

An Agent-Based Model Studying the Acquisition of a Language System of Logical Constructions

Josefina Sierra-Santibáñez

Technical University of Catalonia

Jordi Girona Sangado, 1–3, 08034 Barcelona, Spain

E-mail: Maria.Josefina.Sierra@upc.edu

Abstract

This paper presents an agent-based model that studies the emergence and evolution of a language system of logical constructions, i.e. a vocabulary and a set of grammatical constructions that allows the expression of logical combinations of categories. The model assumes the agents have a common vocabulary for basic categories, the ability to construct logical combinations of categories using Boolean functions, and some general purpose cognitive capacities for invention, adoption, induction and adaptation. But it does not assume the agents have a vocabulary for Boolean functions nor grammatical constructions for expressing such logical combinations of categories through language. The results of the experiments we have performed show that a language system of logical constructions emerges as a result of a process of self-organisation of the individual agents' interactions when these agents adapt their preferences for vocabulary and grammatical constructions to those they observe are used more often by the rest of the population, and that such a language system is transmitted from one generation to the next.

Content Areas: Cognitive Modeling, Symbolic AI, Simulating Humans, Adaptive Behavior

1 Introduction

Agent-based models, implemented and tested in computer simulations, are one of the approaches that have been used with increasing success to study issues in the origins and evolution of language in the last two decades (Hurford, Studdert-Kennedy, and Kight 1998; Briscoe 2002; Lyon, Nehaniv, and Cangelosi 2007). Depending on whether the designers of these models emphasise the role of biological evolution (e.g. genetic evolution by natural selection) or the role of cultural evolution, agent-based models can be classified into two areas: *Biolinguistics*, which hypothesises that the structure of language is determined to a large extent by biological factors (Di Sciullo and Boeckx 2011); and *Evolutionary Linguistics*, which hypothesises that language is primarily shaped by cultural forces (Minett and Wang 2005; Croft 2008; Steels 2011b).

Agent-based models exploring the evolutionary linguistics approach usually involve a population of artificial agents which can be either software agents operating in a virtual

world (Steels 1995; Smith, Brighton, and Kirby 2003) or physical robots interacting with each other in a real world as experienced by their sensory-motor system (Steels 1998). The agents interact with each other playing language games. A *language game* (Wittgenstein 1953) is played by two agents, a speaker and a hearer. The speaker has a specific communicative goal, conceptualises the world for language, and transforms this conceptualisation into an utterance. The hearer tries to parse the utterance, reconstruct its meaning and map it into its own perceptual experience of the world. Depending on the outcome of the game speaker and hearer use different strategies to expand and adapt their internal languages to be more successful in future language games.

In these experiments, the agents are initially endowed with a set of cognitive abilities that are hypothesised to be necessary for seeing the emergence of language strategies to be successful in a language game (Steels 2012). Then, they are made to play a series of language games, where they configure possible strategies and try them out. The goal of the experiments is to find out whether the population as a whole succeeds in the language game, i.e. communicates effectively, and to observe the conceptualisations and linguistic constructions that emerge in the population as a result of the processes of collective invention, negotiation and language transmission across generations.

Theories of language evolution study language change at two different levels: that of language systems and that of language strategies. *Language systems* (also called *paradigms*) capture the regularity observed in some part of the vocabulary or grammar of a language, for example, a system of colour terms, tense-aspect distinctions, cases or determiners. Language systems group a set of paradigmatic choices both on the side of meaning (the conceptual system) and on the side of form (the linguistic system). The *conceptual system* includes semantic distinctions that are expressible in a language system and can therefore be used as building blocks for conceptualisation. The *linguistic system* includes the syntactic and morphological categories and grammatical constructions necessary to turn a conceptualisation into a concrete utterance. A given language comprises thousands of language systems, which are closely integrated.

Linguists call the approach underlying a language system a *language strategy*. They talk about relative-clause formation strategies, coordination strategies (for combining

nouns) or case strategies. Knowledge about a language strategy requires both a semantic component for handling the formation, learning and adaptation of the relevant conceptual system, and a linguistic component doing the same for the related linguistic system (Steels 2011b).

Agent-based experiments in evolutionary linguistics aim to explain how particular language systems may emerge and evolve, assuming all the individuals in the population share the same strategy. In the long term these experiments should also be able to explain how new language strategies can emerge and propagate in a population. But most work has focused so far on the emergence of particular language systems, and on the study of the evolution over time of various macroscopic properties of these language systems, such as their effectiveness for communication, the average size of the agents' vocabularies or the similarity of the grammatical constructions they use to express certain types of meanings.

Examples of language systems that have been studied using agent-based computer simulations are: (1) *case systems* to express the role of participants in events (Steels 1998; Batali 1998; van Trijp 2010); (2) *vague context-dependent quantifiers* (e.g. many, some) (Pauw and Hilferty 2012); (3) *agreement markers* as a way to group words together (Beuls and Steels 2013); or (4) *colour vocabularies* in co-evolution with *colour categories* (Steels and Belpaeme 2005).

The rest of this paper is organised as follows. Firstly, we explain what we mean by a language system of logical constructions and specify the set of logical categories the agents can use to construct logical combinations of basic categories. Secondly, we introduce the formalism used to represent the agents' grammars. Then, we describe the particular type of linguistic interaction that allows the population to construct a shared lexicon and a grammar, focusing on two important cognitive abilities: induction and adaptation. Next, we present the results of some experiments in which a population of autonomous agents constructs a language system of logical constructions and transmits it to succeeding generations. Finally, we discuss related work and summarise the main contributions of the paper.

2 Logical Constructions

In this paper we focus on the study of the emergence and evolution of a language system of logical constructions. We consider a scenario in which a group of agents try to communicate about *subsets of objects* of the set of all the objects in a given context. We assume the agents have developed already a common vocabulary for referring to some individual object features, such as being up or to the left. These features are represented by propositional symbols such as *up* or *le* in the agents' memories, and denote the propositions 'I am referring to the objects which are up' and 'I am referring to the objects which are to the left' respectively.

We also assume the agents have developed some logical categories which allow them to construct propositional logic formulas from propositional symbols, although they have not learnt yet to express these logical categories nor the logical formulas they can construct with them in their shared language. The agents are therefore able to construct complex meanings, such as 'I am referring to the objects

which are either up or to the left, but not both', but they do not know how to express such meanings through language. The goal of the experiment is thus to show that a population of autonomous agents with the characteristics we have described above can construct a shared language (i.e. a vocabulary and a set of grammatical constructions) that allows them to communicate such complex meanings.

The particular set of logical categories the agents can use to construct logical formulas is not the set of connectives of propositional logic (i.e. $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$), but the set of unary and binary Boolean functions *not*, *and*, *nand*, *or*, *nor*, *if*, *nif*, *oif*, *noif*, *iff* and *xor*. The meaning of these functions, assuming they are applied to propositions *A* and *B*, can be expressed using the five connectives of propositional logic as follows $\neg A, A \wedge B, \neg(A \wedge B), A \vee B, \neg(A \vee B), A \rightarrow B, \neg(A \rightarrow B), B \rightarrow A, \neg(B \rightarrow A), A \leftrightarrow B$ and $(A \vee B) \wedge \neg(A \wedge B)$ respectively. However, the representation of their meaning as propositional logic formulas which use only these five connectives requires sometimes the use of more than one logical connective in the same formula, as in $[\text{xor}, \text{up}, \text{le}] \equiv (\text{up} \vee \text{le}) \wedge \neg(\text{up} \wedge \text{le})$, and thus the ability to construct meanings (formulas) with several levels of recursion, which is not assumed in the present experiment¹.

3 Agents' Grammars

We use *Prolog Grammar Rules* (Colmerauer et al. 1972; Pereira and Warren 1980) to represent the grammars constructed by the individual agents. The head of such rules is an atomic formula whose predicate symbol denotes a syntactic category (e.g. *s* for sentence) and whose arguments specify a number of aspects of the phrase or constituent described by that rule. In this paper we use two arguments. The first argument conveys semantic information and the second one a score in the interval $[0,1]$ that estimates the usefulness of that rule in previous communication. In our experiments, semantic information can be a proposition, a Boolean function, or a non-recursive logical formula constructed from the others. Logical formulas are represented using Lisp-like (McCarthy 1960) prefix notation.

Let us consider some examples of grammars the agents could use to express formula $[\text{and}, \text{le}, \text{up}]$. The first grammar consists of a single rule which states that 'izqyarr' is a valid sentence meaning $[\text{and}, \text{le}, \text{up}]$.

$$s([\text{and}, \text{le}, \text{up}], S) \rightarrow \text{izqyarr}, \{S \text{ is } 0.1\} \quad (1)$$

The same formula could also be expressed by using the following compositional grammar:

$$s(\text{up}, S) \rightarrow \text{arr}, \{S \text{ is } 0.70\} \quad (2)$$

$$s(\text{le}, S) \rightarrow \text{izq}, \{S \text{ is } 0.25\} \quad (3)$$

$$c2(\text{and}, S) \rightarrow y, \{S \text{ is } 0.50\} \quad (4)$$

$$s([P, Q, R], S) \rightarrow 2, c2(P, S1), s(Q, S2), s(R, S3), \{S \text{ is } S1 \cdot S2 \cdot S3 \cdot 0.1\} \quad (5)$$

¹Formulas constructed with Boolean functions are written using prefix notation, e.g. the list $[\text{iff}, A, B]$ is equivalent to $A \leftrightarrow B$.

The number appearing in the first place on the right hand side of some grammar rules (e.g. rule 5) indicates the position of the word associated with the Boolean function in the sentence: 1 indicates that the word associated with the Boolean function is placed in the first position, 2 in the second position, and 3 in the third position. This convention is necessary, because left recursive grammar rules cannot be used in Prolog. So the actual sentence generated by the second grammar is ‘2yizqarr’, which can be parsed into ‘izqyarr’. This grammar breaks down the sentence ‘izqyarr’ into subparts with independent meanings and the whole sentence is constructed by concatenating these subparts. The meaning of the sentence is obtained by combining the meanings of its subparts², using variables P , Q and R .

4 Language Games

Language acquisition is seen in this paper as a collective activity by which a group of autonomous agents construct a *common language system* as a result of a self-organisation process of the linguistic interactions that take place among them. In our experiments the agents interact with each other playing *language games* (Wittgenstein 1953). The particular type of language game we use is played by two agents, a speaker and a hearer, randomly chosen from the population. The main steps of its algorithm are described in figure 1.

1. The speaker chooses a Boolean formula (i.e. a meaning) from its set of conceptualisations of the subset of objects that constitutes the topic of the language game, generates or invents a sentence that expresses this formula, and communicates that sentence to the hearer.
2. The hearer tries to interpret the sentence communicated by the speaker. If it can parse it using its lexicon and grammar, it extracts a meaning (i.e. a Boolean formula). If the hearer cannot parse the sentence, the speaker communicates the Boolean formula it had in mind to the hearer, and the hearer adopts an association between that formula and the sentence used by the speaker.
3. Depending on the outcome of the language game speaker and hearer expand and adapt their grammars to be more successful in future language games.

A language game is considered *successful* if the hearer can parse the sentence communicated by the speaker and if its interpretation of that sentence is a Boolean formula which is logically equivalent to the formula the speaker had in mind; otherwise, the language game *fails*.

4.1 Generation and Invention

At the early stages of a simulation run the agents cannot use their internal grammars to generate sentences for most meanings, because they all start with a common lexicon for basic categories, but no grammar rules. In order to let language get off the ground, the agents are allowed to invent new sentences for those meanings they cannot express using their grammars.

²Note that in Prolog variables start with capital letters and constants with lower case.

function GENERATE($Agent, Meaning$) **returns** $Sentence, CompetingSentences$

$Sentence$ is one of the sentences with highest score $Agent$ can generate to express $Meaning$. The rest of the sentences are $CompetingSentences$.

function INTERPRET($Agent, Sentence$) **returns** $Meaning, CompetingMeanings$

$Meaning$ is one of the meanings (formulas) with highest score $Agent$ can obtain parsing $Sentence$. The rest of the meanings are $CompetingMeanings$.

procedure LANGUAGE_GAME($Speaker, Topic, Hearer$)
 $Speaker$ chooses $Meaning$ from CONCEPT($Speaker, Topic$)
 $S_s, CS_s \leftarrow$ GENERATE($Speaker, Meaning$)
if $\{S_s\} \cup CS_s \neq \emptyset$ **then** $Speaker$ communicates S_s to $Hearer$
else
 $Speaker$ cannot generate a sentence to express $Meaning$
 $S_s \leftarrow$ INVENT($Speaker, Meaning$)
ADOPT_ASSOCIATION($Speaker, Meaning, S_s$)
 $Speaker$ communicates S_s to $Hearer$
 $M_h, CM_h \leftarrow$ INTERPRET($Hearer, S_s$)
if $\{M_h\} \cup CM_h \neq \emptyset$ **then**
if $Meaning$ is logically equivalent to M_h **then**
the language game succeeds
else the language game fails
 $Speaker$ communicates $Meaning$ to $Hearer$
ADAPT($Hearer, S_s, Meaning, M_h, CM_h$)
else
 $Hearer$ cannot parse sentence S_s , language game fails
 $Speaker$ communicates $Meaning$ to $Hearer$
ADOPT_ASSOCIATION($Hearer, Meaning, S_s$)

Figure 1: Algorithm of the language game used in the model.

A new sentence E for a formula F is invented as follows. If F is atomic, invention is not necessary because there exists a word in the common lexicon that expresses F . New words are generated for logical categories (i.e. Boolean functions of one or two arguments) as sequences of three to six letters randomly chosen from the alphabet. If F is a non-recursive Boolean formula such as $[not, A]$ or $[\otimes, A, B]$, a word is generated for each propositional symbol using the common lexicon, a new word is invented for the Boolean function, and the two or three words generated (depending on the type of formula) are concatenated in random order.

As the agents play language games, they learn associations between expressions and meanings, and induce linguistic knowledge from such associations in the form of grammar rules and lexical entries. Once they can generate a sentence to express a particular meaning using their grammars, they do not keep inventing new sentences. Instead, they select the sentence with the highest score from the set of all the sentences they can generate to express that meaning, and communicate that sentence to the hearer.

The *score of a sentence* (or a *meaning*) generated using a compositional grammar rule is computed using the arithmetic expression on the right hand side of that rule (Vogt 2005). Consider the generation of sentence ‘izqyarr’ for expressing meaning $[and, le, up]$ using rules 2, 3, 4 and 5.

The score $S = 0.00875$ of sentence ‘izqyarr’, generated by rule 5, is computed multiplying the score of that rule (0.1) by the scores of rules 2, 3 and 4, which generate the words associated with the constituents of that sentence (0.70, 0.25 and 0.50, respectively). The *score of a grammar rule* is the last number in the arithmetic expression that appears on the right hand side of that rule.

4.2 Interpretation and Adoption

In the second step of a language game the hearer tries to interpret the sentence communicated by the speaker using its own grammar. However, at the early stages of a simulation run the agents cannot parse most of the sentences communicated by the speakers, because they start with a common lexicon for basic categories but no grammar rules. In this case the speaker communicates the formula (meaning) F it had in mind to the hearer, and the hearer adopts an association between that formula and the sentence E used by the speaker, adding a new rule of the form $s(F, S) \rightarrow E, \{S \text{ is } 0.1\}$ to its grammar. The scores of the rules invented or adopted by the agents are given an initial value of 0.1.

Invention and adoption allow thus the agents to construct and learn associations between sentences and meanings. From these associations they induce linguistic knowledge, i.e. grammatical constructions and lexical entries, they incorporate to their grammars and use in subsequent language games to generate and interpret other sentences.

Once the agents can interpret a sentence using their own grammar, they select the meaning (formula) with the highest score from the set of all the meanings they can obtain for that sentence. However, at halfway stages of a simulation run the grammars of speaker and hearer are often not compatible, and the interpretation chosen by the hearer is not logically equivalent to meaning the speaker had in mind. The reason for this is that each agent constructs its internal language from the linguistic interactions it participates in, and speaker and hearer never share the same history of linguistic interactions unless the population consists only of these two agents. The strategy used to coordinate the grammars of speaker and hearer when this happens is to decrease the scores of the rules used by the hearer to obtain its interpretation of the sentence communicated by the speaker.

4.3 Induction

As mentioned above the agents extract generalisations in the form of grammar rules and lexical entries from the associations between sentences and meanings they invent or adopt from other agents. The induction rules used in the experiment are the rules *simplification* and *chunk* in (Kirby 2002). Induction is applied whenever the agents invent or adopt an association between a sentence and a meaning, to avoid redundancy and increase generality in their grammars.

Simplification *Let $r1$ and $r2$ be a pair of grammar rules such that the semantic argument of the left hand side of $r1$ contains a subterm $m1$, $r2$ is of the form $n(m1, S) \rightarrow e1, \{S \text{ is } C1\}$, and $e1$ is a substring of the terminals of $r1$. Then simplification can be applied to $r1$ replacing it*

with a new rule that is identical to $r1$ except that: (1) $m1$ is replaced with a new variable X in the semantic argument of the left hand side; (2) $e1$ is replaced with $n(X, S)$ on the right hand side; and (3) the arithmetic expression $\{R \text{ is } E \cdot C2\}$ on the right hand side of $r1$ is replaced with an arithmetic expression of the form $\{R \text{ is } E \cdot S \cdot 0.1\}$, where $C1$ and $C2$ are constants in the range $[0,1]$, and E is the product of the score variables on the right hand side of $r1$.

Let us see an example of how simplification works. Suppose an agent’s grammar contains rules 2 and 3. It plays a language game, and invents or adopts the following rule.

$$s([and, le, up], S) \rightarrow izqyarr, \{S \text{ is } 0.1\}. \quad (6)$$

It could apply simplification to rule 6 (using rule 2) and replace it with 7.

$$s([and, le, R], S) \rightarrow izqy, s(R, SR), \{S \text{ is } SR \cdot 0.1\} \quad (7)$$

Now rule 7 could be simplified again (using rule 3), replacing it with rule 8, which contains specific information about the word ‘y’ associated with Boolean function *and*, the position of such word in the sentence (2nd), and the relative positions of the words associated with the arguments of Boolean function *and* in the sentence (not inverted³).

$$s([and, Q, R], S) \rightarrow 2, y, s(Q, SQ), s(R, SR), \{S \text{ is } SQ \cdot SR \cdot 0.1\} \quad (8)$$

If the agent invents or adopts a rule that associates sentence ‘izqoarr’ with formula $[or,le,up]$ and applies simplification, then its grammar would contain the following rule.

$$s([or, Q, R], S) \rightarrow 2, o, s(Q, SQ), s(R, SR), \{S \text{ is } SQ \cdot SR \cdot 0.1\} \quad (9)$$

Chunk I *Let $r1$ and $r2$ be a pair of rules with the same left hand side category symbol. If the semantic arguments of the left hand sides of the rules differ only in one subterm $m1$ and $m2$, and there exist two strings of terminals $e1$ and $e2$ that, if replaced with the same non-terminal, would make the right hand sides of the rules identical, then chunk can be applied as follows. A new category symbol c is created and the following new rules are added to the grammar.*

$$c(m1, S) \rightarrow e1, \{S \text{ is } 0.1\} \quad c(m2, S) \rightarrow e2, \{S \text{ is } 0.1\}$$

Rules $r1$ and $r2$ are replaced with a single rule that is identical to $r1$ except that: (1) $m1$ is replaced with a new variable X in the semantic argument of the left hand side; (2) $e1$ is replaced with $c(X, S)$ on the right hand side; and (3) the arithmetic expression $\{R \text{ is } E \cdot C1\}$ on the right hand side of $r1$ is replaced with a new arithmetic expression of the form $\{R \text{ is } E \cdot S \cdot 0.1\}$, where $C1$ is a constant in the range $[0,1]$ and E is the product of the score variables that appeared on the right hand side of $r1$.

The agent could now apply chunk I to rules 8 and 9 generating a syntactic category $c2$ for binary Boolean functions.

$$s([P, Q, R], S) \rightarrow 2, c2(P, S1), s(Q, S2), s(R, S3), \{S \text{ is } S1 \cdot S2 \cdot S3 \cdot 0.1\} \quad (10)$$

³Rule $s([and, Q, R], S) \rightarrow 2, y, s(R, SR), s(Q, SQ), \{S \text{ is } SR \cdot SQ \cdot 0.1\}$ that generates sentence ‘arryizq’ to express $[and, le, up]$, instead of sentence ‘izqyarr’, inverts the order of the arguments.

$$c2(\text{and}, S) \rightarrow y, \{S \text{ is } 0.1\} \quad (11)$$

$$c2(\text{or}, S) \rightarrow o, \{S \text{ is } 0.1\} \quad (12)$$

Rules 8 and 9 would be replaced with rule 10, which can be applied to formulas constructed with other binary Boolean functions, and rules 11 and 12, which state that words ‘y’ and ‘o’ mean ‘and’ and ‘or’, and belong to syntactic category $c2$, would be added to the grammar.

Chunk II *If the semantic arguments of the left hand sides of two rules $r1$ and $r2$ can be unified applying substitution $X/m1$ to $r1$, and there is a string of terminals $e1$ in $r2$ that corresponds to $c(X, S)$ in $r1$, then rule $r2$ can be replaced with a new rule of the form $c(m1, S) \rightarrow e1, \{S \text{ is } 0.01\}$.*

Suppose the agent adopts or invents the following rule.

$$s([\text{if}, le, up], S) \rightarrow \text{izqsiarr}, \{S \text{ is } 0.1\} \quad (13)$$

Simplification of 13, 2, 3 would replace rule 13 with 14.

$$s([\text{if}, Q, R], S) \rightarrow 2, \text{si}, s(Q, SQ), s(R, SR), \\ \{S \text{ is } SQ \cdot SR \cdot 0.1\} \quad (14)$$

Chunk II, applied to 14 and 10, would replace 14 with 15.

$$c2(\text{if}, S) \rightarrow \text{si}, \{S \text{ is } 0.1\} \quad (15)$$

4.4 Adaptation

Coordination of the agents’ grammars is necessary, because different agents can invent different words to refer to the same Boolean function and they may concatenate the words associated with the components of a Boolean formula in different orders when they try to express it as a sentence. In the experiments discussed in this paper coordination is achieved through a process of self-organisation of the agents’ linguistic interactions that takes place when these agents adapt their preferences for vocabulary and grammatical constructions to those they observe are used more often by other agents.

The agents adapt the scores of their grammar rules (i.e. their preferences for vocabulary and grammatical constructions) at the last step of a language game, when the speaker communicates the meaning it had in mind to the hearer, and only in the case in which the speaker can generate at least one sentence for the meaning it is trying to communicate using its grammar and the hearer can parse the sentence communicated by the speaker. In a language game only the agent playing the role of hearer adapts the scores of its grammar rules. However, as all the agents in the population play both the role of speaker and that of hearer in different language games, all the agents have plenty of opportunities to adapt the scores of their grammar rules during a simulation run.

The procedure for adapting the scores of the hearer’s grammar rules is sketched in figure 2. It assumes that: (1) the speaker has communicated to the hearer a sentence S_s ; (2) the hearer has been able to parse S_s , obtaining a meaning M_h with highest score and a possibly non-empty set of alternative meanings CM_h , called *competing meanings*; and (3) the speaker has communicated to the hearer a formula *Meaning* that represents the meaning the speaker had in mind.

If the meaning interpreted by the hearer is logically equivalent to the meaning the speaker had in mind, the game succeeds. The hearer takes this interaction as a positive example

function RULES_G(*Agent, Meaning, Sent*) **returns** *Rules*
Rules are the grammar rules used by *Agent* to generate sentence *Sent* in order to express *Meaning*.

function RULES_I(*Agent, Sent, Meaning*) **returns** *Rules*
Rules are the grammar rules used by *Agent* to obtain *Meaning* parsing sentence *Sent*.

procedure ADAPT(*Hearer, S_s, Meaning, M_h, CM_h*)

S_s is the sentence communicated by the speaker

Meaning is the meaning the speaker had in mind

M_h is the meaning chosen by the hearer

if *Meaning* is logically equivalent to M_h **then**

The language game succeeds

Hearer adjusts scores at the level of interpretation

$R_{hi} \leftarrow \text{RULES_I}(\text{Hearer}, S_s, M_h)$

INCREASE_SCORES(Hearer, R_{hi})

for each M **in** CM_h **do**

$R_M \leftarrow \text{RULES_I}(\text{Hearer}, S_s, M)$

DECREASE_SCORES(Hearer, ($R_M \setminus R_{hi}$))

end

Hearer adjusts scores at the level of generation

$S_h, CS_h \leftarrow \text{GENERATE}(\text{Hearer}, \text{Meaning})$

if $S_s \in \{S_h\} \cup CS_h$ **then**

$R_{hg} \leftarrow \text{RULES_G}(\text{Hearer}, \text{Meaning}, S_s)$

INCREASE_SCORES(Hearer, R_{hg})

for each S **in** ($\{S_h\} \cup CS_h$) $\setminus \{S_s\}$ **do**

$R_S \leftarrow \text{RULES_G}(\text{Hearer}, \text{Meaning}, S)$

DECREASE_SCORES(Hearer, ($R_S \setminus R_{hg}$))

end

else

The language game fails

$R_{hi} \leftarrow \text{RULES_I}(\text{Hearer}, S_s, M_h)$

DECREASE_SCORES(Hearer, R_{hi})

Figure 2: Algorithm for adapting the scores of the hearer’s grammar rules. The score of each grammar rule is adjusted only once, even if it is used for generating or obtaining more than one sentence or more than one meaning. Set difference, $R_M \setminus R_{hi}$ or $R_S \setminus R_{hg}$, is used to avoid decreasing scores of rules used for obtaining the speaker’s meaning or sentence.

and adjusts the scores of its grammar rules both at the level of interpretation and at that of generation. First, the hearer increases the scores of the rules it used for obtaining the meaning the speaker had in mind and decreases the scores of the rules it used for obtaining competing meanings. Then, the hearer tries to simulate what it would have said if it were in the speaker’s place: it tries to express the meaning the speaker had in mind using its own grammar, and it increases the scores of the grammar rules that generate the sentence that was chosen by the speaker and decreases the scores of the rules that generate competing sentences.

Adjusting scores at the level of interpretation means decreasing the scores of associations between the sentence used by the speaker and ‘competing meanings’. This reduces ambiguity by discouraging a phenomenon similar to homonymy but happening at sentence level. Adjusting the scores at the level of generation means decreasing the scores

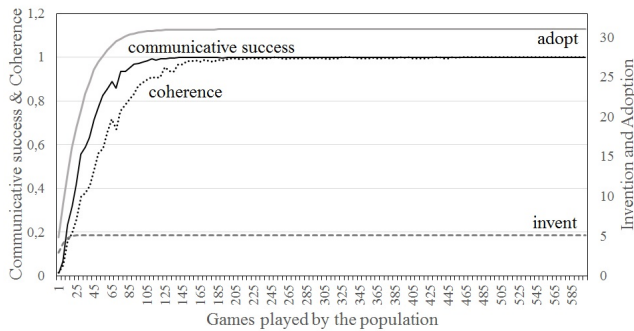


Figure 3: Evolution of communicative success, coherence, invention and adoption in an experiment with a population of ten agents playing 6060 language games. The results are the average of ten simulation runs with different random seeds.

of associations between the meaning the speaker had in mind and ‘competing sentences’ the hearer could use for expressing that meaning. This reduces ambiguity by discouraging a phenomenon similar to synonymy but happening at sentence level. The scores of the rules used by the hearer to obtain the meaning the speaker had in mind are increased only once.

If the meaning interpreted by the hearer is not logically equivalent to the meaning the speaker had in mind, the game fails, and the hearer decreases the scores of the rules it used for interpreting the sentence communicated by the speaker.

The scores of grammar rules are updated using the following formulas with alignment rate $\mu = 0.1$. The rule’s original score S is replaced with the result of evaluating expression 16 if the score is *increased*, and with the result of evaluating expression 17 if the score is *decreased*.

$$S \cdot (1 - \mu) + \mu \quad (16)$$

$$S \cdot (1 - \mu) \quad (17)$$

5 Experiments

The agent-based model described in this paper has been implemented in Prolog (Bueno et al. 1997) and tested conducting a series of experiments that study both the emergence of a shared language system of logical constructions and its transmission from one generation to the next.

In the first experiment, which studies language emergence, the agents start with a common lexicon for six basic categories but no grammar rules. Then they play 6060 language games about logical formulas constructed applying unary and binary Boolean functions to basic categories. Figure 3 shows the evolution over time⁴ of four measures which monitor the population’s global performance. *Communicative success* is the average of successful language games in the last ten games played by the agents. The population reaches full communicative success (i.e. 1.0) in 1950 language games, i.e. after each agent has played 390 games. *Coherence* measures the similarity of the agents’ grammars. It is the average of language games in which: (1) the hearer understands correctly the sentence communicated by the

⁴Each point on the x-axes of figures 3, 4 and 5 represents ten consecutive language games in the simulation.

speaker, and (2) the hearer would use the same sentence as the sentence used by the speaker to communicate the meaning the speaker had in mind. Coherence increases more slowly than communicative success. Full coherence (1.0) is reached in 4600 games (i.e. 920 games per agent).

Invention (respectively *adoption*) is the number of sentences invented (respectively adopted) by an agent in past language games. The data shown in figure 3 are the average values of invention (adoption) for a population of ten agents in ten simulation runs⁵. As it can be observed, invention grows relatively rapidly during the first 550 games, reaching a maximum average value of 5.14 inventions per agent. At that point the agents must have learnt all the vocabulary and grammatical constructions required to express logical formulas constructed with unary and binary Boolean functions. The average number of sentences adopted by the agents keeps growing during the first 1900 games, reaching a maximum average value of 31.03 adoptions per agent. At that point the agents must have learnt the vocabulary and grammatical constructions used by the other agents.

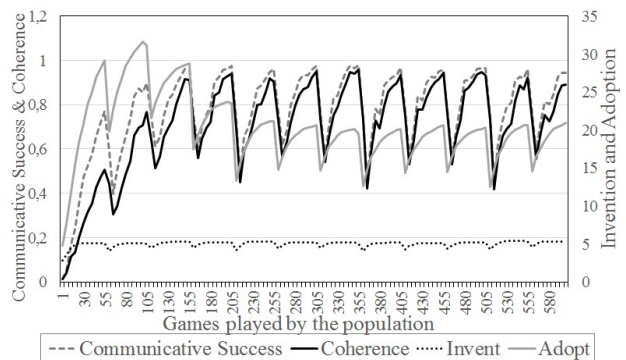


Figure 4: Evolution of communicative success, coherence, invention and adoption in an experiment studying language transmission with a population of ten agents playing 6060 language games. The data shown in the graph are the average of ten simulation runs with different initial random seeds.

Figure 4 shows the results of an experiment studying language transmission across generations. The agents in the population are divided into three groups: the elder, the adults and the young. Every 500 games the elder (approximately one third of the population) are replaced with new agents which have neither a vocabulary for Boolean functions nor grammar rules. The previous adults become the elder, the young the adults, and the new ones the younger generation. As a consequence the population is completely renewed after 1500 games. Every time new agents are introduced the four measures decrease, but they catch up before the next generation of agents is introduced. Communicative success typically reaches values over 0.96, which dip to 0.50 when new agents are introduced. Coherence follows closely communicative success, reaching values over 0.94. The final average value of inventions per agent (5.36)

⁵The same number of simulation runs is used in language emergence and evolution studies such as (Vogt 2005) and (Kirby 2002).

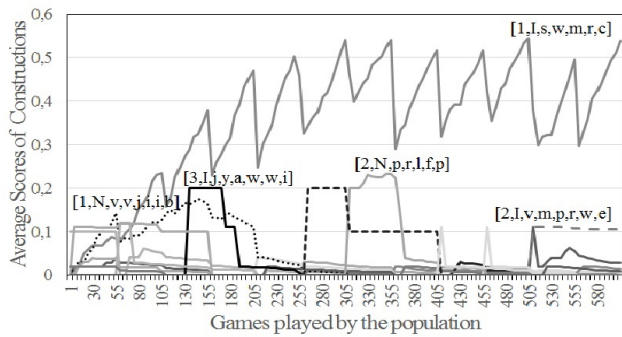


Figure 5: Average score evolution of constructions expressing exclusive disjunction (*xor*) in one simulation run of the experiment studying language transmission. Once construction [1,I,s,w,m,r,c] is established as the preferred way to express exclusive disjunction in the second generation, it is transmitted without change to succeeding generations.

is slightly larger than in the experiment with a single generation (5.14). Adoption, however, reaches a smaller final average value (20.95) than in the experiment with a single generation (31.03). This is due to the fact that new agents learn a language already established in the population (i.e. the language that is transmitted from generation to generation in the experiment), which uses fewer variations –typically one or two– for expressing a given meaning. Whereas agents created in the first stages of the experiment must learn all the constructions invented by the rest of the population.

Figure 5 shows the evolution of the scores of the twenty nine constructions invented by the agents to express exclusive disjunction (*xor*) in one simulation run of the experiment studying language transmission. Each line in the graph displays the evolution of the average score of one particular construction for the ten agents in the population. For readability reasons we have labelled only five of the twenty nine lines shown in the graph. These correspond to some constructions that have had a significant average score for some relatively long period of time during the simulation. Constructions are labelled with the word used to express Boolean function *xor* in that particular construction, preceded by a number which indicates the position of that word in the sentence, and by the letter ‘I’ if the construction inverts the order of the words associated with the arguments of *xor* in the sentence or by the letter ‘N’ if it does not invert them. As it can be observed the agents use constructions which associate different words with Boolean function *xor*, place these words in the first, second or third position of the sentence, and invert (e.g. [2,I,v,m,p,r,w,e]) or do not invert (e.g. [2,N,p,r,l,f,p]) the order of the words associated with the arguments of *xor* in the sentence. However, once construction [1,I,s,w,m,r,c] begins to be preferred by the agents in the second generation (after the first 750 games), its average score keeps dominating the scores of the other constructions in succeeding generations, although it decreases every time new agents are introduced in the population. This means that this construction is transmitted without change from the second generation to the last one in the experiment.

6 Contributions and Related Work

This paper has used an agent-based model to study the emergence and evolution of a language system of logical constructions. The model has been implemented in Prolog and it has been tested conducting a series of experiments which simulate a scenario where a group of autonomous agents try to communicate about subsets of objects characterised by logical combinations of basic categories. The results of these experiments show that a shared vocabulary for Boolean functions and a set of grammatical constructions (which use word order to express the relation between Boolean functions and their arguments in a sentence) emerge as a result of a process of self-organisation of the agents’ interactions when these agents adapt their preferences for vocabulary and grammatical constructions to those they observe are used more often by other agents. And they also show that the same invention, adoption, induction and adaptation mechanisms that allow a group of agents to construct a shared language system of logical constructions enable the transmission of such a language system from one generation to succeeding ones in the experiments.

Within the evolutionary linguistics literature, the work presented in this paper belongs to the set of experiments that study grammar acquisition (Batali 1998; Steels 1998; Kirby 2002; Smith, Brighton, and Kirby 2003) rather than lexicon acquisition (Steels 1995; Lara and Alfonseca 2000; 2002; Steels and Belpaeme 2005). Word order plays a crucial role in our experiments, because the position of each subexpression in a sentence determines how it is semantically related to the rest of subexpressions: whether it names a proposition or a Boolean function, or whether it is the first or the second argument of a Boolean function (e.g. the antecedent or the consequent of an implication). (Steels 1998) and (Kirby 2002) also study the emergence of word-order based grammar. But they do not address the issue of negotiation, because the populations in these works consist only of two agents. In our experiments, however, the population consists of ten agents, that need to agree on how to order the expressions associated with the constituents of each different type of Boolean formula to construct a sentence. There are eleven types of Boolean formulas and six possible orderings for each (except for negation, which has only two possible orderings). And they also need to agree on how to name eleven Boolean functions.

Our work also differs from (Kirby 2002) in testing communicative success using logical equivalence of formulas (meanings), rather than syntactic equality. This is reflected in the language systems constructed by the agents in our experiments, which may use a single word for referring to different Boolean functions if these Boolean functions have redundant semantics, and which do not always impose a strict word-order between the expressions associated with the arguments of commutative Boolean functions.

(Beuls and Steels 2013) and (van Trijp 2010) also study grammar acquisition, but they use agreement and case markers (i.e. suffixes attached to words) instead of word order as the syntactic means for semantic disambiguation. They also use simulations with software agents (rather than experiments with physical robots interacting with each other

in a real world environment), initialise the agents with a pre-defined vocabulary for basic properties, and use a language game in which the topic consists of several objects. However, their agents conceptualise the topic using a set of distinctive properties in which each property refers to a single object of the topic, and the role of *agreement* (respectively *case*) markers is to indicate which properties of the distinctive set refer to the same object (or which properties of the distinctive set refer to the same role for *case* markers). Therefore, the set of meanings the agents can construct in these works, and in (Vogt 2005), are only conjunctions of basic properties. In our experiments, in contrast, the agents conceptualise the topic constructing a discriminating logical formula, which can be a composition of basic properties through conjunction, disjunction, negation or any other Boolean function, and which is true for every object in the topic and false for the rest of the objects in the context.

Finally in order to conduct the experiments presented in this paper, we have implemented a set of software tools that allow performing language evolution experiments in Prolog. These tools are different from the Lisp simulation and grammar processing systems (Steels 2011a) used in other experiments (van Trijp 2010; Steels 2012; Beuls and Steels 2013), but can also be combined in interesting ways with them.

Acknowledgments

The author would like to thank Manuel Alfonseca, Diane Fleming, Luc Steels, Paul Vogt, Glyn Morrill and Ivan Balañá for their help at different stages during the realisation of this work. The research leading to these results has received funding from the Spanish Ministry of Economy and Competitiveness under project TIN2011-27479-C04-03.

References

- Batali, J. 1998. Computational simulations of the emergence of grammar. In *Approaches to the evolution of language: social and cognitive bases*, 405–426. Cambridge Univ Press.
- Beuls, K., and Steels, L. 2013. Agent-based models of strategies for the emergence and evolution of grammatical agreement. *PLoS ONE* 8(3):e58960.
- Briscoe, T., ed. 2002. *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press.
- Bueno, T.; Cabeza, D.; Carro, M.; Hermenegildo, M.; Lopez-Garcia, P.; and Puebla, G. 1997. The Ciao Prolog system. Reference manual. Technical Report CLIP3/97.1, Technical University of Madrid.
- Colmerauer, A.; Kanoui, H.; Pasero, R.; and Roussel, P. 1972. Un système de communication en français. Technical Report I Groupe Intelligence Artificielle, Université Aix-Marseille II.
- Croft, W. 2008. Evolutionary linguistics. *Annual Review of Anthropology* (37):219–234.
- Di Sciullo, M., and Boeckx, C., eds. 2011. *The biolinguistic enterprise. New perspectives on the evolution and nature of the human language faculty*. Oxford University Press.
- Hurford, J.; Studdert-Kennedy, M.; and Kight, C., eds. 1998. *Approaches to the evolution of language: social and cognitive bases*. Cambridge University Press.
- Kirby, S. 2002. Learning, bottlenecks and the evolution of recursive syntax. In *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, 96–109. Cambridge University Press.
- Lara, J., and Alfonseca, M. 2000. Some strategies for the simulation of vocabulary agreement in multi-agent communities. *Journal of Artificial Societies Social Simulation* 3(4).
- Lara, J., and Alfonseca, M. 2002. The role of oblivion, memory size and spatial separation in dynamic language games. *Journal of Artificial Societies and Social Simulation* 5(2).
- Lyon, C.; Nehaniv, C.; and Cangelosi, A., eds. 2007. *Emergence of language and communication*. Lecture Notes in Computer Science. Berlin: Springer-Verlag.
- McCarthy, J. 1960. Recursive functions of symbolic expressions and their computation by machine, part I. *Commun. ACM* 3(4):184–195.
- Minett, J., and Wang, W., eds. 2005. *Language acquisition, change and emergence: essays in evolutionary linguistics*. Hong Kong: City University of Hong Kong Press.
- Pauw, S., and Hilferty, J. 2012. The emergence of quantification. In *Experiments in cultural language evolution*, 277–304. John Benjamins.
- Pereira, F., and Warren, D. 1980. Definite clause grammars for language analysis a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence* 13(3):231–278.
- Smith, K.; Brighton, H.; and Kirby, S. 2003. Complex systems in language evolution: the cultural emergence of compositional structure. *Advances in Complex Systems* (6(4)):537–558.
- Steels, L., and Belpaeme, T. 2005. Coordinating perceptually grounded categories through language: a case study for colour. *Behavioral and Brain Sciences* 28:469–529.
- Steels, L. 1995. A self-organizing spatial vocabulary. *Artificial Life* 2(3):319–332.
- Steels, L. 1998. The origins of syntax in visually grounded robotic agents. *Artificial Intelligence* 103(1-2):133–156.
- Steels, L., ed. 2011a. *Dessign patterns in Fluid Construction Grammar*. John Benjamins.
- Steels, L. 2011b. Modeling the cultural evolution of language. *Physics of Life Reviews* 8:339–356.
- Steels, L., ed. 2012. *Experiments in Cultural Language Evolution*. John Benjamins.
- van Trijp, R. 2010. Grammaticalization and semantic maps: evidence from artificial language evolution. *Linguistic Discovery* 8(1):310–326.
- Vogt, P. 2005. The emergence of compositional structures in perceptually grounded language games. *Artificial Intelligence* 167(1-2):206–242.
- Wittgenstein, L. 1953. *Philosophical Investigations*. New York: Macmillan.