This is a previous version of the article published by Springer in Foundations of Science, Jan. 2015. doi:http://dx.doi.org/10.1007/s10699-015-9415-1

A computational model of the belief system under the scope of social communication

María Teresa Signes Pont[1], Higinio Mora Mora[1], Gregorio De Miguel Casado[2], David Gil Méndez[1]
[1]Specialized Processor Arquitecture Laboratory
Departamento de Tecnología Informática y Computación
Universidad de Alicante
{teresa, hmora, gil}@ua.es

[2]Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
gmiguel@unizar.es

Abstract
This paper presents an approach to the belief system based on a computational framework in three levels: first, the logic level with the definition of binary local rules, second, the arithmetic level with the definition of recursive functions and finally the behavioural level with the definition of a recursive construction pattern. Social communication is achieved when different beliefs are expressed, modified, propagated and shared through social nets. This approach is useful to mimic the belief system because the defined functions provide different ways to process the same incoming information as well as a means to propagate it. Our model also provides a means to cross different beliefs so, any incoming information can be processed many times by the same or different functions as it occurs is social nets.

Key-words. Binary local rules, recursive functions, behavioural patterns, belief system, social communication

1- Introduction

Beliefs are very important for individuals since they give sense to their actions when available information is incomplete or inconsistent. Beliefs have the capability to impact on our behavior and are a powerful engine to move and change the social environment. Reciprocally, societal changes may trigger belief revision [1, 2]. Belief systems are sets of norms that provide an organized interpretation of the world to the human beings in order to allow a viable interaction human/society [3]. Everybody may have a belief system which is shared totally, partially or not shared with others. Generally, belief systems have no need to be constructed upon reason and survive as long as they provide a satisfactory approach or explanation to events that are poorly understood. Many different disciplines such as mathematics [1,4]], biology [5-8], psychology [9-11], philosophy [12, 13], sociology [14-16], politics [17-20] and more recently computational science [21-29] have supported important advances at different levels of analysis from the molecular/neurological, to the cognitive/psychological and finally to the social/institutional and motivate a great interest for the study of belief systems.

In this paper we consider the belief systems under the scope of the communication of ideas which is crucial in society evolution. Communication between humans allows a wide expansion of ideas grounded in beliefs that configure a time varying map of trends. Nowadays, social nets are responsible of the speedy and ubiquitous information propagation which implies the emergence of trending topics. Our approach deals with the keys of message modification and propagation in society as a case of complex system behaviour. We define a set of binary local rules that apply recursively and trigger functions with complex emergent behaviours. These provide a model for social communication based in beliefs. After introduction, Section 2 develops the computational model in three levels: first, the logic level with the definition of binary local rules, second, the arithmetic level with the definition of recursive functions and finally the behavioural level with the definition of a recursive construction pattern. Section 3 explains why the model is suitable to mimic social communication as manifestation of beliefs and Section 4 summarizes and presents some concluding remarks.

2- The computational framework

2.1 Definition of binary local rules

Equation (1) defines binary local rules ⊗ as follows:

$$\begin{aligned} &\otimes : (0,1) \rightarrow (0,1) \\ &(a,b) \rightarrow c = \otimes(a,b) = a \otimes b \end{aligned} \qquad (1)$$

The rule is particularized by a two input table so, 16= 2^4 different rules (tables) can be defined, see Fig.1: m stands for the index of the table which is the four bit value stored in the cells, m = a_3 a_2 a_1 a_0, i ∈ [0, 3] and a_i ∈(0, 1); m ∈ [0, 2^4-1],

⊗	**0**	**1**
0	a_3	a_1
1	a_2	a_0

Fig. 1. Generic local rule represented by a table.

Without loss of generality we set the row operand is the left one and the column operand is the right one in a one dimensional space. For example: ⊗ (0, 1) = 0 ⊗ 1 = a_1 (0 "acts" on 1).

2.2 Definition of arithmetic functions

The recursive application of rule ⊗ generates a function f_m as shown in Equation (2) where p stands for the sequence length (in bits).

$$\begin{aligned} &\otimes (0,1)^p \rightarrow (0,1)^p \\ &(x_{p-1} ......x_1 x_0) \rightarrow f_m (x_{p-1} ......x_1 x_0) = \\ &[((...(x_0 \otimes x_1) \otimes x_2) \otimes ...) \otimes x_{p-1}]....[((x_0 \otimes x_1) \otimes x_2) x_3][(x_0 \otimes x_1) \otimes x_2](x_0 \otimes x_1) x_0) \end{aligned} \qquad (2)$$

As an example:

m = 7= 0111; f_7 is defined by a table where: a_3 = 0, a_2 = 1, a_1 = 1, a_0 = 1,

so for p = 3, if $(x_2\, x_1\, x_0)$ = (101), we have $f_7\, (x_2\, x_1\, x_0) = f_7$ (101) =111,

for p = 4, if $(x_3\, x_2\, x_1\, x_0)$ = (1101), we have $f_7\, (x_3\, x_2\, x_1\, x_0) = f_7$ (1101) =1111,

etc…

We now analyse the functions f_m when p varies by mapping a set of input sequences with different values of p. The left lattice represents the input sequences (rows). For p=2, we have four possible input sequences of two elements: 00, 01, 10 and 11 (four initial rows). For p=3, we have eight possible input sequences of three elements: 000, 001, 010, 011, 100, 101, 110 and 111 (eight initial rows) and so on (without loss of generality it becomes quite easier to organize all the possible input sequences as if they were decimal values 0, 1, 2, 3 etc…). The right lattice represents the output sequences after applying recursively the rule $\otimes$ on the input sequences. The corresponding input/output pair is on the same row. As an example, *f0* and *f14* are shown in Fig.2. and Fig.3. respectively, for p = 2, 3 and 4. The same can be done for the rest of functions. When p increases one unit, the new input sequences include both the previous input sequences updated with one more bit and the new sequences. In the output sequences the bold underlined characters represent the changes that occur on the new input sequences and the cursive characters are the changes that occur on the updated sequences. In can be observed that for *f0* any input value has the same corresponding output value, irrespectively of it has been updated with one more bit or not. Not so for *f14.*

f0

INPUT				
Decimal value	**Binary value**			
0	0	0	0	**0**
1	0	0	0	**1**
2	0	0	1	**0**
3	0	**0**	1	**1**
4	0	1	0	**0**
5	0	1	0	**1**
6	0	1	1	**0**
7	0	1	1	**1**
8	1	0	0	**0**
9	1	0	0	**1**
10	1	0	1	**0**
11	1	0	1	**1**
12	1	1	0	**0**
13	1	1	0	**1**
14	1	1	1	**0**
15	1	1	1	**1**

OUTPUT						
Binary value				**Decimal value**		
0	*0*	0	**0**	0	0	0
0	*0*	0	**1**	1	1	1
0	*0*	**<u>0</u>**	**0**	0	0	0
0	*0*	**<u>0</u>**	**1**	1	1	1
0	**<u>0</u>**	0	**0**		0	0
0	**<u>0</u>**	0	**1**		1	1
0	**<u>0</u>**	**<u>0</u>**	**0**		0	0
0	**<u>0</u>**	**<u>0</u>**	**1**		1	1
<u>0</u>	0	0	**0**			0
<u>0</u>	0	0	**1**			1
<u>0</u>	0	0	**0**			0
<u>0</u>	0	0	**1**			1
<u>0</u>	**<u>0</u>**	0	**0**			0
<u>0</u>	**<u>0</u>**	0	**1**			1
<u>0</u>	**<u>0</u>**	**<u>0</u>**	**0**			0
<u>0</u>	**<u>0</u>**	**<u>0</u>**	**1**			1

Fig. 2. The function *f0* for p=2, 3 and 4.

f14

INPUT				
Decimal value	**Binary value**			
0	0	0	0	**0**
1	0	0	0	**1**
2	0	0	1	**0**
3	0	**0**	1	**1**
4	0	1	0	**0**
5	0	1	0	**1**
6	0	1	1	**0**
7	0	1	1	**1**
8	1	0	0	**0**
9	1	0	0	**1**
10	1	0	1	**0**
11	1	0	1	**1**
12	1	1	0	**0**
13	1	1	0	**1**
14	1	1	1	**0**
15	1	1	1	**1**

OUTPUT						
Binary value				**Decimal value**		
1	*1*	**1**	**0**	2	6	14
1	*1*	**1**	**1**	3	7	15
1	*1*	1	**0**	2	6	14
1	*1*	**0**	**1**	1	5	13
1	**0**	**1**	**0**		2	10
1	**0**	**1**	**1**		3	11
1	**0**	1	**0**		2	10
1	1	**0**	**1**		5	13
0	**1**	**1**	**0**			6
0	**1**	**1**	**1**			7
0	**1**	1	**0**			6
0	**1**	**0**	**1**			5
1	**0**	**1**	**0**			10
1	**0**	**1**	**1**			11
1	**0**	1	**0**			10
0	1	**0**	**1**			5

Fig. 3. The function *f14* for p=2, 3 and 4.

2.3 Emergence of behavioural patterns

We have run a program in C to analyse the relationship between the output sequences when p increases in order to infer a construction pattern. As we mentioned upper, it becomes easier to consider the input and output sequences as if they were unsigned binary coded numbers. In the table shown in Fig.4, we show the results for the functions *f2, f7, f11* and *f14* as an example. At left we represent the successive output sequences for the input values 0, 1, 2, 3,.., 2^{p-1}, and for different values of p, and at right we note the sequences of M and S (M = *"replication of the previous corresponding output*" and S = *"addition of the value* 2^{p+1} *to the previous corresponding output"*) which code the sequence of elementary actions that are needed to carry out the sequence at p+1 when considering the sequence at p. The first sequence of M, S is the seed. When p increases one unit the output pattern doubles its length and each function reveals a recursive construction pattern which consists of two steps and each step performs one of the following actions: copy, translation, enhancement or equalization, see Table1. Any function is defined by a seed and a construction pattern.

p	p+1	Actions
M	M	copy
S	M	
M	S	translation
S	S	
M	M	enhancement
S	S	
M	S	equalization
S	M	

Table 1..Actions in the construction patterns

$f_m(0, 1, …., 2^p-1)$
(in decimal notation)

f_2

0 1
0 1 2 1
0 1 2 1 4 5 2 5
0 1 2 1 4 5 2 5 8 9 10 9 4 5 10 5
0 1 2 1 4 5 2 5 8 9 10 9 4 5 10 5 16 17 18 17 20 21 18 21 8 9 10 9 20 21 10 21
………………………………………………………………………………

f_7

0 1
0 3 2 3
0 7 6 7 4 7 6 7
0 15 14 15 12 15 14 15 8 15 14 15 12 15 14 15
0 31 30 31 28 31 30 31 24 31 30 31 28 31 30 31 16 31 30 31 28 31 30 31 24 31 30 31 28 31 30 31
0 63 62 63 60 63 62 63 56 63 62 63 60 63 62 63 48 63 62 63 60 63 62 63 56 63 62 63 60 63 62 63 32 63 62 63 60 63 62 63 56 63 62 63 60 63 62 63 48 63 62 63 60 63 62 63 56 63 62 63 60 63 62 63
……………………………………………………………………

f_{11}

0 1
2 1 2 3
2 5 2 3 6 5 6 7
10 5 10 11 6 5 6 7 10 13 10 11 14 13 14 15
10 21 10 11 22 21 22 23 10 13 10 11 14 13 14 15 26 21 26 27 22 21 22 23 26 29 26 27 30 29 30 31
42 21 42 43 22 21 22 23 42 45 42 43 46 45 46 47 26 21 26 27 22 21 22 23 26 29 26 27 30 29 30 31 42 53 42 43 54 53 54 55 42 45 42 43 46 45 46 47 58 53 58 59 54 53 54 55 58 61 58 59 62 61 62 63
………………………………………………………………………………………

Pattern construction

f_2

p=1
p=2 MMSM (seed)
p=3 MMMMSSMS
p=4 MMMMMMMMSSSSMMSM
p=5 ………………………………
.
. ………………………………
.
. **[copy, equalization]**

f_7

p=1
p=2 MSSS (seed)
p=3 MSSSSSSS
p=4 MSSSSSSSSSSSSSSS
p=5 ……………………………
………………..
p=6 **[enhancement, translation]**
.
.
.

f_{11}

p=1
p=2 SMSS (seed)
p=3 MSMMSSSS
p=4 SMSSMMMMSSSSSSSS
p=5 ………………………………
…………………………………
p=6 **[equalization, translation]**
.
.
.

f14		*f14*
0 1	p=1	
2 3 2 1	p=2	SSSM (seed)
6 7 6 5 2 3 2 5	p=3	SSSSMMMS
14 15 14 13 10 11 10 13 6 7 6 5 10 11 10 5	p=4	SSSSSSSSMMMMSSSM
30 31 30 29 26 27 26 29 22 23 22 21 26 27 26 21 14 15 14 13 10 11 10 13 22 23 22 21 10 11 10 21	p=5	....
62 63 62 61 58 59 58 61 54 55 54 53 58 59 58 53 46 47 46 45 42 43 42 45 54 55 54 53 42 43 42 53 30 31 30 29 26 27 26 29 22 23 22 21 26 27 26 21 46 47 46 45 42 43 42 45 22 23 22 21 42 43 42 21	p=6 . . .	**[translation, equalization]**
....		

Fig. 4. Recursive construction patterns of the functions *f2*, *f7*, *f11*and *f14*.

Table 2. shows the recursive construction patterns for all the studied functions. The first action is denoted 1 and the second, 2. The order of the actions is crucial to the overall result of a function.

Function	**Actions**			
Seed	**Copy**	**Translation**	**Enhancement**	**Equalization**
f0 MMMM	1,2			
f1 MMMS	1		2	
f2 MMSM	1			2
f3 MMSS	1	2		
f4 MSMM	2		1	
f5 MSMS			1,2	
f6 MSSM			1	2
f7 MSSS		2	1	
f8 SMMM	2			1
f9 SMMS			2	1
f10 SMSM				1,2
f11 SMSS		2		1
f12 SSMM	2	1		
f13 SSMS		1	2	
f14 SSSM		1		2
f15 SSSS		1, 2		

Table 2. Recursive construction patterns for the functions

3- An approach to social communication model

In our model the functions are suitable models of beliefs in the sense they have the capability to define different ways to process the incoming information. These functions outline a two step recursive construction pattern which allow extend the processed information release as it occurs in propagation. The processing consists on applying series of modifications (replication and/or additions) to the incoming information, and the structure of the series characterizes the function. The beliefs are defined by the seed and the recursive construction pattern. Social communication is achieved when different beliefs are expressed, modified, propagated and shared through social nets. The recursive formula allows changing the ongoing function at a step and going on with another function at the following step, so, any incoming information can be processed many times by the same or different functions, as it occurs is social nets.

4- Conclusions

In this paper, we approach the belief system through a set of functions that have the capability to define different ways to process incoming information. The functions come from the recursive application of very simple local rules and trigger more complex emerging behaviours formalized by a recursive pattern construction. This approach is useful to mimic the social communication because it provides a model for modification and propagation of the information as well as a means to cross different beliefs through the change of the ongoing function in the processing system. As a future work we plan to improve the model by application and validation in realistic scenarios.

References

[1] Dixon, E. (1994). Belief Revision: A Computational Approach

[2] Luam, S. (2005). A computational framework for belief change. Springer-Verlag Berlin

[3] Nescolarde-Selva, J. A systemic vision of belief systems and ideologies. Ph.D. *Department ofApplied Mathematics. University of Alicante. Alicante. Spain.*

[4] Schelling, T. C. (1971). Dynamic models of segregation. *The Journal of Mathematical Sociology, 1*, 143-186.

[5] Alford, J. R., Funk, C. L., & Hibbing, J. R. (2005). Are political orientations genetically transmitted? *American Political Science Review, 99*, 153-167

[6] Amodio, D. M., Jost, J., Master, S. L., & Yee, C. M. (2007). Neurocognitive correlates of liberalism and conservatism. *Nature Neuroscience, 10*, 1246-1247. doi:10.1038/nn1979 Journal of Social and Political Psychology 2013, Vol. 1(1), 337–363 doi:10.5964/jspp.v1i1.36 Ideologies as Complex Systems 356

[7] Duncan, S., & Barrett, L. F. (2007). Affect is a form of cognition: A neurobiological analysis. *Cognition and Emotion, 21*, 1184-1211. doi:10.1080/02699930701437931

[8] Eliasmith, C. (2013). *How to build a brain: A neural architecture for biological cognition.* New York, NY: Oxford University Press.

[9] Findlay, S. D., & Thagard, P. (2011). Emotional change in international negotiation: Analyzing the Camp David accords using cognitive-affective maps. *Group Decision and Negotiation*. Advance online publication. doi:10.1007/s10726-011-9242-x

[10] Fontaine, J. R. J., Scherer, K. R., Roesch, E. B., & Ellsworth, P. C. (2007). The world of emotions is not two-dimensional. *Psychological Science, 18*, 1050-1057. doi:10.1111/j.1467-9280.2007.02024.x Journal of Social and Political Psychology 2013, Vol. 1(1), 337–363 doi:10.5964/jspp.v1i1.36 Homer-Dixon, Maynard, Mildenberger et al. 357

[11] Freeden, M. (2010). Social and psychological bases of ideology and system justification – Edited by John T. Jost, Aaron C.

[12] Gell-Mann, M. (1997). The simple and the complex. In D. Alberts & T. Czerwinski (Eds.), *Complexity, global politics, and national security* (pp. 3-28). Washington, DC: National Defense University.

[13] Haidt, J., Graham, J., & Joseph, C. (2009). Above and below left-right: Ideological narratives and moral foundations. *Psychological Inquiry, 20*, 110-119. doi:10.1080/10478400903028573

[14] Gilbert, N., & Troitzsch, K. (2005). *Simulation for the social scientist*. Berkshire, United Kingdom: Open University Press.

[15] Hagan, J., & Rymond-Richmond, W. (2008). The collective dynamics of racial dehumanization and genocidal victimization in Darfur. American Sociological Review, 73, 875-902. doi:10.1177/000312240807300601

[16] Hardin, C. D., & Higgins, E. T. (1996). Shared reality: How social verification makes the subjective objective. In E. T. Higgins & R. M. Sorrentino (Eds.), *Handbook of motivation and cognition: The interpersonal context* (Vol. 3, pp. 28-84). New York, NY: Guilford.

[17] Howarth, D., Norval, A., & Stavrakakis, Y. (2000). *Discourse theory and political analysis: Identities, hegemonies and social change.* Manchester, United Kingdom: Manchester University Press.

[18] Jost, J. T., Glaser, J., Kruglanski, A. W., & Sulloway, F. J. (2003). Political conservatism as motivated social cognition. *Psychological Bulletin, 129*, 339-375. doi:10.1037/0033-2909.129.3.339

[19] Leader Maynard, J. (2013). A map of the field of ideological analysis. *Journal of Political Ideologies, 18*, 299-327. doi:10.1080/13569317.2013.831589.

[20] Liebes, T., Katz, E., & Ribak, R. (1991). Ideological reproduction. *Political Behavior, 13*, 237-252. doi:10.1007/BF00992921.

[21] Lang, J., & De Sterck, H. (2012). *The Arab Spring: A simple compartmental model for the dynamics of a revolution.*arXiv:1210.1841v1.

[22] Helbing, D., & Balietti, S. (2012). Agent-based modeling. In D. Helbing (Ed.), *Social self-organization* (pp. 25-70). New York, NY: Springer. Journal of Social and Political Psychology 2013, Vol. 1(1), 337–363 doi:10.5964/jspp.v1i1.36 Ideologies as Complex Systems 358.

[23] Helbing, D., & Carbone, A. (Eds.). (2012). Participatory science and computing for our complex world. *The European Physical Journal: Special Topics, 214*(1). doi:10.1140/epjst/e2012-01686-y

[24] Helbing, D., & Kirman, A. (2013). *Rethinking economics using complexity theory.* Retrieved from http://www.soms.ethz.ch/paper_economics_complexity_theory

[25] Sobkowicz, P. (2009). Modelling opinion formation with physics tools: Call for closer link with reality. *Journal of Artificial Societies and Social Simulation, 12*(1), Article 11. Retrieved from http://jasss.soc.surrey.ac.uk/12/1/11.html

[26] Squazzoni, F. (2012). *Agent-based computational sociology*. Chichester, United Kingdom: Wiley. Journal of Social and Political Psychology 2013, Vol. 1(1), 337–363 doi:10.5964/jspp.v1i1.36. Homer-Dixon, Maynard, Mildenberger et al. 361.

[27] Wolf, I., Neumann, J., Schröder, T., & de Haan, G. (2012). The adoption of electric vehicles: An empirical agent-based model of attitude formation and change. In A. Koch (Ed.), *Proceedings of the 8th Conference of the European Association for Social Simulation* (pp. 93-98). Salzburg, Austria: Universität Salzburg.

[28] Bello P. Cognitive Foundations for a Computational Theory of Mindreading. Advances in Cognitive Systems 1 (2012) 59–72.

[29] Homer-Dixon, T et al.. A Complex Systems Approach to the Study of Ideology: Cognitive-Affective Structures and the Dynamics of Belief Systems. Journal of Social and Political Psychology 2013, Vol. 1(1), 337–363.