,L,MRESULT,F,L0,LU,Err) :-
  L=[E1|_],
  E1=..E,
  % compruebo si hay error
  E=[P1|ER],
  ifthenelse(ER==[],
  ( Err=si,
    mensaje('Error en: <#')
  ),
  ( E=[P1,P3], % P1 sera <#
  ifthen(nonvar(P3),
  ( P3=..P,
    P=[C1|_]
  )),
  ifthenelse(C1==#>', (Err=si, mensaje('Error en: #>') ),
  (

```

```

ifthenelse(C1==';',
( alternativos(L,MRESULT1,L0,F,L1,L11,_,Err),
ultimoL(LU,L1),
SRESULT=L11
),
%— varios componentes entre parentesis
% sino
( alternativos(L,MRESULT1,SyntStr,L0,LU,F,Err),
SRESULT=[SyntStr] ), % un solo componente
%— reconstruye la lista para que queden separados por ','
reconstruir_alternativos(MRESULT1,MRESULT)
))
)).

simbolos('<##',SRESULT,L,MRESULT,F,L0,LU,Err) :-
L=[E1_],
E1=..E,
E=[P1|ER],
ifthenelse(ER==[],
( Err=si,
mensaje("Error en: <##")
),
( E=[P1,P3], % P1 sera <##
ifthen(nonvar(P3),
( P3=..P,
P=[C1_]
)),
ifthenelse(C1=='##>',(Err=si, mensaje("Error en: ##>")),
(
ifthenelse(C1==';',
( alternativos(L,MRESULT1,L0,F,L1,L11,_,Err), % varios componentes
ultimoL(LU,L1),
SRESULT=L11
),
% sino
( alternativos(L,MRESULT1,SyntStr,L0,LU,F,Err),
SRESULT=[SyntStr] ), % un solo componente
insercionLFin(LU=L0,MRESULT1,MRESULT2),
reconstruir_alternativos(MRESULT2,MRESULT)
))
)).
simbolos(_,SRESULT,L,MRESULT,F,L0,L1,_) :-
simple(L,M1,R1,L0,L1,F),
MRESULT=[M1],
SRESULT=[R1].

```

/******

alternativos(L,M,SyntStr,L0,L1,LN,Err): Traduce el caso de varios componentes separados por comas, pero sin parentesis.

L: Lista que contiene los predicados a partir de los cuales se va a traducir.

M: Lista donde se va a ir guardando la traduccion de predicados.

SyntStr: Estructura Sintactica.

L0: Componente a insertar como penultimo y que es el ultimo del predicado anterior.

L1: Componente a insertar como ultimo en el predicado.

LN: Lista donde devolvemos lo que queda por procesar. Es la misma que L, pero sin los predicados que se han traducido.

Err: Indicador de error.

```
alternativos(L,M,SyntStr,L0,L1,LN,Err) :-
```

```
  alternativos_lista(L,LS,LN,Err),
```

```
  ifthen(var(Err),
```

```
  alternativos_componentes(LS,M,SyntStr,L0,L1,Err) .
```

```
alternativos_lista(L,M,LN,Err) : Construye una lista para los predicados que forman parte de los alternativos.
```

L: Lista que contiene los predicados a traducir. Los alternativos se encuentran en la forma <# pred1, pred2,(<<pred3,pred4>>) #> .

M: Lista que contiene unicamente los predicados que forman parte de los alternativos. El formato es [pred1,pred2,(<<pred3,pred4>>)].

LN: Lista con los predicados que quedan por traducir, una vez extraidos los alternativos.

Err: Indicador de error.

```
alternativos_lista([],_,_,Err) :- Err=si,
```

```
  mensaje('Error: Falta cierre en alternativos') .
```

```
alternativos_lista(L,M,LN,Err) :-
```

```
  L=[C1|C2],
```

```
  ifthenelse(nonvar(C1),
```

```
    ( C1=..LL,
```

```
    LL=[P1|P2],
```

```
    ifthenelse( ( (P1=='<#') ; (P1=='<##') ; P1=='#>' ; (P1=='##>' ) ),
```

```
    ( ifthenelse((P1=='#>' ; P1=='##>' ) ,
```

```
      ( ifthenelse(P2==[], (Err=si, mensaje('Error en',P1) ),
```

```
        ( P2=[P22],
```

```
          ifthenelse(nonvar(P22),
```

```
            ( P22=..PAUX,
```

```
              PAUX=[PP1|PP2],
```

```
              ifthenelse(PP1=='',
```

```
                ( PP2=[PP3_],
```

```
                ifthen(nonvar(PP3),(PP3=..PP4, PP4=[PP5_]) ),
```

```
                ifthenelse(PP5=='<<', ( Err=si, mensaje('Error en',P22)),
```

```
                  cambiar_lista(PAUX,M1) ) % controlar aqui
```

```
                ),
```

```
                % sino
```

```
                  M1=P2
```

```
            ),
```

```
            M1=P2
```

```
          ))
```

```
        ),
```

```
        M1=P2
```

```
      ),
```

```

% sino
( ifthenelse(P1=='',
  ( P2=[A,P3],
    P3=..PL,
    PL=[D1|PR],
    ifthenelse( D1=='#>' ; D1=='##>' ),
    ( PR=[PR1],
      MAUX=..'':A,PR1],
      M1=[MAUX]
    ),
    % sino
      M1=[C1]
    ),
    % sino
    ( C1=..ER,
      ER=[ER1|_],
      ifthenelse(ER1=='', (Err=si, mensaje('Error en',C1) ),
        M1=[C1]
      )) % controlar aqui
    )) .
ifthen(var(Err),
  ( ifthenelse( P1\=='#>', P1\=='##>', D1\=='#>', D1\=='##>' ),
    (
      alternativos_lista(C2,N,LN,Err),
      append(M1,N,M)
    ),
    % sino
    ( M=M1,
      LN=C2
    ))
  ))
),
% sino
(
  alternativos_lista(C2,N,LN,Err),
  insercionL(C1,N,M)
)) .

```

alternativos_componentes(L,N,SyntStr,L0,L1,Err) : Extrae uno a uno los componentes de los alternativos y se hace la traducción .

L: Lista que contiene unicamente los predicados que encierran los alternativos.

N: Lista donde se guarda el resultado de la traducción.

L0: Componente a insertar como penultimo, y que es el ultimo insertado en la traducción del predicado anterior.

L1: Componente a insertar como ultimo.

Err: Indicador de error.

alternativos_componentes([],N,_,_,_) :- N=[] .

alternativos_componentes(L,[M|N],SyntStr,L0,L1,Err) :-

alternativos_argumentos(L,SyntStr,L0,L1,M,F,Err),

ifthen(var(Err),

```
alternativos_componentes(F,N,SyntStr,L0,L1,Err) .
```

```
*****
```

alternativos1(L,M,L0,LN,L1,L11,Err): Es lo mismo que 'alternativos' excepto que se trata del caso de tener varios predicados entre parentesis y separados por comas.

Se hace una llamada a la funcion 'alternativos2', en la cual se trata la traduccion de los predicados que hay entre parentesis.

L: Lista con los predicados que quedan por traducir.

M: Lista que guarda el resultado de la traduccion.

L0: Componente que se inserta como penultimo y es el ultimo del predicado anterior traducido.

LN: Lista que contendra los predicados que queden por traducir una vez traducidos los alternativos.

L1: Es una lista donde se encuentran los argumentos que se han de insertar como ultimo en cada predicado.

L11: Lista con las estructuras sintacticas.

Err: Indicador de error.

```
*****
```

```
alternativos1([],_,_,_,_,Err) :- Err=si,
    mensaje('Error: Falta cierre en alternativos').
```

```
alternativos1(L,M,L0,LN,L1,L11,Ncomp,Err) :-
```

```
    Ncomp1=0,
```

```
    L=[E1|F],
```

```
    E1=..E,
```

```
    E=[P1|P2],
```

```
    ifthenelse((P1='#>'; P1='##>'),
```

```
    ( ifthenelse(P2==[],
```

```
        ( Err=si,
```

```
        mensaje('Error en',P1)
```

```
    ),
```

```
    ( E=[P1,P3],
```

```
    P3=..P,
```

```
    noeL(P,ER),
```

```
    ifthenelse(ER==1,
```

```
    ( Err=si,
```

```
    mensaje('Error. Componente no valida',P3)
```

```
    ),
```

```
    ( cambiar_lista(P,NP),
```

```
      alternativos2(_NP,L0,L1,L11,_,M,Ncomp1,Ncomp,Err),
```

```
      LN=F
```

```
    ))
```

```
    ))
```

```
),
```

```
% sino
```

```
(
```

```
    ifthenelse(((P1='<#'); P1='<##') ,
```

```
    (
```

```
    E=[_P3],
```

```
    P3=..P,
```

```
    P=[ER_] ,
```

```
    ifthenelse((ER=='#>'; ER=='##>'), % no va bien aqui
```

```

( Err=si,
  mensaje('Error. Los alternativos deben englobar como minimo dos predicados')
),
( cambiar_lista(P,NP),
  Primer=si,
  alternativos2(Primer,NP,L0,L1,L11,[],[],M1,_,_,Err),
  noeL(L1,Ncomp)
)
),
% sino
( E1=..P,
  noeL(P,ER),
  ifthenelse(ER==1,
    ( Err=si , P3=E1 ),
    ( cambiar_lista(P,NP),
      alternativos2( NP,L0,L1,L11,_,_,M1,Ncomp1,Ncomp,Err),
      ifthen(nonvar(Err), P3=E1)
    )
  )
),
ifthenelse(var(Err),
( alternativos1(F,N,L0,LN,L1,L11,Ncomp,Err),
  append(M1,N,M)
),
  mensaje('Error. Componente no valida',P3)
)
).

```

alternativos2(Primer,L,L0,L1,L11,L1AUX,L11AUX,M,Ncomp1,Ncomp,Err): Traduce

los predicados que hay entre parentesis, en el caso 'alternativo'.

Primer: Indica si se va a hacer la traduccion del primer componente entre parentesis, ya que si es asi se forma la lista con las estructuras sintacticas y otra con los ultimos componentes.

L: Lista donde se encuentran los predicados entre parentesis

L0: Componente a enlazar.

L1: Lista que contiene los componentes que se han de insertar como ultimos en cada predicado.

L11: Lista que contiene las estructuras sintacticas a insertar en cada componente entre parentesis.

L1AUX: Lista auxiliar para formar la lista L1.

L11AUX: Lista auxiliar para formar la lista L11.

M: Lista donde devolvemos el resultado de la traduccion de los predicados.

Ncomp1: Contador de numero de predicados de cada componente.

Ncomp: Numero de predicados que debe tener cada componente.

Err: Indicador de error.

alternativos2(Primer,[],_,L1,L11,L1AUX,L11AUX,M,Ncomp1,Ncomp,Err) :-

M=[],

ifthenelse(Primer==si,

(L1=L1AUX, L11=L11AUX),

% sino

ifthen((Ncomp1\==Ncomp),

(Err=si,

mensaje('Error. Cada componente de alternativos debe tener el mismo numero de predicados')

```

))
).

alternativos2(Primer,L,L0,L1,L11,L1AUX,L11AUX,M,Ncomp1,Ncomp,Err) :-
  ifthenelse(Primer==si,
    ( insercionLFin(C1,L1AUX,L10),
      insercionLFin(SyntStr,L11AUX,L110)
    ),
    ( Ncomp2 is Ncomp1+1,
      ifthenelse((Ncomp2 > Ncomp),
        ( Err=si,
          mensaje('Error: Numero incorrecto de predicados')
        ),
        ( L1=[C1|L10],
          L11=[SyntStr|L110]
        ))
    )),
  ifthen(var(Err),
    (
      L=[_ _],
      alternativos_argumentos(L, SyntStr,L0,C1,M1,F,Err),
      ifthen(var(Err),
        (
          ifthenelse(Primer==si,
            alternativos2(Primer,F,C1,L1,L11,L10,L110,N,_ _Err),
            % sino
            alternativos2(_ ,F,C1,L10,L110,_ _N,Ncomp2,Ncomp,Err) ),
          insercionL(M1,N,M2),
          noeL(M2,Num),
          ifthenelse(Num>1,
            (
              insercionL(' ',M2,M3),
              RES=..M3,
              M=[(RES)]
            ),
            M=M2 )
          ))
    ))
).

```

```

/*****
alternativos_argumentos(P,SyntStr,L0,L2,M1,F,Err): Realiza la
traduccion de un predicado encerrado entre alternativos.
P: Lista que contiene el predicados tratar. Puede contener varios para
que sea posible controlar el caso de los opcionales [<<pred1,pred2>>].
SyntStr: Estructura sintactica.
L0: Argumento a enlazar.
L2: Argumento a insertar como ultimo en el predicado.
M1: Lista donde se devuelve el predicado traducido.
F: Lista donde devolvemos los predicados que quedan por traducir
en el caso de que sean un componente de alternativos encerrado
entre parentesis, y tambien para controlar el caso de opcionales.
Err: Indicador de error.
*****/

```

```

alternativos_argumentos(P,SyntStr,L0,L2,M1,F,Err) :-

```



```

P=[EL1|R],
ifthenelse(nonvar(EL1),
(
funcion(EL1,Funcion,_),
alternativos_simbolos(Funcion,P,M1,L0,L2,SyntStr,F,Err)
),
% sino
(
M1=(L0=L2),
F=R
)).
)

/*****
alternativos_simbolos(Sim,P,M1,L0,L2,SyntStr,F,Err) : Realiza la traduccion
de un predicado que se encuentra en los alternativos.
Sim : Simbolo del predicado a traducir.
P : Lista que contiene el predicado a traducir.
M1 : Lista donde guardamos el resultado de la traduccion.
L0 : Componente penultimo a enlazar.
L2 : Componente ultimo a insertar.
SyntStr : Estructura Sintactica.
F : Lista con los componentes que quedan por traducir en el
caso de ser un componente de alternativos entre parentesis
Err: Indicador de error
*****/

alternativos_simbolos(':',P,M1,L0,L2,SyntStr,F,_):-
P=[P1|F],
P1=..P11,
acceso_es(P11,M2,L0,L2,R1) ,
M1=(M2,SyntStr=R1) .

alternativos_simbolos('{}',P,M1,L0,L2,_,F,_):-
% un predicado o varios entre llaves
P=[P1|F],
P1=..P11,
P11=[_ ,C2], % C1 es {}
C2=..L,
llaves(L,MAUX) ,
cuerpo_lista(MAUX,MAUX1),
M1=(MAUX1, L0=L2) .

alternativos_simbolos('.',P,M1,L0,L2,SyntStr,F,Err):-
P=[PR|F],
noeL(PR,Num),
ifthenelse(Num>1, (Err=si, mensaje('Error en',PR)),
(
PR=[Pred],
MRESULT=[(L0=[w(Pred,_)|L2])],
MRESULT=[M2],
M1=(M2,SyntStr=w(Pred))
)).

alternativos_simbolos('<<',P,M1,L0,L2,SyntStr,F,Err):-

```

```

alternativo_opcional(P,M1,L0,L2,SyntStr,F,Err) .

alternativos_simbolos(_P,M1,L0,L2,SyntStr,F,_):-
    simple(P,M1,SyntStr,L0,L2,F) .

simple(L,M1,R1,L0,L1,F):-
    L=[E1|F],
    E1=..E,
    insertar_argumentos(E,R1,L0,L1,M1) .

/*****
acceso_es(P1,M3,L0,L1,R1): Hace la traduccion de un predicado 'A:pred'
como un predicado suelto(sin tener en cuenta ',' - que hayan
mas predicados por traducir. Esta funcion se llama desde
simbolos(':', ) y desde simbolos('<<', ).
L: Lista que contiene el predicado a traducir, y se envia
en la forma [':',A,Pred].
M3: Resultado de la traduccion del Predicado, se devuelve
como un solo componente.
L0: Componente a enlazar. Es el ultimo del predicado anterior
traducido, y sera el penultimo del que se va a traducir.
L1: Componente ultimo a insertar en el predicado a traducir.
LU: Devolvemos el ultimo componente que se inserta, ya que
se ha de enlazar con los siguientes.
R1: Componente que es el ultimo argumento que tiene el predicado
sin traducir, y que formara parte de la estructura sintactica.
*****/
acceso_es(P1,M3,L0,L1,R1):-
    P1=[_R1,RR1],
    RR1=..E,
    append(E,[R1,L0,L1],M2),
    M3=..M2 .

/*****
llaves(L,[M|N]): Traduccion del predicado que se encuentra encerrado entre
llaves.
L: Lista que contiene los predicados encerrados entre llaves.
[M|N]: Lista donde guardamos los predicados traducidos.
*****/
llaves(L, [M|N]) :-
    L=[E1|R],
    ifthenelse(E1=';',
    (
        R=[EL1, EL2],
        M=EL1,
        EL2=..F,
        llaves(F, N)
    ),
    % sino
    (
        M=..L,

```

```

N=[]
)).
/*****
alternativo_opcional(L,MRESULT,L0,L1,SyntStr,F) : Realiza la traduccion
de los opcionales dentro de los alternativos.
L : Lista que contiene el predicado a traducir.
MRESULT : Lista donde guardamos el resultado.
L0 : Componente a enlazar.
L1 : Componente a insertar como ultimo.
SyntStr : Estructura Sintactica.
F : Lista con los predicados que quedan por traducir
*****/
alternativo_opcional(L,MRESULT,L0,L1,SyntStr,F,Err) :-
L=[E1|R],
E1=..E,
E=[_,P3],
P3=..E2,
E2=[EL3|EL5],
ifthenelse(EL3=':',
(
EL5=[A,C1],
C1=..C2,
C2=[EL6|C3]
),
% sino
ifthen(EL3='>>', EL6='>>') ,
ifthenelse(EL6='>>',
(
ifthenelse(EL3=':',
(
C3=[PR],
E3=[EL3,A,PR],
acceso_es(E3,X,L0,L1,R1)
),
% sino
(
EL5=[EL4],
EL4=..E3,
insertar_argumentos(E3,SyntStr,L0,L1,X)
)),
E4=[X],
insercionL(opcional,E4,RESULT),
M1=..RESULT,
F=R,
ifthenelse(EL3=':',
MRESULT=(M1, SyntStr=R1),
% sino
MRESULT=M1 )
),
% caso de <<Pred1,Pred2,...,PredN>>
% sino
(
%— opcionales trata los predicados a partir de Pred2 a PredN

```

```

opcionales(R,N,F,Err),
%— comprueba si Pred1 lleva :
ifthenelse(EL3=':',
acceso_es(E2,X,L0,L1,R1),
% sino
insertar_argumentos(E2,SyntStr,L0,L1,X),
insercionL(X,N,E4),
insercionL(opcional,E4,E5),
reconstruir_cuerpo(E5,LB),
LB=[_C22,C33],
RESULT=[C22,C33],
M1=..RESULT,
ifthenelse(EL3==':',
MRESULT=(M1,SyntStr=R1),
% sino
MRESULT=M1)
)).

```

opcionales(L,M,NL) : Funcion que deja todos los predicados, que encierra los opcionales como estaban, excepto el primero.

L : Lista con los predicados que van entre opcionales, excepto el primero, y los demas predicados que quedan por traducir.

M : Lista donde guardamos el resultado.

NL : Lista donde devolvemos los predicados que quedan por traducir.

opcionales([],_,_,Err) :- Err=si, mensaje("Error falta ">>").

opcionales(L,M,NL,Err) :-

```

L=[E1|R],
E1=..E,
E=[P1|P2],
ifthenelse(P1='>>',
( ifthenelse(P2==[],
( Err=si,
mensaje("Error en",E1)
),
( E=[P1,P3],
M=[P3],
NL=R
))
),
% sino es que son predicados intermedios
( ifthenelse(P1==':',
( Err=si,
mensaje("Error en",E1)
),
(
opcionales(R,N,NL,Err),
insercionL(E1,N,M)
))
)).

```

```

/*****
insertar_argumentos(L,R1,L0,L1,RESULT): Inserta en una lista tres
argumentos.
L: Lista donde se va a insertar.
R1,L0,L1: Argumentos a insertar en la lista.
RESULT: Lista donde guardamos el resultado.
*****/
insertar_argumentos(L,R1,L0,L1,RESULT) :-
    append(L,[R1,L0,L1],AUX2),
    RESULT=..AUX2. % le mando una lista y devuelve en forma ()

cuerpo_lista(LRESULT, CUER) :-
    % tengo una lista y la transformo en parentesis
    noel(LRESULT, Num),
    ifthenelse(Num>1,
    (
        reconstruir_cuerpo(LRESULT, CUER1),
        CUER=..CUER1
    ),
    % sino
    LRESULT=[CUER] ).

reconstruir_cuerpo([E1,E2], B) :-
    % L=[E1,E2],
    B1=[E1,E2],
    insercionL(';', B1, B).

reconstruir_cuerpo(L, B) :-
    L=[E1|E2],
    reconstruir_cuerpo(E2, R),
    R1=..R,
    R2=[(R1)],
    insercionL(E1, R2, L1),
    insercionL(';', L1, B).

reconstruir_alternativos([E1,E2], B) :-
    L=[C,D],
    insercionL(';',L,B2),
    R=..B2,
    C=E1,D=E2,
    B=[(R)].

reconstruir_alternativos(L, B) :-
    L=[E1|E2],
    reconstruir_alternativos(E2,R1),
    insercionL(E1,R1,L1),
    insercionL(';',L1,B2),
    R=..B2,
    B=[(R)].

%----- Hechos SUG -----
sugTraduceAux(coordinated, Regla, Err, Consult) :- % Hecho 'coordinated'

```

```

Regla=..Le,
noeL(Le,Num1),
ifthenelse(Num1\==4,
( mensaje('Error en',Regla),
  Err=si
),
% sino
(
  arg(1,Regla,Pred1),
  arg(2,Regla,Pred2),
  arg(3,Regla,Pred3),
  Pred1=..LT1,
  Pred2=..LT2,
  LT1=[EL1|ELR],
  noeL(ELR,Num),
  ifthen( Num>0,
  (
    insercionL(c,ELR,ARG),
    ARGF=..ARG
  )),
  C1=[EL1,ARGF,_,P1,C,P2],
  C2=..C1,
  append(LT1,[C2,L,LF],LT1AUX2),
  Cab=..LT1AUX2,
  LT2=[E1|ER],
  insercionL(E1,ELR,LT2AUX),
  append(LT2AUX,[P1,L,_,],RES1),
  N1=..RES1,
  insercionL(E1,ER,LT2AUX2),
  append(LT2AUX2,[_,_,],RES2),
  N2=..RES2,
  ifthenelse(var(Pred3),
  N3=true,
  % sino
  N3=Pred3 ),
  Cuerpo=coordinable(N1,N2,N3,C,P2,LF),
  %--- Preveer situación de tener dos copias de la misma regla
  quitarPredicadoMemoria(Cab,Cuerpo,Consult)
  )).

sugTraduceAux(basicWord, Regla, Err, Consult) :-      % Hecho 'basicWord'
  Regla=..Le,
  noeL(Le,Num1),
  ifthenelse(Num1\==2,
  ( mensaje('Error en',Regla),
    Err=si
  ),
  % sino
  (
    arg(1,Regla,Cabeza),
    Cabeza=..L,
    L=[NOM|ER],
    noeL(ER,Num),

```

```

ifthen( Num>0,
(
  insercionL(c,ER,ER1),
  ERF=..ER1
)),
S1=[NOM,ERF,_,Palabra],
SyntStr=..S1,
append(L,[SyntStr,L0,LF],LAUX2),
Cab=..LAUX2,
Cuerpo=analizarPalabra(Cabeza,Palabra,L0,LF),
%— Preveer situación de tener dos copias de la misma regla
quitarPredicadoMemoria(Cab,Cuerpo,Consult)
)).

```

sugTraduceAux(juxtaposition,Regla,Err, Consult) :-

```

  Regla=..Le,
  noeL(Le,Num1),
  ifthenelse(Num1\==4,
( mensaje('Error en',Regla),
  Err=si
),
% sino
(
  arg(1,Regla,Pred1),
  arg(2,Regla,Pred2),
  arg(3,Regla,Pred3),
  Pred1=..LT1,
  Pred2=..LT2,
  append(LT1,[B,L,LF],LAUX1),
  Cab=..LAUX1,
  append(LT2,[A1,L,L1],LAUX2),
  S1=..LAUX2,
  append(LT1,[A2,L1,LF],LAUX3),
  P=..LAUX3,
  Cuerpo=(
    S1,
    (ifthenelse( ( call(Pred3),
      P
    ),
      B=[A1|A2],
      (LF=L1, B=[A1])
    ))
  ),
  quitarPredicadoMemoria(Cab,Cuerpo,Consult)
)).

```

sugTraduceAux(fusion,Regla,Err, Consult) :-

```

  Ncomp=1,
  Primer=si,
  obtenerRegla(Primer,Regla,Ncomp,Cabeza,L,Err),
  ifthenelse(var(Err),
( fusion(L,RESULT1,Err),
  ifthenelse(var(Err),
(

```

```

cuerpo_lista(RESULT1,RES),
Regla1=(Cabeza ++> RES),
%   mensaje('Regla1',Regla1),
    sugTraduceAux(++>,Regla1,Err, Consult)
),
mensaje('Revise',Regla)
),
mensaje('Revise',Regla) ).

/*****
obtenerRegla(Regla,Ncomp,Cabeza,L) : Transforma la 'regla fusion'
dejando en L, cada una de las subreglas que la componen.
Regla: Tiene la siguiente forma: fusion( (sn ++> adv),
      (sn ++> sust),
      ).
Ncomp: Numero de componente en la regla.
Cabeza: Cabeza de las subreglas.
L: Lista formada por cada una de las subreglas sin la Cabeza.
Tiene la siguiente forma: [[adv),(sust)] .
*****/

obtenerRegla(Primer,Regla,Ncomp,Cabeza,L,Err) :-
  ifthenelse(arg(Ncomp,Regla,Regla1),
    ( arg(1,Regla1,Cabeza1),
      ifthenelse(nonvar(Primer),
        Cabeza=Cabeza1,
        % sino
        ( ifthen(Cabeza1\==Cabeza,
          ( mensaje("Las subreglas deben tener la misma "cabeza"."),
            Err=si
          ))
        )),
      ifthen(var(Err),
        ( arg(2,Regla1,R1),
          Ncomp1 is Ncomp+1,
          obtenerRegla(_,Regla,Ncomp1,Cabeza,L1,Err),
          insercionL(R1,L1,L)
        ))
      ),
      L=[] ).

/*****
fusion(L,S) : Procedimiento que se repite tantas veces como componentes
tengan las subreglas.
L: Lista con los componentes de cada subregla.
S: Lista con el resultado de tratar los componentes
*****/

fusion(L,S,Err) :-
  unirRamas(L,Pred2,LN2,Err),
  ifthen(var(Err),
    (
      LN2=[P1_|_],
      ifthenelse(nonvar(P1),
        ( fusion(LN2,N,Err),

```



```

    componerFusion(Pred2,S1),
    append(S1,N,S)
),
( componerFusion(Pred2,S)

))

unirRamas(L,Pred2,LN2,Err) :-
    L=[RN2|F],
    ifthenelse(F\==[],
    ( obtenerPred1(RN2,Pred,LN),
    ifthen(var(Err),
    (
        unirRamas(F,Pred1,LN1,Err),
        obtenerUnion(Pred,Pred1,Pred2,Err),
        insercionL(LN,LN1,LN2)
    ))
    ),
    % sino
    ( obtenerPred1(RN2,Pred2,LN1),
    ifthen(var(Err), LN2=[LN1])
    )) .

obtenerPred1(L,Pred,LN) :-
    L=..LIS, LIS=[Com|_],
    ifthenelse(Com==',', arg(1,L,R1),
    (R1=L, Ultimo=si) ),
    R1=..PN,
    PN=[Sim|P1],
    ifthenelse(Sim=='<<',
    ( P1=[P11|_],
    P11=..P12,
    P12=[Sim1|C],
    ifthen(Sim1==',',
    ( C=[A,C1],
    C1=..C2,
    C2=[Sim2|C3],
    ifthen(Sim2=='>>',
    ( C3=[C3AUX],
    CAUX=[A:C3AUX],
    append([opcional],CAUX,Pred),
    ifthen(arg(2,L,LN),true)
    ))
    ))
    ),
    ifthenelse(Sim1=='>>',
    ( append([opcional],C,Pred),
    ifthen(arg(2,L,LN),true)
    ),
    ( ifthen(Sim2\=='>>',
    ( fusionOpcional(L,Predaux,LN),
    insercionL(opcional,Predaux,Pred)
    ))
    ))

```

```

))
),
% sino
(
append([simple],[R1],Pred),
ifthen(arg(2,L,LNAUX),
  ifthen(var(Ultimo), LN=LNAUX)
)) .
fusionOpcional(L,Pred,LN) :-
L=..LIS, LIS=[Com_],
ifthenelse(Com=='', arg(1,L,R1),
  R1=L),
R1=..L1,
L1=[Sim|RES],
ifthenelse(Sim=='>>',
  ( arg(2,L,R2),
fusionOpcional(R2,Pred1,LN),
ifthenelse(Sim=='<<',
  ( RES=[Pred2],
insercionL(Pred2,Pred1,Pred)
),
insercionL(R1,Pred1,Pred) )
),
( Pred=RES,
ifthen(arg(2,L,LN), true)
)) .

obtenerUnion(Pred,Pred1,Pred2,Err):-
ifthenelse(Pred==Pred1, Pred2=Pred,
% sino
( Pred=[I|P],
noeL(P,Numero),
ifthenelse((I==opcional,Numero>1),
  ( mensaje('Error. No pueden haber opcionales que englobe más de un predicado y distintos'),
mensaje('al resto de filas'),
Err=si
),
( Pred1=[I1|P1],
ifthenelse( ((I==simple,I1==simple) ; (I==simple, I1==alternativos)),
  ( ifthenelse(incluidaL(P,P1),
insercionL(alternativos,P1,Pred2),
% sino
( append(P,P1,Paux),
insercionL(alternativos,Paux,Pred2)
))
),
ifthenelse(incluidaL(P,P1),
insercionL(alternativos1,P1,Pred2),
% sino
( append(P,P1,Paux),
insercionL(alternativos1,Paux,Pred2)
))
)
)

```

```

))
)) .

componerFusion(Pred2,S1) :-
  Pred2=[|R|,
  ifthenelse(l==opcional,
  (
    componerPredicado(opcional,R,S1)),
  % sino
  ifthenelse(l==simple, S1=R,
  % sino
  ifthenelse(l==alternativos,
  componerPredicado(alternativos,R,S1),
  % sino
  componerPredicado(alternativos1,R,S1)
  )
  )
  ) .

componerPredicado(Simbolo,R,S1) :-
  noeL(R,Num),
  ifthenelse(Num==1, % es opcional
  ( R=[X],
  X=..X1,
  X1=[X2|X3],
  ifthenelse(X2==':',
  (X3=[A,C], S1=[<<A:C>>]),
  S1=[<<X>>]
  ),
  % sino
  ( R=[P1|RES],
  ifthenelse(Simbolo==opcional, C1=(<<P1),
  ifthenelse(Simbolo==alternativos,
  C1=(<#P1),
  % sino
  C1=(<##P1) )
  ),
  restoPredicados(Simbolo,RES,L1),
  insercionL(C1,L1,S1)
  )) .

restoPredicados(Simbolo,RES,L1) :-
  RES=[C1|F],
  ifthenelse(F==[],
  ( restoPredicados(Simbolo,F,L2),
  insercionL(C1,L2,L1)
  ),
  % sino
  ifthenelse(Simbolo==opcional, L1=[(C1>>)],
  ifthenelse(Simbolo==alternativos,
  L1=[(C1#>)],
  % sino
  L1=[(C1##>)] )
  )

```

```
)  
  
%----- Reglas DCG -----  
sugTraduceAux('->', Regla, _ Consult) :-  
  expand_term(Regla, ReglaProlog),  
  arg(1, ReglaProlog, Cab), arg(2, ReglaProlog, Cuer),  
  %--- Preveer situación de tener dos copias de la misma regla  
  quitarPredicadoMemoria(Cab, Cuer, Consult).  
  
%----- Regla o Hecho Prolog -----  
sugTraduceAux(_, Regla, _ _) :-  
  %--- Preveer situación de tener dos copias de la misma regla  
  ifthen(retract(Regla), true),  
  assertz(Regla).
```

Universitat d'Alacant
Universidad de Alicante

Apéndice H: algoritmo de resolución de la anáfora

```

/*-----
Módulo de resolución de la anáfora.
Toma como entrada la lista de antecedentes de todas las frases anteriores a la actual (LAnt) y la estr.sint. de la
frase actual (ES)
y la devuelve modificada (resuelta la anáfora) en ESF y LAntF
-----*/

anafora(LAnt,LAntF,ES,ESF) :-
    nl,mensaje(' ***** MÓDULO DE RESOLUCIÓN DE LA ANÁFORA *****'),
    % mensaje('ES=',ES),
    % imprimirLAnt('Lista de antecedentes Anteriores a la Oración Actual:',LAnt),
    nl,
    buscarAnaphor(LAnt,LAntF,ES,ESLF),      % Buscar y resolver la anáfora
    ESF =.. ESLF,
    % mensaje('ESF=',ESF),
    nl,mensaje(' ***** FINAL DEL MÓDULO DE RESOLUCIÓN DE LA ANÁFORA *****'),nl,
    anafora(_____) :- mensaje(' *** OJO: ERROR EN MÓDULO DE RESOLUCIÓN DE LA ANÁFORA ***').

/*-----
Busca en ES anaphors y los resuelve.
ENTRADA:
* ES=orac(conc,X, constituyente1, const2, ...) = Es la estr.sint. de la oración actual
* LAnt = lista de antecedentes de las oraciones anteriores =
    [Ant1, Paral1, Ant2, Paral2, ...]
Donde: Ant1 = sn(...)
        Paral1 = paral(Orac,Clausula,AntesDelVerbo,I,II) = Información para resolver el paralelismo sintáctico entre
antecedentes
        Orac = Variable sin instanciar que nos indicará a los antecedentes de esa oración
        Clausula = Nº de clausula dentro de Orac empezando desde 1 y se incrementa por cada conjunción
que se encuentre
        AntesDelVerbo = av = Ant1 está antes del verbo de la "clausula" Clausula
        dv =      está después
        I = Nº de constituyente dentro de la oración Orac
        II = Nº de subconstituyente. Caso de sn coordinados será el nº de sn dentro del principal
* OracActual = Variable sin instanciar que nos indicará a los antecedentes de la oración actual

```

SALIDA:

* una lista de constituyentes (sin anáfora) de la oración actual en ESLF.

* La lista de antecedentes final, añadidos los de la oración actual (LAntF).

-----*/

buscarAnaphor(LAnt,LAntF,ES,ESLF) :-

 functor(ES,Funcionador,Arity),

 buscarAnaphorAux(LAnt,LAntF,ES,[],ESLFT,3,Arity,av,1,_,_),

 componentesConcX(ES,Conc,X),

 insercionL(X,ESLFT,ESLF1), insercionL(Conc,ESLF1,ESLF2),

 insercionL(Funcionador,ESLF2,ESLF). % Funcionador ppal de la oración

%----- Busca linealmente de 1 a Arity todos los argumentos de ES

buscarAnaphorAux(LAnt,LAntF,ES,ESL,ESLF,I,Arity,AntesDelVerbo,Clausula,OracActual,IrsasVerbo) :-

 ifthenelse(I =< Arity,

 (arg(I,ES,ArgI), confIR(IRSAS),

 ifthenelse(nonvar(ArgI),

 (funcionador(ArgI,Namel,_) ,

 ifthenelse((nonvar(IRSAS), var(IrsasVerbo)), % Aquí se calcula la primera vez

 (J is I+1, buscarVerbo(ES, J, Arity, IrsasVerboF)),

 %Sino

 IrsasVerboF = IrsasVerbo

),

 buscarAnaphorAux2(Namel,LAnt,LAntFT,I,1,ArgI,ArgIF,

 AntesDelVerbo,Clausula,OracActual,_,_

 IrsasVerboF) % Busca dentro de ArgI que tiene nombre Namel

),

 %Sino

 (LAntFT = LAnt, IrsasVerboF = IrsasVerbo)

),

 Nuevol is I+1,

 ifthenelse(Namel==conj, % Actualización de datos de comprobación del paralelismo

 (NuevaClausula is Clausula+1,

 NuevoAntesDelVerbo=av,

 ifthen(nonvar(IRSAS),

 (J is I+1, buscarVerbo(ES, J, Arity, IrsasVerboFF))

),

),

 %Sino

 (NuevaClausula=Clausula,

 IrsasVerboFF = IrsasVerboF,

 ifthenelse(Namel==verbo,

 NuevoAntesDelVerbo=dv,

 %Sino

 NuevoAntesDelVerbo=AntesDelVerbo

),

),

),

 insercionLFin(ArgIF,ESL,ESL1),

 buscarAnaphorAux(LAntFT,LAntF,ES,ESL1,ESLF,Nuevol,Arity,

 NuevoAntesDelVerbo,NuevaClausula,OracActual,IrsasVerboFF)

),

 %Sino

 (ESLF = ESL, LAntF = LAnt)

```

).

%----- Busca dentro de un SN. Lo devuelve modificado (sin la anáfora) en SNF
% también devuelve LAnt modificada en NuevoLAnt, incluyendo a SNF
% DentroSN: será variable cuando el constituyente analizado no se encuentra modificando a otro SN
% (esto se controla para las c-command)
buscarAnaphorAux2( sn,LAnt,NuevoLAnt,I,II,SN,SNF,AntesDelVerbo,Clausula,
    OracActual,SNppal,DentroSN,IrsasVerbo) :-
    componentesSN(SN,SNCoordinado),
    buscarAnaphorAux21( sn,LAnt,NuevoLAntT,I,II,SNCoordinado,SNCoordinadoF,
        AntesDelVerbo,Clausula,OracActual,SNppal,DentroSN,
        IrsasVerbo),
    recomponerSN(SN,SNCoordinadoF,SNF),
    ifthenelse(nonvar(SNppal), % Significa que hay que meter en la lista de antecedentes el sn conjunto con
        número plural
        ( insercionL(paral(OracActual,Clausula,AntesDelVerbo,I,0),NuevoLAntT,NuevoLAntT1),
            insercionL(SNF,NuevoLAntT1,NuevoLAnt) ),
    %Sino
        NuevoLAnt = NuevoLAntT
).

%----- Busca dentro de un SNCoordinado
buscarAnaphorAux21( sn,LAnt,NuevoLAnt,I,II,SNCoordinado,SNCoordinadoF,
    AntesDelVerbo,Clausula,OracActual,SNppal,DentroSN,
    IrsasVerbo) :-
    componentesCoord(SNCoordinado,SN1,Conj,SN2),
    componentesSNSinPodar(SN1,SN1Podado),

    tipoSN(SN1Podado,TipoSN1), % Si TipoSN1=sustAdj, Entonces existe la posibilidad de que dentro de
    SN1Podado haya un anaphor
    ifthenelse( ( TipoSN1 \== pronSuj, TipoSN1 \== pronSP, TipoSN1 \== verboInf ),
        ( componentesSNSimple(sustAdj,SN1Podado,_,_,_,SP,AP,_,_),
            ifthenelse(nonvar(SP),
                ( buscarAnaphorAux2( sp,LAnt,LAnt1,I,II,SP,SPF,AntesDelVerbo,
                    Clausula,OracActual,_,dentroSN,IrsasVerbo),
                    duplica(SN1Podado,SPF,SN1Podado1,10)
                ),
            %Sino
                ( SN1Podado1=SN1Podado, LAnt1=LAnt )
            ),
        ifthenelse(nonvar(AP),
            ( componentesAP(AP,C1,SNAP,C2),
                componentesConcX(SNAP,ConcAP,XAP),
                APSNC = snCoordinado(ConcAP,XAP,SNAP,_,_),
                buscarAnaphorAux21( sn,LAnt1,LAnt2,I,II,APSNC,APSNCF,
                    AntesDelVerbo,Clausula,OracActual,_,dentroSN,
                    IrsasVerbo),
                componentesCoord(APSNCF,SNAPF,_,_),
                recomponerAP(AP,C1,SNAPF,C2,APF),
                duplica(SN1Podado1,APF,SN1Podado2,11)
            ),
            %Sino
                ( SN1Podado2=SN1Podado1, LAnt2=LAnt1 )
        )
    )

```

```

)
),
%Sino
( SN1Podado2=SN1Podado, LAnt2=LAnt )
),

resolverAnaphor( TipoSN1,LAnt2,NuevoLAntT2,I,II,SN1Podado2,SN1PodadoF,
  AntesDelVerbo,Clausula,OracActual,DentroSN,IrsasVerbo),
ifthenelse(funcion(SN1PodadoF,sn,_), % Sol.Anáf. es un sn Coordinado
  ( componentesSN(SN1PodadoF, SNCoordinadoF),
    buscaSNConj(SN1PodadoF, Conj, SN2F) % Enlaza con la Conj y SN2F
  ),
  %Sino
  recomponerSNCoordinado(SNCoordinado,SN1PodadoF,Conj,SN2F,SNCoordinadoF) % SNCoordinado sin
  anáforas
),

ifthen( ( var(SNppal),nonvar(SN2), % Sólo se inserta en la LAnt el 1ºSN conjunto (pl,masc,...), II=0
  (TipoSN1==comun ; TipoSN1==propio ; TipoSN1==adj) % Y Si es de uno de estos tipos
  ),
  SNppal=si
),

ifthenelse(nonvar(SN2),
  ( NuevoII is II+1,
    buscarAnaphorAux21( sn,NuevoLAntT2,NuevoLAnt,I,NuevoII,SN2,SN2F,
      AntesDelVerbo,Clausula,OracActual,SNppal,DentroSN,
      IrsasVerbo)
  ),
  %Sino
  NuevoLAnt=NuevoLAntT2
),

%----- Busca dentro de un SP. Lo devuelve modificado (sin la anáfora) en SPF
buscarAnaphorAux2( sp,LAnt,NuevoLAnt,I,II,SP,SPF,AntesDelVerbo,Clausula,
  OracActual,_,DentroSN,IrsasVerbo ) :-
  componentesCoord(SP,SP1,Conj,SP2),
  componentesSPSimple(SP1,_,SN),
  buscarAnaphorAux2( sn,LAnt,NuevoLAntT,I,1,SN,SNF,AntesDelVerbo,Clausula,
    OracActual,_,DentroSN,IrsasVerbo ),
  recomponerSPSimple(SP1,SNF,SP1F), % SP1 sin anáforas
  recomponerSP(SP,SP1F,Conj,SP2F,SPF), % SPF sin anáforas

ifthenelse(nonvar(SP2),
  ( NuevoII is II+1,
    buscarAnaphorAux2( sp,NuevoLAntT,NuevoLAnt,I,NuevoII,SP2,SP2F,
      AntesDelVerbo,Clausula,OracActual,_,DentroSN,
      IrsasVerbo )
  ),
  %Sino
  ( NuevoLAnt = NuevoLAntT ) ).

%----- Busca dentro de una oración coordinada. Lo devuelve modificado (sin la anáfora) en ORCF

```



```

buscarAnaphorAux2( orac,LAnt,NuevoLAnt,I,II,ORC,ORCF,AntesDelVerbo,Clausula,
                  OracActual,_,DentroSN,IrsasVerbo ) :-
    componentesCoord(ORC,OR1,Conj,OR2),
    functor(OR1,FuncOr1,_),
    buscarAnaphorAux2( FuncOr1,LAnt,NuevoLAntT,I,1,OR1,OR1F,AntesDelVerbo,
                      Clausula,OracActual,_,DentroSN,IrsasVerbo ),
    recomponerOrac(ORC,OR1F,Conj,OR2F,ORCF),    % ORF sin anáforas

    ifthenelse(nonvar(OR2),
              ( NuevoII is II+1,
                buscarAnaphorAux2( orac,NuevoLAntT,NuevoLAnt,I,NuevoII,OR2,OR2F,
                                    AntesDelVerbo,Clausula,OracActual,_,DentroSN,
                                    IrsasVerbo )
              ),
              %Sino
              ( NuevoLAnt = NuevoLAntT ) ).

%———— Busca dentro de una oración simple. Lo devuelve modificado (sin la anáfora) en ORF
buscarAnaphorAux2( oracSimple,LAnt,NuevoLAnt,I,_,OR,ORF,AntesDelVerbo,Clausula,
                  OracActual,_,DentroSN,IrsasVerbo ) :-
    componentesOracion(OR,Suj,SV),
    ifthenelse(nonvar(Suj),
              buscarAnaphorAux2( sn,LAnt,LAnt1,I,1,Suj,SujF,av,
                                  Clausula,OracActual,_,DentroSN,IrsasVerbo ),
              %Sino
              LAnt1 = LAnt
    ),
    buscarAnaphorAux2( sv,LAnt1,NuevoLAnt,I,1,SV,SVF,AntesDelVerbo,
                      Clausula,OracActual,_,DentroSN,IrsasVerbo ),
    recomponerOracSimple(OR,SujF,SVF,ORF).

%———— Busca dentro de un SV. Lo devuelve modificado (sin la anáfora) en SVF
buscarAnaphorAux2( sv,LAnt,NuevoLAnt,I,_,SV,SVF,_,Clausula,
                  OracActual,_,DentroSN,IrsasVerbo ) :-
    componentesSV(SV,ADV,PRON,Verbo,CC1,SP1,SN,SP2,CC2),
    ifthenelse(nonvar(PRON),
              buscarAnaphorAux2( pron,LAnt,LAnt1,I,1,PRON,PRONF,av,
                                  Clausula,OracActual,_,DentroSN,IrsasVerbo ),
              %Sino
              LAnt1 = LAnt
    ),
    ifthenelse(nonvar(SP1),
              buscarAnaphorAux2( sp,LAnt1,LAnt2,I,1,SP1,SP1F,dv,
                                  Clausula,OracActual,_,DentroSN,IrsasVerbo ),
              %Sino
              ifthenelse(nonvar(SP2),
                        buscarAnaphorAux2( sp,LAnt1,LAnt2,I,1,SP2,SP2F,dv,
                                            Clausula,OracActual,_,DentroSN,IrsasVerbo ),
                        %Sino
                        LAnt2 = LAnt1    % Solamente está uno de SP1 o SP2
              )
    ),
    ifthenelse(nonvar(SN),

```

```

    buscarAnaphorAux2( sn,LAnt2,LAnt3,I,1,SN,SNF,dv,
                      Clausula,OracActual,_,DentroSN,IrsasVerbo ),
%Sino
    LAnt3 = LAnt2
),
ifthenelse( (nonvar(CC1), tipoCC(CC1,fsp)),
            (componentesCC(CC1,SPCC1,_,_),
              buscarAnaphorAux2( sp,LAnt3,LAnt4,I,1,SPCC1,SPCC1F,dv,
                                Clausula,OracActual,_,DentroSN,IrsasVerbo ),
              recomponerCC(CC1,SPCC1F,_,_,CC1F)
            ),
%Sino
    LAnt4 = LAnt3
),
ifthenelse( (nonvar(CC2), tipoCC(CC2,fsp)),
            (componentesCC(CC2,SPCC2,_,_),
              buscarAnaphorAux2( sp,LAnt4,LAnt5,I,1,SPCC2,SPCC2F,dv,
                                Clausula,OracActual,_,DentroSN,IrsasVerbo ),
              recomponerCC(CC2,SPCC2F,_,_,CC2F)
            ),
%Sino
    LAnt5 = LAnt4
),

NuevoLAnt = LAnt5,

recomponerSV(SV,ADV,PRONF,Verbo,CC1F,SP1F,SNF,SP2F,CC2F,SVF). % SVF sin anáforas

%—— Busca dentro de un PRON personal de complemento. Devuelve su antecedente en PRONF
buscarAnaphorAux2( pron,LAnt,NuevoLAnt,I,II,PRON,PRONF,AntesDelVerbo,Clausula,
                  OracActual,_,DentroSN,IrsasVerbo ) :-
    resolverAnaphor( pronComplemento,LAnt,NuevoLAnt,I,II,PRON,PRONF,AntesDelVerbo,
                   Clausula,OracActual,DentroSN,IrsasVerbo).

%—— Busca dentro de un PRON relativo (el cual). Devuelve su antecedente en PRONF
buscarAnaphorAux2( pronRelativo,LAnt,NuevoLAnt,I,II,PRON,PRONF,AntesDelVerbo,Clausula,
                  OracActual,_,DentroSN,IrsasVerbo ) :-
    resolverAnaphor( pronRelativo,LAnt,NuevoLAnt,I,II,PRON,PRONF,AntesDelVerbo,
                   Clausula,OracActual,DentroSN,IrsasVerbo).

%—— Sgca que es un constituyente que no influye en la resolución de la anáfora o que ha habido un error
buscarAnaphorAux2( LAnt,LAnt,_,_,ArgI,ArgI,_,_,_,_).

/*
Resuelve Anaphor
*/
resolverAnaphor( TipoAnaphor,LAnt,NuevoLAnt,I,II,Anaphor,AnaphorF,
                AntesDelVerbo,Clausula,OracActual,DentroSN,IrsasVerbo ) :-

    (TipoAnaphor==pronComplemento ; % Anaphor pronominal dentro de pron(..., pers,...)
     TipoAnaphor==pronRelativo ; % Anaphor pronominal dentro de pron(..., rel,...) (el cual, la cual, ...)
     TipoAnaphor==pronSuj ; TipoAnaphor==pronSP ; % Anaphor pronominal dentro de SN
     (TipoAnaphor==adj,anaphorADJ(Anaphor)) ; % SN tipo adjetivo

```

```

(TipoAnaphor==comun,anaphorSNDef(Anaphor)) % SN definido tipo nombre común
).

ParalAnaf=para(OracActual,Clausula,AntesDelVerbo,I,II), % Estr.para para el anaphor

write('Anáfora tipo '),write(TipoAnaphor),mensaje(' '),
% confES(EscrES),ifthenelse(var(EscrES),mensaje(Anaphor),
  escrEstr(Anaphor,3) ,
% ),
nl,mensaje('Con paralelismo',ParalAnaf),

restricciones( TipoAnaphor,Anaphor,ParalAnaf,OracActual,LAnt,LAntFiltrado,
  DentroSN), % Selección de posibles antecedentes
noeL(LAntFiltrado,NoeLAntFiltrado), % N°de antec.seleccionados
ifthenelse(NoeLAntFiltrado == 0, % Anáfora no resuelta o catáfora
  ( mensaje2('— OJO: Anáfora no resuelta o exófora'),
  NuevoLAnt=LAnt, AnaphorF=Anaphor ),
%Sino
  ( ifthenelse(NoeLAntFiltrado == 2, % Se resuelve la anáfora. Hay un sólo antec.posible [Ant, para(Ant)]
    LAntFiltrado = [AnaphorFT | _],
  %Sino
    ( preferencias( Anaphor,TipoAnaphor,ParalAnaf,LAntFiltrado,LAntPref,
      lrsasVerbo), % Ordenar por preferencias los antecedentes
      LAntPref = [AnaphorFT | _] ), % Como solución se escoge el primero de ellos
    recomponerAntec(TipoAnaphor, Anaphor, AnaphorFT, AnaphorF), % Comprueba el caso de que no co-
    refiera con el antec., en cuyo caso recompone el antec.
    mensaje('— Sol. Anáf. '),
  % confES(EscrES),ifthenelse(var(EscrES),mensaje(AnaphorF),
    escrEstr(AnaphorF,3) , nl,
  % ),nl,
    insercionL(ParalAnaf,LAnt,NuevoLAnt1), % Sólo si se resuelve se añade su solución a la lista de
    antecedentes
    insercionL(AnaphorF,NuevoLAnt1,NuevoLAnt)
  )
).

%—— Sgca que no es una anáfora, o ha habido un error
resolverAnaphor( _,LAnt,NuevoLAnt,I,II,ArgI,ArgI,AntesDelVerbo,Clausula,
  OracActual,_) :-
  insercionL(para(OracActual,Clausula,AntesDelVerbo,I,II),LAnt,NuevoLAnt1),
  insercionL(ArgI,NuevoLAnt1,NuevoLAnt).

/*
- Comprueba el caso de que no co-refiera con el antec., en cuyo caso recompone el antec.
*/
recomponerAntec(adj, Anaf, Ant, AntF) :-
  datosAnaphor(adj,Anaf,Num,_,_,_),
  ifthenelse( ( \+ igualEstr(Anaf,Ant)
    ; \+ igualValor(Anaf,Ant)
    ; \+ concordanciaSNSimplePodado(Ant,Num,_,_,_)
  ),
  ( mensaje('Antec. y anaf. no co-refieren'),
  recomponerAntecAux(Anaf, Ant, AntF) ),

```

```

%Sino
  AntF=Ant
).

recomponerAntec(TipoAnaphor, __, AntF, AntF) :-
  TipoAnaphor \== adj. % En los demás casos de anáfora no hay que recomponer el antec.

/*-----
- Recompone Ant en Antf, completando lo que no tenga Anaf
*/
recomponerAntecAux(Anaf, Ant, AntF) :-
  componentesConcX(Anaf, ConcAnaf, _),
  componentesSNSimple( sustAdj, Anaf, D1, D2, A1, __, A2, __, SP, AP, GSN, OR, _),
  componentesSNSimple( sustAdj, Ant, __, __, SustAnt, A2Ant, __,
    SPAnt, APAnt, GSNAnt, ORAnt, _),
  ifthenelse(var(A2), A2F=A2Ant, A2F=A2),
  ifthenelse(var(SP), SPF=SPAnt, SPF=SP),
  ifthenelse(var(AP), APF=APAnt, APF=AP),
  ifthenelse(var(GSN), GSNF=GSNAnt, GSNF=GSN),
  ifthenelse(var(OR), ORF=ORAnt, ORF=OR),
  recomponerSNSimplePodado( sustAdj, AntF, D1, D2, __, SustAnt, A2F, A1, SPF,
    APF, GSNF, ORF, ConcAnaf, _).

/*-----
Filtra la Lista de antecedentes (LAnt) en LAntFiltrado, para Anaphor que es
de TipoAnaf con un paralelismo: ParalAnaf.

OracActual es el número de oración en el que se encuentra Anaphor
Según los criterios de: número, género y persona
Se filtrarán el nº de orac. NumOrac, el cual depende del tipo de anáfor:
- Pronombres: NumOrac = 2 = Orac.actual y la previa
- SN definidos: NumOrac = 4 = Orac.actual y las 3 anteriores
*/
restricciones(TipoAnaf, Anaphor, ParalAnaf, OracActual, LAnt, LAntFiltrado, DentroSN) :-
  datosAnaphorAux(TipoAnaf, NumOrac), % Obtener el nº de oraciones a buscar del anaphor
  restriccionesAux( TipoAnaf, Anaphor, ParalAnaf, OracActual, 1, NumOrac, LAnt, [],
    LAntFiltrado, DentroSN),
  imprimirLAnt('Lista de antecedentes seleccionados:', LAntFiltrado).

restriccionesAux( TipoAnaf, Anaphor, ParalAnaf, OracActual, I, NumOrac, LAnt,
  LAntFiltrado, LAntFiltradoF, DentroSN) :-
  ifthen( (LAnt \== [], I =< NumOrac),
    ( LAnt = [Ant1, ParalAnt1 | LAntR],
      ParalAnt1 = paral(NuevaOracActual, __, __, __),
      ifthenelse(NuevaOracActual \== OracActual, % Hemos de cambiar de oración
        Nuevol is I+1,
        %Sino
        Nuevol is I ), % Estamos todavía en la misma orac.
      ifthen(Nuevol =< NumOrac, % Todavía hemos de comprobar los antec.de esta orac.
        ( ifthenelse( cumpleRestr( TipoAnaf, Anaphor, ParalAnaf, Ant1,
          ParalAnt1, DentroSN ),
          ( insercionLFin(Ant1, LAntFiltrado, LAntFiltradoFT1), % Se selecciona como antec.possible
            insercionLFin(ParalAnt1, LAntFiltradoFT1, LAntFiltradoFT1) ),

```

```

    %Sino
    LAntFiltradoFT = LAntFiltrado    % No se selecciona
  ),
  restriccionesAux( TipoAnaf, Anaphor, ParalAnaf, NuevaOracActual,
                  NuevoI, NumOrac, LAntR, LAntFiltradoFT,
                  LAntFiltradoF, DentroSN )
  )
)
),
ifthen( var(LAntFiltradoF),    % Finalización del bucle de oraciones a explorar
       LAntFiltradoF = LAntFiltrado ).

/*-----
- Tiene éxito cuando el antecedente Ant con ParalAnt, cumple las
  restricciones de Anaphor que tiene ParalAnaf y es de tipo TipoAnaf
- Fracasa cuando no
*/
cumpleRestr(TipoAnaf, Anaphor, ParalAnaf, Ant, ParalAnt, DentroSN) :-
  restrMorfologicas(TipoAnaf, Anaphor, Ant),
  restrCCommand(TipoAnaf, ParalAnaf, ParalAnt, DentroSN),
  retrOneAnaf(TipoAnaf, Ant).

/*-----
- Restricciones morfológicas que ha de cumplir un antecedente Ant de un
  Anaphor de tipo TipoAnaf
*/
restrMorfologicas(TipoAnaf, Anaphor, Ant) :-
  ( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ; TipoAnaf == pronRelativo ),
  datosAnaphor(TipoAnaf, Anaphor, Num, Gen, Pers, _),
  concordanciaSNSimplePodado(Ant, Num, Gen, Pers, _). % Si concuerda en Num, Gen y Pers

restrMorfologicas(adj, Anaphor, Ant):-
  datosAnaphor(adj, Anaphor, _ , Gen, Pers, _);
  concordanciaSNSimplePodado(Ant, _ , Gen, Pers, _). % Si concuerda en Gen y Pers

restrMorfologicas(comun, _ , _). % El resto de anafors no tendrán estas restricciones

/*-----
- Restricciones sintácticas del tipo c-command que ha de cumplir un
  antecedente Ant de un Anaphor de tipo TipoAnaf
*/
restrCCommand(TipoAnaf, ParalAnaf, ParalAnt, DentroSN) :-
  ( TipoAnaf == pronSuj ; TipoAnaf == pronSP),
  ParalAnaf = para(OracAnaf, ClausulaAnaf, _ , _),
  ParalAnt = para(OracAnt, ClausulaAnt, _ , _),
  ( nonvar(DentroSN) % Si el anaphor modifica al núcleo de un SN todos los SN de la oración excepto los
    que también modifican a ese núcleo pueden ser antecedentes
  ;
  OracAnaf \== OracAnt % Cumple la restricción si están en oraciones diferentes
  ;
  ( OracAnaf == OracAnt, % O si están en la misma

```

```

ClausulaAnaf \== ClausulaAnt ) % si están en distintas cláusulas
).

restrCCommand(TipoAnaf,_) :- % El resto de anaforas no tendrán estas restricciones
( TipoAnaf == pronComplemento ; % No tiene esta restricción ante el caso: 'A Ana no le gusta ...' en la que 'le'
correfiere con 'Ana'
TipoAnaf == pronRelativo ; % No tiene esta restricción
TipoAnaf == adj ; TipoAnaf == comun ).

/*
- Restricción para la one-anaphora (anáfora tipo adjetivo) que hace que sus antec. sólo pueden ser de tipo comun
*/
retrOneAnaf(adj,Ant) :-
functor(Ant,snSimplePodado,_), % No acepta sn coordinados
tipoSN(Ant, TipoAnt), TipoAnt == comun. % Ha de ser sn tipo comun

retrOneAnaf(TipoAnaf,_) :- TipoAnaf \== adj. % El resto de anaforas no tendrán estas restricciones

/*
- Ordena la Lista de antecedentes (LAnt) en LAntPref, para una anáfora con los datos de paralelismo ParalAnaf
- Posteriormente (a la salida de 'preferencias') se escogerá el primero de LAntPref como antecedente (el más
cercano)
- OJO: El criterio de preferencia del más cercano no hay que ponerlo
*/
preferencias(Anaf,TipoAnaphor,ParalAnaf,LAnt,LAntPref,IrsasVerbo) :-
preferenciasAux(1,Anaf,TipoAnaphor,ParalAnaf,LAnt,LAntPref,IrsasVerbo),
noeL(LAnt, NumLAnt), noeL(LAntPref, NumLAntPref),
ifthen( NumLAnt == NumLAntPref, % Si se devuelve la misma lista de antecedentes
mensaje('No escoge ningún antecedente como preferido: ESCOGE EL MÁS RECIENTE')
).

/*
- Realiza el bucle de criterios de preferencia de distinto nivel:
* Ejecuta preferencia nivel 1.
* Si no queda ningún antec. que lo cumpla, recoge lista antec. anterior
* Si queda sólo un antec. finaliza.
Si no, y todavía quedan niveles de preferencia por ejecutar,
ejecuta sigte nivel de preferencia
*/
preferenciasAux(NivelPref,Anaf,TipoAnaphor,ParalAnaf,LAnt,LAntPref,IrsasVerbo) :-
preferenciasAux2(NivelPref,Anaf,TipoAnaphor,ParalAnaf,LAnt,LAntPref,IrsasVerbo),
ifthenelse(LAntPrefT == [], % No se ha seleccionado ningún antecedente
LAntPrefT1 = LAnt, % Se continua con la lista de antec. anterior
%Sino
LAntPrefT1 = LAntPrefT % Se continua con la lista de antec. filtrada
),
noeL(LAntPrefT1,NumLAntPrefT1),
maximoNivelPref(TipoAnaphor, MaximoNivelPref),
ifthenelse( ( NumLAntPrefT1 > 2, NivelPref < MaximoNivelPref ), % Si se ha seleccionado más de 1 antec. y
todavía quedan preferencias por aplicar, se continua
( NuevoNivelPref is NivelPref+1,
preferenciasAux( NuevoNivelPref,Anaf,TipoAnaphor,ParalAnaf,LAntPrefT1,
LAntPref,IrsasVerbo)
),

```

```

%Sino
  LAntPref=LAntPrefT1
).

preferenciasAux2(NivelPref,Anaf,TipoAnaphor,ParalAnaf,LAnt,LAntPref,IrsasVerbo) :-
  ifthenelse(LAnt \== [],
    ( LAnt=[Ant1, Para1 | LAntR],
      ifthenelse( cumplePref(NivelPref,Anaf,TipoAnaphor,ParalAnaf,Ant1,Para1,IrsasVerbo),
        LAntPref=[Ant1, Para1 | LAntPrefR],
        %Sino
          LAntPref=LAntPrefR
        ),
      preferenciasAux2(NivelPref,Anaf,TipoAnaphor,ParalAnaf,LAntR,LAntPrefR,IrsasVerbo)
    ),
  %Sino
    LAntPref = []
  ).

/*-----
- Por cada tipo de anaphor, debe haber:
  * Un predicado 'maximoNivelPref' que devuelva el número de preferencias existentes
  * Varios predicados 'cumplePref', cada uno de ellos referente a los niveles de preferencias
*/
maximoNivelPref(TipoAnaf, 5) :-
  ( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ).

/*-----
- TipoAnaf = pronombre
- Pref.1: Los antec. que estén en distinta oración
- Pref.2: Los antec. que estén en distinta oración
- Pref.3: Los antec. que tengan la misma posición relativa al verbo
- Pref.4: Los que tengan la misma posición relativa en la oración (mismo I)
- Pref.5: Los SN de tipo nombre común o propio
- Pref.6: Los que tengan la misma posición dentro del constituyente coordinado(mismo II)
*/
cumplePref(1,_,TipoAnaf,ParalAnaf,Ant,_,IrsasVerbo) :-
  ( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ),
  confIR(IRSAS),
  ifthen(nonvar(IRSAS),
    ( ParalAnaf=paral(,_,AVAnaf,_,_),
      componentesSNSimple(sustAdj,Ant,_,_,Sust,_,_,_,_),
      componentesBasicWord(Sust,PalSust),
      wordIrsas(PalSust,_,IrsasSust),
      ifthenelse(AVAnaf == av, arg(1,IrsasVerbo,IrsasPron), arg(2,IrsasVerbo,IrsasPron) ),
      mensaje('IrsasSust',IrsasSust),
      mensaje('IrsasPron',IrsasPron),!,
      compatib(IrsasSust,IrsasPron)
      ,mensaje('compatibles')
    )
  ).

cumplePref(2,_,TipoAnaf,ParalAnaf,_,Para1,_) :-

```

```
( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ),
ParalAnaf=paral(OracAnaf,___),
Paral1=paral(OracAnt1,___),
OracAnaf == OracAnt1.
```

```
cumplePref(3, TipoAnaf,ParalAnaf,Paral1,_) :-
( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ),
ParalAnaf=paral(____,AntesDelVerboAnaf,___),
Paral1=paral(____,AntesDelVerboAnt1,___),
AntesDelVerboAnaf == AntesDelVerboAnt1.
```

```
cumplePref(4, TipoAnaf,ParalAnaf,Paral1,_) :-
( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ),
ParalAnaf=paral(____,IAnaf,___),
Paral1=paral(____,IAnt1,___),
IAnaf == IAnt1.
```

```
cumplePref(5, TipoAnaf, Ant,_) :-
( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ),
mensaje('antes'),
ifthen(tipoSN(Ant, TipoAnt), true),
mensaje('después'),
(var(TipoAnt) ; TipoAnt == comun ; TipoAnt == propio).
```

```
cumplePref(6, TipoAnaf,ParalAnaf,Paral1,_) :-
( TipoAnaf == pronComplemento ; TipoAnaf == pronSuj ; TipoAnaf == pronSP ),
ParalAnaf=paral(____,IIAnaf),
Paral1=paral(____,IIAnt1),
IIAnaf == IIAnt1.
```

```
/*-----
- TipoAnaf = pronombreRelativo
- No hay preferencias se coge el más cercano.
*/
maximoNivelPref(pronRelativo, 0).
cumplePref(1, pronRelativo, ___).
```

```
/*-----
- TipoAnaf = SN adjetivo
- Pref.1: Los antec. que tengan la misma estructura de modificadores
- Pref.2: Los antec. que tengan el mismo valor en esos modificadores
- Pref.3: Los antec. que tengan el mismo número y género
*/
maximoNivelPref(adj, 3).
```

```
cumplePref(1, Anaf, adj, Ant1,_) :-
igualEstr(Anaf, Ant1).
```

```
cumplePref(2, Anaf, adj, Ant1,_) :-
igualValor(Anaf, Ant1).
```

```
cumplePref(3, Anaf, adj, Ant1,_) :-
datosAnaphor(adj, Anaf, Num, ___),
```



```

concordanciaSNSimplePodado(Ant1,Num,_,_). % Si concuerda en Num

/*-----
- TipoAnaf = SN definido
*/
maximoNivelPref(comun, 1).

cumplePref(1,comun,_,_).

/*-----
Deja ES sin los funtores "oracTag, oracTagSuelto" en ESF
ESF=oracTag(conc(),X,Arg1, Arg2, ..., ArgN)
Siendo Arg1 los constituyentes buscados: sn, v, sp, conj
*/
limpia(ES,ESF) :- limpiaAux(ES,[oracTag,'conc',_],ES_L), ESF =.. ES_L.

limpiaAux(ES,ES_Lista,ES_Lista_F) :-
  ifthenelse(nonvar(ES),
    ( functor(ES,Name,Arity),
      ifthenelse( (Name \== oracTag, Name \== oracTagSuelto), % Constituyentes a saltar
        insercionLFIn(ES,ES_Lista,ES_Lista_F), % Constituyente válido
        %Sino % Constituyente NO válido
        limpiaAux2(ES,ES_Lista,ES_Lista_F,3,Arity) % Empieza desde 3 al no considerar "conc, X". Si
        ES=w('palabra'), no pasa nada, se salta por completo
      )
    ),
    %Sino
    ES_Lista_F = ES_Lista
  ).

limpiaAux2(ES,ES_Lista,ES_Lista_F,I,Arity) :-
  ifthenelse(I =< Arity,
    ( arg(I,ES,ArgI),
      limpiaAux(ArgI,ES_Lista,ES_Lista_F1),
      Nuevol is I+1,
      limpiaAux2(ES,ES_Lista_F1,ES_Lista_F,Nuevol,Arity)
    ),
    %Sino
    ES_Lista_F = ES_Lista
  ).

/*-----
Escribe las ES de las oraciones almacenadas en
LES={oracTag1(...), oracTag2(...), ...}
*/
escrLAnaf(LES) :-
  nl,mensaje2(' *** MÓDULO DE IMPRESIÓN DE ORACIONES ANTERIORES A LA ACTUAL ***'),
  escrLAnafAux(LES,1),
  mensaje(' *** FIN DEL MÓDULO DE IMPRESIÓN DE ORACIONES ANTERIORES A LA ACTUAL ***'),
  mensaje2('-----').

escrLAnafAux([],_) :- nl.
escrLAnafAux([H|T],I) :-

```

```

mensaje('Oración Anterior número',I),
confES(EscrES),
ifthenelse(var(EscrES),mensaje(H), escrEstr(H,0) ), tab(1),
Nuevol is I+1, escrLAnafAux(T,Nuevol).

/*
anaforaAux(ES,ESR,ESF) :-
    ES=[ES1 | ESR],
    anaforaRes(ES,ES1,ES1F),
    anaforaAux(ES,ESR,ESF),
    ESF=[ES1F],
    ESF = ES.
*/
/*-----
Módulo de escritura de fórmulas lógicas de la lista de estructuras sintacticas de cada frase (ES)
*/
escrFormulasLogicas(ES) :-
    mensaje(' *** MÓDULO DE IMPRESIÓN DE FÓRMULAS LÓGICAS ***'),
    escrFormulasLogicasAux(ES).

escrFormulasLogicasAux(ES) :-
    ifthen(ES \== [],
        ( ES=[ES1 | ES_R],
          ifthenelse(formulaLogica(ES1,FL),
            ( write(' *** '),write(decl(FL)),nl ),
            %Sino
              mensaje(' *** ERROR: No se puede hallar la fórmula lógica'
            ),
          escrFormulasLogicasAux(ES_R)
        )
    ).

/*-----
Imprime en pantalla una lista de antecedentes de manera tabular
*/
imprimirLAnt(Mensaje, LAnt) :-
    mensaje(Mensaje),
    escrTabL2(LAnt,0).
% escrLAnaf(LAnt).

/*-----
escrTabL2(Lista,Collnicio): escritura de LISTAS en forma tabular,
siendo Collnicio la columna inicial de escritura
-----*/
escrTabL2([H|T],I) :- !,J is I+3, escrTabL2(H,J), escrTabL2(T,0).
escrTabL2(X,I) :- tab(I),
% confES(EscrES),ifthenelse(var(EscrES),mensaje(X),
    escrEstr(X,3) ,
% ),
    nl.
escrTabL2([],_).

```

```

/*
escrTabL2([H|T],I) :- !,J is I+3, escrTabL2(H,J), esc(T,J), nl.
escrTabL2(X,I) :- tab(I), confES(EscrES),
    ifthenelse(var(EscrES),mensaje(X), escrEstr(X,3) ),nl.
*/

/*
Comprueba si en Anaphor hay una anáfora de tipo ADJ:
Será cierto si el adjetivo está precedido por un determinante (D1 o D2)
*/
anaphorADJ(Anaphor) :-
    componentesSNSimple(sustAdj,Anaphor,D1,D2,_,_,_,_,_,_),
    (nonvar(D1) ; nonvar(D2)).

/*
Comprueba si en Anaphor hay una anáfora de tipo SN Definido:
Será cierto si el SN está precedido por un determinante (D1 o D2)
*/
anaphorSNDef(_) :-
    fail. % De momento no la vamos a tratar

/*
** + Devuelve las características de concordancia del anaphor Anaf de tipo
** TipoAnaf=pron ; adj ; sn
** siendo NumOrac el nº de orac. hacia atrás a buscar en función de TipoAnaf
**
datosAnaphor(TipoAnaf,Anaf,Num,Gen,Pers,NumOrac) :-
    arg(1,Anaf,Conc), arg(1,Conc,Num), arg(2,Conc,Gen),arg(3,Conc,Pers),
    datosAnaphorAux(TipoAnaf,NumOrac).

datosAnaphorAux(adj,4).
datosAnaphorAux(pronRelativo,1).
datosAnaphorAux(_,2).

```