

Aportaciones a la Resolución de la elipsis en la coordinación

Manolo Palomar

Depto. Lenguajes y Sistemas Informáticos

Universidad de Alicante

mpalomar@dtic.ua.es

Antonio Ferrández

Depto. Lenguajes y Sistemas Informáticos

Universidad de Alicante

antonio@dtic.ua.es

Lidia Moreno

Depto. Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

lmoreno@dsic.upv.es

Abstract

En [9] estudiamos la formalización de la coordinación basada en la estructura de huecos. En este artículo presentamos un método de tratamiento, recuperación y resolución del fenómeno gramatical de la elipsis que se produce en oraciones coordinadas, donde presentamos una gramática, extensión de las DCG's, que nos permitirá recuperar los elementos elididos y reconstruir el árbol sintáctico con la recuperación de los elementos elididos, lo que nos permitirá obtener la forma lógica en base a un formalismo, extensión del calculo de predicados, definido en [8]. Para la obtención de la forma lógica de las oraciones coordinadas, presentamos un estudio lingüístico de casos de elipsis, con su forma lógica correspondiente. Los mecanismos de recuperación de elementos y de obtención de la forma lógica están desarrollados en Sicstus Prolog.

1 Introducción

Uno de los problemas que tiene la teoría lingüística en general y lingüística computacional en particular es cómo determinar las condiciones bajo las que se puede producir elipsis y los mecanismos de recuperar los elementos elididos. En una oración elíptica siempre ha de quedar alguna marca que nos permita recuperar el elemento elidido y, por tanto, interpretar de manera completa la oración.

Se distinguen tres tipos de elipsis, desde el punto de vista computacional:

1. Intrasentencial.
2. Fragmentos.
3. Elipsis semánticas.

Nos centraremos, en la resolución de la elipsis intrasentencial, es decir que ocurre en una sentencia, para recuperar los elementos elididos durante el análisis sintáctico y poder obtener la forma lógica en el proceso de interpretación.

Consideremos el siguiente ejemplo, de una oración con elipsis intrasentencial,

El ingeniero conectó la impresora y Juan el servidor.

Esta oración contiene una elipsis del núcleo del sintagma verbal, donde en la segunda parte de la oración no aparece el verbo *conectar*.

La resolución de la elipsis consistirá en la recuperación de los elementos elididos, siempre y cuando exista una marca que nos permita recuperar el elemento elidido y cumpla los principios de recuperabilidad de los elementos vacíos [1]

La elipsis tiene un papel central en la coordinación, pues la única forma de explicar la unión de algunos elementos no equivalentes es suponiendo la existencia de uno o más elementos elididos. Muchos fenómenos ligados con la elipsis están originados, desde el punto de vista sintáctico, por la existencia de la coordinación.

Nos centraremos, en la resolución de la elipsis en oraciones coordinadas, y definiremos esta resolución como:

1. Análisis de la oración y detección de la secuencia elíptica.
2. Recuperación de elementos elididos y reconstrucción de su representación.

2 Caracterización de la elipsis.

Es necesario una caracterización lingüística de la elipsis según [6], para posteriormente realizar una caracterización particular para nuestro sistema de resolución.

1. La construcción elíptica implica que hay una categoría funcional sin realización fonética, que ha de tener una posición estructural obligatoria.
2. La construcción con elipsis y la construcción completa deben ser semánticamente idénticas.
3. Los elementos elididos han de ser recuperables, es decir, se han de poder interpretar a partir de la información presente en la propia oración.

2.1 Principios para la recuperación de los elementos vacíos

1. Regla de Recuperabilidad. [12]

- (a) Los elementos vacíos se controlan por elementos paralelos del primer miembro.

Pedro vendrá hoy y Marta mañana.

- (b) Si los elementos del primer miembro fuerán vacíos, entonces se controlarán por elementos externos a la estructura coordinada o por elementos desplazados al extremo derecho de la estructura.

Mañana traerán las camisas: la azul y la roja son para ti.

Pedro compra y Luis vende coches viejos.

2. Principio de paralelismo. Los elementos controlador y controlado (o vacío) ocupan una posición estructural idéntica.

3. Principio de aplicación total. Una regla de recuperabilidad debe aplicarse a todos los miembros de la estructura coordinada con elipsis.

2.2 Casos de Elipsis Intrasentencial.

Podríamos clasificar las oraciones coordinadas entendiendo a diferentes criterios. Con objeto de procesar estas oraciones y resolver la elipsis, distinguiremos algunos casos agrupados según dos criterios:

- Componentes o constituyentes coordinados.
- Componentes o constituyentes elididos.

1. Según los componentes o constituyentes coordinados:

Cuando se coordinan dos componentes o constituyentes de una misma categoría gramatical (verbo, nombre, ...) o la misma categoría funcional en el caso de los sintagmas (nominal, adjetival, ...).

- (a) Oraciones donde aparecen dos verbos coordinados.

María edita e imprime el informe.

En función de la categorización de los verbos, que seán compatibles, se obtendrá la siguiente forma lógica,

$decl(def(X, informe(X), editar(Maria, X) \& imprimir(Maria, X)))$.

- (b) Coordinación de sintagmas nominales

Juan y Pedro van a la catedral.

En este caso, se dará un tratamiento genérico para este tipo de coordinación de constituyentes o componentes de una oración coordinada, donde estarían incluidos muchos de los siguientes casos, planteandose la forma lógica siguiente:

$decl(def(X, catedral(X), ir(juan \& pedro, X)))$.

- (c) Coordinación de objetos indirectos.

Juan se lo pidió a Pedro y a María.

En este caso, obtendríamos la siguiente forma lógica,

$decl(a(pedro \& a(maria, pedir(Juan))))$

- (d) Coordinación de Adjetivos.

Juan es rico y famoso.

Cuya forma lógica,

$decl(rico(juan) \& famoso(juan))$

- (e) Coordinación de preposiciones.

María trabaja por y para ganar dinero.

Cuya forma lógica,

$decl(ex(X, dinero(X), por(X, trabaja(maria)) \& para(ganar(X), trabaja(maria))))$

- (f) Coordinación de adverbios.

Juan camina lenta y cuidadosamente.

Forma lógica,

$decl(lento(camina(juan)) \& cuidado(camina(juan)))$

- (g) Comparativos que no introducen grado de comparación

Pedro y Juan son parecidos.

Este caso se resolvería según la categorización de los verbos y estarían incluidos verbos como *parecerse*, *peinarse*, *lavarse*, y tendrían las siguientes formas lógicas,

decl(parecerse(pedro,juan)).

Otro tipo de frases contempladas en estos casos serían las siguientes:

La gramática categorial y la transformacional divergen en muchos aspectos.

decl(def(X,gramatica(X) & categorial(X),def(Y,gramatica(Y) & transformacional(Y),converger(X,Y))))

- (h) Casos de coordinación de sintagmas nominales en los que se produce una elipsis del presentador de uno de ellos

Los moros y cristianos estan en guerra.

decl(todo(X,moro(X) & todo(Y,cristiano(Y),guerrear(X,Y)))

- (i) Casos de coordinación de sintagmas nominales en los que se produce elipsis del núcleo de uno de ellos. Pero cuando no actúa de núcleo de la frase (podría ser un sintagma preposicional).

La economia esta determinada por las limitaciones físicas y psíquicas de los seres humanos.

decl(def(X,economia(X),todo(Y,ser(Y) & humano(Y),todo(Z,limitacion(Z,Y) & fisica(Z) & limitacion(V,Y) & psiquico(V),ser-pasivo(X,(Z & V))))))

2. Según los componentes o constituyentes elididos:

- (a) Elipsis nominal.

El ingeniero conectó la impresora y cambio el servidor.

En estos casos, si los verbos coordinados tienen la misma subcategorización, entonces se debería unificar el sintagma nominal. Es decir, *conectar(Suj,Obj)* y *cambiar(Suj,Obj)* tienen los mismos huecos obligatorios, y no encontramos en la segunda oración coordinada un sujeto que llene el hueco obligatorio de *cambiar* entonces lo llenamos con el sujeto de la primera oración coordinada. Por tanto, la forma lógica asociada a la oración, podría ser,

decl(def(X,ingeniero(X),def(Y,impresora(Y),conectar(X,Y)) & def(Z,servidor(Z),cambiar(X,Z)).

- (b) Elipsis del núcleo del sintagma verbal.

El mecánico cambio el carburador del coche y Juan la batería.

En este caso de elipsis, si en la segunda oración coordinada los constituyentes que la forman pueden ser elementos potenciales de ser llenados en los huecos del verbo de la primera oración coordinada, se duplicaría la forma lógica para llenar sus huecos,

$decl(def(X, mecanico(X), def(Y, coche(Y), def(Z, carburador(Z, Y), reparar(X, Z))) \# decl(def(V, coche(V), def(W, bateria(W, V), reparar(juan, W)))$

- (c) Elipsis producidas en oraciones comparativas.

Incluimos en este caso, oraciones coordinadas comparativas que introducen grado de comparación:

En estos casos aplicamos el predicado $comp(G, B, X, Y)$, donde G representa el grado de comparación, B la base de la comparación y X e Y los objetos que se compararán. Por ejemplo,

Juan es más alto que Luis y menos que Pepe.

$decl(comp(mas, alto(Juan), Juan, Luis) \& comp(menos, alto(Juan), Juan, Pepe))$

Otro, ejemplo, en este caso pregunta sería:

Cúal es la comunidad con más ríos que valencia y menos que Alicante?

$preg(X, (aut(X) \& comp(mas, num(X1, rio(X1) \& con(X1, X) W), X, valencia)) \& (aut(X) \& comp(menos, num(V, rio(V) \& con(V, X), U, alicante)))$

- (d) Elipsis del sintagma nominal (objeto directo).

Juan vio y María oyo el tren

$decl(ex(tren(X), ver(juan, X))) \# decl(ex(tren(X), oir(maria, X)))$

3 Tratamiento de la elipsis

En [9, 10], presentamos una formalización gramatical de la coordinación con las Gramáticas de Huecos. En este artículo, introducimos una variante en el mecanismo de recuperación de los elementos elididos. Este tratamiento mejora el anterior respecto a la posibilidad de recuperar elementos elididos y obtención de la forma lógica.

3.1 Método de coordinación y recuperación de elementos

Según el estudio de los casos de elipsis intrasentencial, y en base a la obtención de la forma lógica de cada uno de los casos, definimos el siguiente método de recuperación y coordinación de elementos. Si consideramos que la posición de la conjunción en una oración es la X entonces decimos que:

- Si el componente de la oración X-1 y el componente X+1 tienen la misma función sintáctica y morfológica entonces se coordinan considerando la coordinación de elementos, sin necesidad de recuperar ningún elemento elidido para poder obtener la forma lógica.

- Si el constituyente o grupo de constituyentes a la izquierda de la conjunción tienen la misma función o funciones que los de la derecha, entonces se coordinan y los consideraremos coordinación de constituyentes, sin necesidad de obtener la estructura profunda de la frase, para obtener la forma lógica correspondiente.

- Si el componente o constituyente o grupo de constituyentes a la izquierda de la conjunción no tiene/tienen la/s misma/s función/es que el/los de la derecha, entonces no hay coordinación de elementos y es necesario una recuperación de elementos elididos (según el mecanismo que veremos a continuación), con la reconstrucción de la estructura profunda en el análisis sintáctico, para poder posteriormente obtener la forma lógica.

El desarrollo de este último punto se expondrá a continuación.

3.2 Mecanismo de recuperación de elementos elididos

Resolvemos la coordinación de elementos y constituyentes mediante la siguiente estructura:

Ejemplo: ... por Juan, por Pepe y por Antonio ...

```

sp(Numero,Genero,Persona,Tipo,sp(SP1, C, SP2)) -->
  spAux(Numero,Genero,Persona,Tipo,SP1), % Primer SP
  ( ( conj(C), sp(N2,G2,P2,Tipo,SP2), % Segundo SP
    { nonvar(SP2) } % Obligatorio si hay conj haya SP2
  ) % Coordinación de SP
  ;
  { true} % SP Simple %
  ).

spAux(Numero,Genero,Persona,Tipo, sp(P,SN)) -->
  prep(P), % COMPTE.OBLIGATORIO
  sn(Numero,Genero,Persona,ITIPO,sp,SN), nonvar(SN). % COMPTE.OBLIG.

```

Con ello, incluimos flexibilidad para la definición de distintos tipos de componentes gramaticales coordinados que lo resolvemos añadiendo un nuevo predicado -Aux.

```
snAux(..., tipoAdjCal, ...), snAux(..., tipoVerboInfinitivo, ...), ...
```

Y se mantiene almacenada la estructura sintáctica completa de la oración que contiene las características de coordinación de cada elemento gramatical:

```
sv(Numero,Genero,Persona,copul,
   sv(ADV1,PRON1, % Estructura sintáctica del SV
      nucleo(V,conc(Numero,Genero,Persona,TipoVerbo)), % Coord. verbo
      CC1,SP1,SN,SP2,CC2)) % Componentes Opcionales
```

Lo que nos permite las siguientes ventajas:

1. Posibilidad de definir componentes gramaticales opcionales u obligatorios, veamos algún ejemplo:

- (a) Componente que siempre va a ser opcional:

```
sp(INUM,IGEN,IPERS,ITIPO,IESTR,R,R). % No aparece.
```

- (b) Componente obligatorio, que en determinadas ocasiones conviene que sea opcional:

```
..., ( pron(Numero,Genero,IPERS2,pos,comp1,A2) ;
      {true} ), ...
```

- (c) Caso inverso al anterior, en el que un componente opcional conviene que en un determinado instante sea obligatorio:

```
sp(INUM3,IGEN3,IPERS3,ITIPO,S), {nonvar(S)} % S=E.Sintáctica
```

Estos componentes opcionales nos permiten eliminar puntos de backtracking en las situaciones en que hay una estructura gramatical con un número variable de subcomponentes, almacenándose los elementos analizados en la estructura sintáctica de la oración.

Ejemplo: Componentes opcionales del SV (CC, SN, SP, etc)

Además, los huecos que dejen los elementos opcionales podrán ser rellenados en caso de corresponder a elementos elididos.

```
ifthen( ( var(SujOr2), nonvar(SujOr1) ), % Suj(OR2) <-- Suj(OR1)
        SujOr2=SujOr1
      ),
```


2. Incluimos la posibilidad de admitir varios componentes del mismo tipo seguidos, colocándolos en una lista dentro del argumento correspondiente a la estructura sintáctica:

```
adjAux(Numero, Genero, Tipo, adj(B), [A|R1], R2) :-
    es(AS, adj(Numero,Genero,Tipo), A),
    ( ( Tipo=cal, adjAux(Numero,Genero,Tipo,C,R1,R2), B=[AS|C])
      ;
      (R2=R1, B=AS) % Un sólo adjetivo
    ).
```

3. Permitimos establecer restricciones entre distintos componentes gramaticales de la oración, veamos algunos ejemplos:

- (a) El adjetivo ordinal ha de ser precedido por un determinante:

```
det(Numero,Genero,D1), % D1 = estruc. sintáctica
( ( {nonvar(D1)}, adj(Numero,Genero,ord,D2) ) % Adjetivo Ordinal
  ;
  det2(Numero,Genero,D2) % Otro determinante
).
```

- (b) Si es un verbo copulativo se exige que sea acompañado por algún objeto:

```
ifthen (Tipoverbo = copul)
( nonvar(CC1) ; nonvar(SP1) ; nonvar(SN) ;
  nonvar(SP2) ; nonvar(CC2) ).
```

- (c) Se obliga la presencia de un determinante delante del sustantivo cuando hace función de sujeto:

```
..., sust(Numero,Genero,Tipo,S), % COMPTE.OBLIGATORIO
{ ifthen( (Tipo=comun,nonvar(IFUNCION), IFUNCION=suj),
  (nonvar(D1) ; nonvar(D2))
) },
```

4. Resolvemos la elipsis verbal en el predicado 'oracAux2', cuando se llega a una oración no válida precedida por una conjunción:

Por ejemplo:

Juan se parece a Antonio y ana a ella.

Que la resolvería del siguiente modo:

```

ifthenelse( ( L0=[C2|LR],      % Posible elipsis verbal
arg(2,ORT1,SV),
      ((es(C2S,conj,C2),Conj=conj(C2S),LR\=[.])
;
buscaFuncionConj(SV,Pos,Termino1,Conj,Termino2)
)),

```

Además del rellenado del hueco correspondiente al verbo, se rellenan los posibles sujetos y objetos elididos, apoyándonos en la estructura sintáctica almacenada.

5. Resolvemos la elipsis de sujetos y objetos en el predicado 'elipsis' después de la fase de análisis sintáctico, trabajando sobre la estructura sintáctica almacenada. La elipsis se resuelve para cualquier número de oraciones coordinadas.

Ejemplo: Pepe canta muy bien los boleros, Antonio no y él sí.

Presentamos un léxico como una estructura de palabras, cada una de ellas con una sólo entrada, en la que se almacenan todas las posibles funciones gramaticales que puedan desempeñar.

```

esPalabra('la', [art(sing,fem,det), pron(sing,fem,terc,pers,compl)]).

```

De este modo se puede implementar el diccionario como un 'trie' o árbol de letras, lo que permite obtener una complejidad en función de la longitud de las palabras, en lugar de en función del número total de palabras almacenadas. Durante la fase de análisis no se trabaja con una lista de palabras ([este,libro,...]), sino con una lista de estructuras:

```

[palabra(este,[adj(...),pron(...)]), palabra(libro,[sust(...)],...)]

```

Lo que nos permite las siguientes ventajas:

1. Se accede al diccionario una sólo vez por palabra durante toda la fase de análisis, evitándose los accesos repetidos durante los puntos de backtracking.
2. Se detectan los errores de palabras que no estén en el diccionario antes de entrar a analizar la frase (predicado 'frase').
3. Se pueden establecer podas sobre el árbol de backtracking (predicado 'poda') en función del tipo de la siguiente o siguientes palabras a analizar:
Ejemplo: Oraciones sin sujeto, saber por anticipado el tipo de SN de que se trata, etc.

```

sn(Numero, Genero, Persona, Tipo, Funcion, sn(SN1, C, SN2), L,LF) :-
    ifthenelse(poda(L,sn,,),
        fail,          % A continuación no viene un posible SN
        %Sino
        poda(L,tipoSn,TipoSn,FuncSn) % Se sabe el tipo de SN
    ).

```

3.3 Tratamiento semántico

Para la obtención de la forma lógica, aplicamos nuestro analizador semántico de [4], siguiendo las hipótesis de obtención de la forma lógica de [8]:

```
sent(Sem) --> np(X,Scope,Sem), vp(X,Scope).
```

```
np(X,Scope,Sem) --> det(X,Restriction,Scope,Sem), noun(X,Restriction).
np(X,Sem,Sem) --> name(X).
```

```
vp(X,Sem) --> verb0(X,Sem).
```

```
vp(X,Sem) --> verb1(X,Y,S0), pp(Y,S0,Sem).
```

```
vp(X,Sem) --> verb1(X,Y,S0), np(Y,S0,Sem).
```

```
vp(X,Sem) --> verb2(X,Y,Z,S0), np(Y,S0,S1), pp(Z,S1,Sem) .
```

```
pp(X,S0,Sem) --> prep( ), np(X,S0,Sem).
```

```
verb0(X,pasear(X)) --> [pasea].
```

```
verb1(X,Y,editar(X,Y)) --> [edita].
```

```
verb1(X,Y,ir(X,Y)) --> [va].
```

```
verb1(X,Y,querer(X,Y)) --> [quiere].
```

```
verb1(X,Y,cambiar(X,Y)) --> [cambio].
```

```
verb1(X,Y,conectar(X,Y)) --> [conecto].
```

```
det(X,Scope,Restriction,exists(X,Scope,Restriction)) --> [un].
```

```
det(X,Scope,Restriction,exists(X,Scope,Restriction)) --> [una].
```

```
det(X,Scope,Restriction,def(X,Scope,Restriction)) --> [el].
```

```
det(X,Scope,Restriction,def(X,Scope,Restriction)) --> [la].
```

```
det(X,Scope,Restriction,def(X,Scope,Restriction)) --> [los].
```

```
det(X,Scope,Restriction,each(X,Scope,Restriction)) --> [cada].
```

```
noun(X,hombre(X)) --> [hombre].
```

```
noun(X,mujer(X)) --> [mujer].
```

```
noun(X,catedral(X)) --> [catedral].
```

```
noun(X,moro(X)) --> [moros].
```

```
noun(X,guerrear(X)) --> [guerra].
```

```
noun(X,informe(X)) --> [informe].
```

```
noun(X,ingeniero(X)) --> [ingeniero].
noun(X,impresora(X)) --> [impresora].
noun(X,carburador(X)) --> [carburador].
noun(X,coche(X)) --> [coche].
```

```
name(juan) --> [juan].
name(maria) --> [maria].
```

```
prep(en) --> [en].
prep(a) --> [a].
prep(del) --> [del].
```

A partir del cual, y con el mecanismo de recuperación de los elementos elididos, tendremos el analizador completo, desarrollado en Sicstus Prolog.

3.4 Ejemplo

Veamos un ejemplo completo del funcionamiento del analizador en todas sus fases, en el caso de una frase con elipsis verbal, que vimos en secciones anteriores.

Frase: El ingeniero conectó la impresora y Juan el servidor.

En la primera fase el Tokenizador nos la traduce a la siguiente lista:

```
[el,ingeniero,conectó,la,impresora,y,Juan,el,servidor,.]
```

Posteriormente, el analizador sintáctico, recupera el elemento elidido, para construir el árbol sintáctico completo, que será el siguiente, según la salida propia del analizador:

POSIBLE ELIPSIS VERBAL: Tipo 2 (conj.en Termino2)

```
** ORACION:
  ** SINT.NOMINAL:
    ** DETERMINANTE 1:
      ** ARTICULO:
        el
    ** NUCLEO:
      ** SUSTANTIVO:
        ingeniero
  ** SINT.VERBAL:
    ** NUCLEO:
      ** VERBO:
        conectó
    ** SINT.NOMINAL:
      ** DETERMINANTE 1:
        ** ARTICULO:
```

```

    la
  ** NUCLEO:
    ** SUSTANTIVO:
      impresora
** CONJUNCION:
  y
** ORACION:
  ** SINT.NOMINAL:
    ** NUCLEO:
      ** SUSTANTIVO:
        Juan
  ** SINT.VERBAL:
    ** NUCLEO:
      ** VERBO:
        conectó
  ** SINT.NOMINAL:
    ** DETERMINANTE 1:
      ** ARTICULO:
        el
    ** NUCLEO:
      ** SUSTANTIVO:
        servidor

```

Y por último el analizador semántico, obtiene la siguiente forma lógica, correspondiente a la frase inicial según las hipótesis de [8]

```

decl(def(X,ingeniero(X),def(Y,impresora(Y),
conecto(X,Y))) & def(Z,servidor(Z),conecto(Juan,Z)))

```

References

- [1] N. Chomsky
La nueva Sintaxis: Teoría de rección y el Ligamento.
1988.
- [2] M.Covington.
Natural Language Processing for Prolog Programmers.
Prentice Hall. 1994
- [3] J.Barreras
Resolución de Elipsis y Tecnicas de parsing en una Interficie de Lenguaje Natural.
Procesamiento del Lenguaje Natural, nº13, 1993

- [4] V.Dahl,P.Tarau,L.Moreno,M.Palomar
Treating Coordination with Datalog Grammars.
Computational Logic for Natural Language Processing workshop. Edinburgh,
1995.
- [5] V.Dahl,M.McCord
Treating Coordination in Logic Grammars American Journal of Computational
Linguistics, V.9, 1983
- [6] M.Hernanz, J.M. Brucart
La sintaxis 1. Principios teóricos. La oración simple.
barcelona, ed. Crítica. 1988
- [7] R.O'Keefe
The Craft of Prolog.
The MIT Press. 1990
- [8] L.Moreno
Formalismos Lógicos para el Análisis e Interpretación Oracional del Lenguaje
Natural.
Tesis Doctoral. Universidad Politécnica de Valencia. 1993
- [9] M.Palomar,L.Moreno,I.Romero,J.Sebastian
Formalización de la coordinación mediante gramáticas de Huecos.
Procesamiento del Lenguaje natural, nº15, 1994
- [10] M.Palomar,L.Moreno,V.Lopez
Aproximación metagramatical sintáctico semántica de la coordinación.
Procesamiento del lenguaje Natural, nº13, 1993
- [11] E.Ristad
The Anaphora Problem
Department of Computer Science, Princeton University
Academic Press, Inc. 1993
- [12] J.Sola
Estudis de sintaxi del català.
Barcelona, ed. 62, 2 vols. 1972