The Acquisition of Linguistic Competence for Communicating Propositional Logic Sentences

Josefina Sierra[1] and Josefina Santibáñez[2]

[1] Universidad Politécnica de Cataluña, Spain (jsierra@lsi.upc.edu**)
[2] Universidad de La Rioja, Spain (josefina.santibanez@unirioja.es)

Abstract. We describe some experiments which show how a *language* expressive enough to allow the communication of meanings of the same complexity as propositional logic formulas can emerge in a population of autonomous agents which have no prior linguistic knowledge. We take an approach based on *general purpose cognitive capacities,* such as invention, adoption and induction, and on *self-organisation* principles applied to a particular type of linguistic interaction known as a *language game.* These experiments extend previous work by considering a larger population and a much larger search space of grammar rules. In particular the agents are allowed to order the expressions associated with the constituents of a logical formula in arbitrary order in the sentence. Previous work assumed that the expressions associated with the connectives should be always placed in the first position of the sentence. Another difference is that communication is considered successful in a language game if the meaning interpreted by the hearer is *logically equivalent* to the meaning the speaker had in mind. In previous experiments the meanings of speaker and hearer were required to be syntactically equal. This allows us to observe how a less strict grammar in terms of word order emerges through the self-organisation process, which minimizes the learning effort of the agents by imposing only those order relations among the components of a sentence that are necessary for language understanding.

1 INTRODUCTION

This paper addresses the problem of the acquisition of a language (i.e., a lexicon and a grammar) expressive enough to allow the communication of meanings that can be represented by propositional logic formulas. We take an approach based on *general purpose cognitive capacities,* such as invention, adoption and induction. Coordination of the linguistic knowledge acquired by the individual agents is achieved through a *self-organisation* process of the linguistic interactions that take place between pairs of agents of the population.

We describe some experiments which show how a shared set of preferred lexical entries, syntactic categories and grammatical constructions (i.e., a *language*) can emerge in a population of autonomous agents which have no prior linguistic knowledge. This shared language is expressive enough to allow the agents

** This work is partially funded by the DGICYT TIN2005-08832-C03-03 project (MOISES-BAR).

to communicate any meaning that can be represented by a propositional logic formula.

These experiments extend previous work [1] by considering a larger population and a much larger search space of grammar rules. In particular the agents are allowed to order the expressions associated with the constituents of a logical formula in arbitrary order in the sentence. Previous work assumed that the expressions associated with the connectives should be always placed in the first position of the sentence. The branching factor of the search space of grammar rules that can be used for expressing formulas constructed with binary connectives is extended thus from two to six.

Another difference is that communication is considered successful in a language game if the meaning interpreted by the hearer is *logically equivalent* to the meaning the speaker had in mind. In previous experiments both meanings were required to be syntactically equal, i.e., the same formula. This allows us to observe how a less strict grammar in terms of word order emerges through the self-organisation process, which minimizes the learning effort of the agents by imposing only those order relations among the components of a sentence that are necessary for language understanding.

To understand how a population of autonomous agents might be able to come up with a language expressive enough to communicate propositional logic formulas is a problem of practical and theoretical interest. The important role of logic as a formalism for *knowledge representation* and *reasoning* [2] is well known in *artificial intelligence*. Much of the knowledge used by artificial intelligent agents today is represented in logic. In particular the recent development of efficient algorithms for checking satisfiability (*SAT solvers*) is increasing the number of practical applications that use propositional logic as its knowledge representation formalism. Logic is relevant as well for computational and cognitive *linguistics,* because it is the standard formalism used in these fields for representing semantic information (i.e., the meanings of words and sentences). On the other hand logical connectives and logical constructions are themselves a fundamental part of *natural language*, and they play a very important role in the *development of intelligence* and deductive reasoning [3–5]. Therefore from a scientific point of view it is necessary to understand how an agent can both conceptualise and communicate logical constructions to other agents.

The research presented in this paper assumes previous work on the conceptualisation of logical connectives [6, 7]. In [8] a grounded approach to the acquisition of logical categories (i.e., logical connectives) based on the discrimination of a "subset of objects" from the rest of the objects in a given context is described. The "subset of objects" is characterized by a logical formula constructed from perceptually grounded categories. This formula is satisfied by the objects in the subset and not satisfied by the rest of the objects in the context. In this paper we only focus on the problem of the acquisition of a language (a vocabulary and a grammar) suitable for expressing propositional logic formulas. Future work will address the complete problem with which children are faced which consists in acquiring both the semantics and the syntax of the logical constructions and connectives that are used in natural language.

The rest of the paper is organised as follows. First we introduce the formalism used for representing the grammars constructed by the agents. Then we describe the particular type of language game played by the agents, focusing on the main cognitive processes they use for constructing a shared lexicon and a grammar: invention, adoption, induction and co-adaptation. Next we present the results of some experiments in which a population of autonomous agents constructs a language that allows communicating propositional logic formulas. Finally we summarize some related work and the contributions of the paper.

2 GRAMMATICAL FORMALISM

The formalism used for representing the grammars constructed by the agents is *definite-clause grammar*. In particular *non-terminals* have three arguments with the following contents: (1) semantic information; (2) a *score* in the interval [0, 1] that estimates the usefulness of the rule in previous communication; and (3) a *counter* that records the number of times the rule has been used.

Let us consider some examples of grammars the agents could use to express the propositional formula $right \wedge light$[3]. The first grammar consists of a single rule which states that 'andrightlight' is a valid sentence meaning $right \wedge light$.

$$s([and, right, light]), S) \rightarrow andrightlight, \{S\ is\ 0.01\} \quad (1)$$

The same formula could be expressed as well using the following compositional, recursive grammar: s is the start symbol, $c2$ is the name of a syntactic category associated with binary connectives. Like in Prolog, variables start with a capital letter and constants with a lower case letter.

The number that appears in first place on the right hand side of a grammar rule (see rule 5) indicates the position of the expression associated with the connective in the sentence: The number 1 indicates that the expression associated with the connective is a *prefix* (first position), number 2 that it is an *infix* (second position), and number 3 that it is a *suffix* (third position). We use this convention because Prolog does not allow the use of left recursive grammar rules.

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

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

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

$$s([P, Q, R], S) \rightarrow 2,\ c2(P, S1), s(Q, S2), s(R, S3), \{S\ is\ S1 \cdot S2 \cdot S3 \cdot 0.01\} \quad (5)$$

This grammar breaks down the sentence 'rightandlight' into subparts with independent meanings. The whole sentence is constructed concatenating these

[3] Notice that we use Prolog grammar rules for describing the grammars. The semantic argument of non-terminals uses Lisp like (prefix) notation for representing propositional formulas (e.g., the Prolog list $[and, [not, right], light]$ is equivalent to $\neg right \wedge light$). The third argument (the use counter) of non-terminals is not shown in the examples.

subparts. The meaning of the sentence is composed combining the meanings of the subparts using the variables P, Q and R.

The agents can invent a large number of grammars to express the same formula, because they can associate different words with the propositional constants and connectives of the formula, and they can concatenate the expressions associated with the constituents of the formula in any order. The following grammar uses the sentence 'claroderechay' for expressing the same formula $right \wedge light$.

$$s(\text{light}, S) \rightarrow \text{claro}, \{S\ is\ 0.60\} \tag{6}$$

$$s(\text{right}, S) \rightarrow \text{derecha}, \{S\ is\ 0.40\} \tag{7}$$

$$c2(\text{and}, S) \rightarrow \text{y}, \{S\ is\ 0.50\} \tag{8}$$

$$s([P,Q,R], S) \rightarrow 3,\ c2(P, S1), s(R, S2), s(Q, S3), \{S\ is\ S1 \cdot S2 \cdot S3 \cdot 0.01\} \tag{9}$$

Coordination of the grammars constructed by the individual agents is therefore not a trivial task, because in order to understand each other the agents must use a common vocabulary and must order the constituents of compound sentences in sufficiently similar ways as to avoid ambiguous interpretations.

3 LANGUAGE GAMES

Language acquisition is seen thus as a collective process by which a population of autonomous agents constructs a *common language* that allows them to communicate some set of meanings. Such an agreement on the agents' vocabularies and individual grammars is achieved through a process of self-organisation of the linguistic interactions that take place among the agents in the population.

In the experiments described in this paper the agents interact with each other playing language games. A *language game* [9, 10], which is played by a pair of agents randomly chosen from the population, consists of the following actions:

1. The speaker chooses a formula (i.e., a meaning) from a given propositional language, 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 formula) which can be logically equivalent or not to the formula intended by the speaker. If the hearer cannot parse the sentence, the speaker communicates the formula it had in mind to the hearer, and the hearer adopts an association between the formula and the sentence used by the speaker[4].
3. Depending on the outcome of the language game both agents adjust their grammars in order to be more successful in future language games.

[4] A language game *succeeds* if the hearer can parse the sentence communicated by the speaker and it extracts a meaning (i.e., a formula) that is logically equivalent to the formula the speaker had in mind; otherwise the language game *fails*.

3.1 Invention

The agents in the population start with an empty lexicon and grammar. Therefore they cannot generate sentences for most meanings at the early stages of a simulation run. In order to allow language to get off the ground, they are allowed to invent new sentences for those meanings they cannot express using their lexicons and grammars in the first step of a language game.

The invention algorithm generates a sentence E for a propositional formula F as follows. If F is atomic, it invents a new word E[5]. If F is a formula constructed using a connective (it is of the form $\neg A$ or $A \otimes B$), it generates an expression for the connective and for each subformula of F using the agent's grammar if it can, or inventing a new one if it cannot, and it concatenates these expressions randomly in order to construct a sentence E for the whole meaning F.

As the agents play language games they learn associations between expressions and meanings, and induce linguistic knowledge from such associations in the form of grammatical rules and lexical entries. Once they can generate sentences for expressing a particular meaning using their own grammars, they select the sentence with the highest score and communicate that sentence to the hearer. The algorithm for computing the score of a sentence from the scores of the grammatical rules used in its generation is explained in detail later.

3.2 Adoption

In the second step of a language game 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, and checks whether its interpretation is right or wrong (i.e., it is logically equivalent to the meaning intended by the speaker) in the third step of the language game. However at the early stages of a simulation run the agents usually cannot parse the sentences communicated by the speakers, since they have no prior linguistic knowledge. In this case the speaker communicates the formula 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 $s(F, S) \rightarrow E, \{S \ is \ 0.01\}$ to its grammar[6].

At later stages of a simulation run it usually happens that the grammars and lexicons of speaker and hearer are not consistent, because each agent constructs its own grammar from the linguistic interactions in which it participates, and it is very unlikely that speaker and hearer share the same history of linguistic interactions unless the population consists only of these two agents. In this case the hearer may be able to parse the sentence generated by the speaker, but its interpretation of that sentence might be different from the meaning the speaker had in mind. The strategy used to coordinate the grammars of speaker and hearer when this happens is to decrement the score of the rules used by speaker

[5] New words are sequences of one to three letters randomly chosen from the alphabet.

[6] The score of the rules generated using invention, adoption or induction is initialized to 0.01.

and hearer in the processes of generation and parsing, respectively, and allow the hearer to adopt an association between the sentence and the meaning used by the speaker. Adoption however does not always take place in this case, because it is possible that the hearer knows the grammatical rules used by the speaker, but the scores of these rules are not higher than the scores of the rules it used for interpretation. The hearer adopts only an association between a sentence and a meaning if it cannot generate such an association using its lexicon and grammar.

3.3 Induction

Besides inventing and adopting associations between sentences and meanings, the agents can use some *induction mechanisms* to extract generalizations from the grammar rules they have learnt so far [11]. The induction mechanisms used in this paper are based on the rules for simplification and chunk in [12], although we have extended them so that they can be applied to grammar rules which have scores and which mark with a number the position of the connective in the sentence. We use the approach proposed in [13] for computing the scores of sentences and meanings from the scores of the rules used in their generation.

The induction rules are applied whenever the agents invent or adopt a new association 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 a new arithmetic expression of the form* $\{R \text{ is } E \cdot S \cdot 0.01\}$*, where* $C1$ *and* $C2$ *are constants in the range [0,1], and* E *is the product of the score variables that appeared on the right hand side of* $r1$*.*

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

$$s([and, light, right], S) \rightarrow andlightright, \{S \text{ is } 0.01\}. \qquad (10)$$

It could apply simplification to rule 10 (using rule 3) and replace it with 11.

$$s([and, light, R], S) \rightarrow andlight, s(R, SR), \{S \text{ is } SR \cdot 0.01\} \qquad (11)$$

Now rule 11 could be simplified again, replacing it with 12 which contains specific information about the position of the connective in the sentence.

$$s([and, Q, R], S) \rightarrow 1, and, s(Q, SQ), s(R, SR), \{S\ is\ SQ \cdot SR \cdot 0.01\} \quad (12)$$

If later on the agent invents or adopts a rule that associates the sentence 'orlightright' with the formula $[or, light, right]$ and applies simplification, then its grammar would contain the following rules that are compositional and recursive, but which do not use a syntactic category for binary connectives.

$$s([and, Q, R], S) \rightarrow 1, and, s(Q, SQ), s(R, SR), \{S\ is\ SQ \cdot SR \cdot 0.01\} \quad (13)$$
$$s([or, Q, R], S) \rightarrow 1, or, s(Q, SQ), s(R, SR), \{S\ is\ SQ \cdot SR \cdot 0.01\} \quad (14)$$

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, chunk can be applied as follows. A new category symbol c is created and the following new rules are added to the grammar.*

$$c(m1, S) \rightarrow e1, \{S\ is\ 0.01\} \qquad c(m2, S) \rightarrow e2, \{S\ is\ 0.01\}$$

Rules $r1$ and $r2$ are replaced by 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 $\{R\ is\ E \cdot C1\}$ on the right hand side of $r1$ is replaced with a new arithmetic expression of the form $\{R\ is\ E \cdot S \cdot 0.01\}$, 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 $r1$.

For example the agent of previous examples could apply chunk to rules 13 and 14 generating a new syntactic category $c2$ for binary connectives as follows.

$$s([P, Q, R], S) \rightarrow 1, c2(P, S1), s(Q, S2), s(R, S3), \{S\ is\ \mathrm{S1} \cdot \mathrm{S2} \cdot \mathrm{S3} \cdot 0.01\} \quad (15)$$
$$c2(\mathrm{and}, S) \rightarrow \mathrm{and}, \{S\ is\ 0.01\} \quad (16)$$
$$c2(\mathrm{or}, S) \rightarrow \mathrm{or}, \{S\ is\ 0.01\} \quad (17)$$

Rules 13 and 14 would be replaced with rule 15, which generalises them because it can be applied to formulas constructed using any binary connective, and rules 16 and 17, which state that the expressions *and* and *or* belong to $c2$ (the syntactic category of binary connectives[7]), would be added to the grammar.

[7] The syntactic category $c2$ is in fact more specific, as we shall see in section 4. It corresponds to binary connectives that are placed at the beginning of the sentence followed in first place by the expression associated with their first argument and in second place by the expression associated with their second argument.

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 exists a string of terminals* $e1$ *in* $r2$ *that corresponds to a nonterminal* $c(X, S)$ *in* $r1$*, then rule* $r2$ *can be replaced by a new rule of the form* $c(m1, S) \rightarrow e1, \{S\ is\ 0.01\}$*.*

Suppose the agent of previous examples adopts or invents the following rule.

$$s([\text{if}, light, right], S) \rightarrow \text{ifflightright}, \{S\ is\ 0.01\} \tag{18}$$

Simplification of rule 18 with rules 2 and 3 would replace rule 18 with 19.

$$s([\text{iff}, Q, R], S) \rightarrow 1, \text{iff}, s(Q, SQ), s(R, SR), \{S\ is\ SQ{\cdot}SR{\cdot}0.01\} \tag{19}$$

Then chunk II, applied to 19 and 15, would replace rule 19 with rule 20.

$$c2(\text{iff}, S) \rightarrow \text{iff}, \{S\ is\ 0.01\} \tag{20}$$

3.4 Co-Adaptation

Coordination of the grammars constructed by the individual agents is not a trivial task, because in order to understand each other the agents must use a common vocabulary and must order the constituents of compound sentences in sufficiently similar ways as to avoid ambiguous interpretations. Such an agreement on the agents' vocabularies and on their individual grammars is achieved through a process of self-organisation of the linguistic interactions that take place among the agents in the population.

It is necessary to coordinate the agents' grammars because different agents can invent different expressions for referring to the same propositional constants and connectives, and because the invention process uses a random order to concatenate the expressions associated with the components of a given meaning. Let us consider an example that illustrates the problem. Imagine that an agent has invented or adopted the following rules for expressing the meaning [if,light,right].

$$s([\text{if}, \text{light}, \text{right}], S) \rightarrow \textit{lightrightif}, \{S\ is\ 0.01\}$$
$$s([\text{if}, \text{light}, \text{right}], S) \rightarrow \textit{rightlightif}, \{S\ is\ 0.01\}$$

Simplification with rules 2 and 3 would replace them with the following rules which not only cannot be used for generating a syntactic category for implications (because they do not satisfy the preconditions of chunk), but that are in fact *incompatible* because they associate the same sentence with two meanings which are not logically equivalent (they reverse the direction of the implication).

$$S([\text{if}, X, Y], SC) \rightarrow 3, \text{if}, s(X, SX), s(Y, SY), \{SC\ is\ SX{\cdot}SY{\cdot}0.01\}$$
$$S([\text{if}, X, Y], SC) \rightarrow 3, \text{if}, s(Y, SY), s(X, SX), \{SC\ is\ SX{\cdot}SY{\cdot}0.01\}$$

The agent would be forced thus to make a choice between one of these rules in order to express implications in a consistent manner, and would try to choose the rule that is understood by most agents in the population.

Self-organisation principles help to coordinate the agents' grammars in such a way that they prefer to use the rules that are used more often by other agents [14–16]. Coordination in the experiments takes place at the third stage of a language game, when the speaker communicates the meaning it had in mind to the hearer. Depending on the outcome of a language game speaker and hearer take different actions. We have explained some of them already (invention and adoption), but they *co-adapt their grammars* as well adjusting the scores of their rules in order to be more successful in future language games.

We consider first the case in which the speaker can generate a sentence for the meaning using the rules in its grammar. If the speaker can generate several sentences for expressing that meaning, it chooses the sentence with the highest score. The rest of the sentences are called *competing sentences*.

The *score of a sentence* (or a *meaning*) generated using a grammar rule is computed using the arithmetic expression on the right hand side of that rule. Consider the generation of a sentence for expressing the meaning $[and, right, light]$ using the following rules.

$$s(\text{light}, S) \rightarrow \text{light}, \{S\ is\ 0.70\} \quad (21)$$

$$s(\text{right}, S) \rightarrow \text{right}, \{S\ is\ 0.25\} \quad (22)$$

$$c2(\text{and}, S) \rightarrow \text{and}, \{S\ is\ 0.50\} \quad (23)$$

$$s([P, Q, R], S) \rightarrow 1, c2(P, S1), s(Q, S2), s(R, S3), \{S\ is\ S1 \cdot S2 \cdot S3 \cdot 0.01\} \quad (24)$$

The score S of the sentence *andrightligth*, generated by rule 24, is computed multiplying the score of that rule (0.01) by the scores of the rules 23, 22 and 21 which generate the substrings of that sentence (0.50, 0.25 and 0.70, respectively). The *score of a grammar rule* is the last number of the arithmetic expression that appears on the right hand side of that rule.

Suppose the hearer can interpret the sentence communicated by the speaker. If the hearer can obtain several meanings for that sentence, the meaning with the highest score is selected. The rest of the meanings are called *competing meanings*.

If the meaning interpreted by the hearer is logically equivalent to the meaning the speaker had in mind, the game succeeds and both agents adjust the scores of the rules in their grammars. The speaker increases the scores of the rules it used for generating the sentence communicated to the hearer and decreases the scores of the rules it used for generating competing sentences. 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. This way the rules that have been used successfully get reinforced. The rules that have been used for generating competing sentences or meanings are inhibited.

If the meaning interpreted by the hearer is not logically equivalent to the meaning the speaker had in mind, the game fails, and both agents decrease the scores of the rules they used for generating and interpreting the sentence, respectively. This way the rules that have been used without success are inhibited.

The scores of grammar rules are *updated* using the scheme proposed in [9]. The rule's original score S is replaced with the result of evaluating expression

25 if the score is *increased*, and with the result of evaluating expression 26 if the score is *decreased.* The constant μ is a learning parameter which is set to 0.1.

$$minimum(1,\ S+\mu) \tag{25}$$

$$maximum(0,\ S-\mu) \tag{26}$$

A mechanism for **forgetting rules** that have not been useful in past language games is introduced to simplify the agents' grammars and avoid sources of ambiguity. Every ten language games the rules which have been used more than thirty times and have scores lower than 0.01 are removed from the grammars.

4 EXPERIMENTS

We describe the results of some experiments in which a population of five agents constructs a common vocabulary and a grammar that allows communicating a set of meanings which corresponds to all the formulas of a propositional logic language.

In the experiments we have taken an incremental learning approach in which the agents first play 10010 language games about propositional constants, and then they play 15010 language games about logical formulas constructed using unary or binary connectives. At the end of a typical simulation run all the agents prefer the same expressions for naming the propositional constants of the language. Table 1 describes the individual grammars built by the agents at the end of a particular simulation run. These grammars, although different, are compatible enough to allow total communicative success. That is, the agents always generate sentences that are correctly understood by the other agents.

It can be observed that all the agents have recursive rules for expressing formulas constructed with unary and binary connectives (see table 1). Agents a2 and a5 have invented a syntactic category for unary connectives. The other agents have specific rules for formulas constructed using negation, which use the same word 'f' preferred by the former agents for expressing negation. The grammar rules used for expressing negation place the word associated with the connective in the second position of the sentence. This is indicated by the number that appears in first place on the right hand side of a grammar rule. For example agent a1 would use the sentence 'ywf' to express the formula $\neg u$, assuming it associates the word 'yw' with the propositional constant u.

Thus the number 1 indicates that the connective is located in the first position of the sentence (it is a prefix), the number 2 that the connective is located in the second position (it is an infix), and the number 3 that the connective is located in the third position (it is a suffix). We use this convention in order to be able to represent two different types of grammar rules for expressing formulas constructed using unary connectives (which place the connective in the first and the second position of the sentence, respectively) and six different types of grammar rules for expressing formulas constructed using binary connectives[8].

[8] The induction rules (simplification and chunk) have been extended appropriately to deal with this convention.

This is so because a grammar rule for expressing formulas constructed using binary connectives must specify the position of the expression associated with the connective in the sentence, and the relative positions of the expressions associated with the arguments of the connective in the sentence.

Consider the second and fourth grammar rules of agent a4. Both rules place the expression associated with the connective in the third position of the sentence, but differ in the positions in which they place the expressions associated with the arguments of the connective. The second rule places the expression associated with the first argument of the connective (variable Y) in the first position of the sentence, the expression associated with the second argument (variable Z) in the second position, and the expression associated with the connective in the third position. The fourth rule places the expression associated with the second argument of the connective (variable Z) in the first position in the sentence, the expression associated with the first argument (variable Y) in the second position, and the expression associated with the connective in the third position[9].

When analyzing the grammar rules built by the agents we distinguish between commutative and non-commutative binary connectives. Because in order to communicate formulas constructed with commutative connectives, the agents only have to agree on a common vocabulary and on the position in which they place the expression associated with the connective in the sentence, since the order of the arguments does not modify the meaning of the sentence. We can observe in table 1 that in fact all agents place in the same position (third) of the sentence the connectives 'and', 'or' and 'iff', and that they use the same words ('dyp', 'yi' and 'iaj', respectively) for expressing them. But that they do not place in the same positions the expressions associated with the arguments of commutative connectives. For example, agents a1, a2 and a5 place the expression associated with first argument of the connective *and* in the second position of the sentence, while agents a3 and a4 place it in the first position of the sentence.

The positions in which the expressions associated with the arguments of non-commutative connectives are placed in a sentence determine however the meaning of the sentence. We can observe in table 1 that all agents use the word 'bqi' for expressing the connective 'if', that they all place it in the first position of the sentence, and that all of them place the expressions associated with the antecedent and the consequent of an implication in the same positions (second and third, respectively).

We can conclude then that the self-organisation process minimizes the learning effort of the agents by imposing only those order relations among the components of a sentence that are necessary for language understanding.

All agents have created syntactic categories for commutative connectives, although the extent of such categories differs from one agent to another depending on the positions in which they place the expressions associated with the arguments of the connectives 'and', 'or' and 'iff' in the sentence. Agents a1, a2 and a3 have invented syntactic categories for non-commutative connectives, whereas

[9] Observe the order in which the non-terminals s(Y,Q) and s(Z,R) appear on the right hand sides of both rules.

Grammar a1
s([not, Y], R) → 2, f, s(Y,Q), {R is Q · 1}
s([and, Y, Z], T) → 3, dyp, s(Z,Q), s(Y,R), {T is Q · R · 1}
s([X, Y, Z], T) → 3, c3(X,P), s(Y,Q), s(Z,R), {T is P · Q · R · 1}
c3(or, X) → yi, {X is 1}
c3(iff, X) → iaj, {X is 1}
s([X, Y, Z], T) → 1, c1(X,P), s(Y,Q), s(Z,R), {T is P · Q · R · 1}
c1(if, X) → bqi, {X is 1}
Grammar a2
s([X,Y],R) → 2, c1(X,P), s(Y,Q), {R is P · Q · 1}
c1(not, X) → f, {X is 1}
s([X, Y, Z], T) → 3, c2(X, P), s(Z,Q), s(Y,R), {T is P · Q · R · 1}
c2(and, X) → dyp, {X is 1}
c2(or, X) → yi, {X is 1}
c2(iff, X) → iaj, {X is 1}
s([X, Y, Z], T) → 1, c3(X,P), s(Y,Q), s(Z,R), {T is P · Q · R · 1}
c3(if, X) → bqi, {X is 1}
Grammar a3
s([not, Y], R) → 2, f, s(Y,Q), {R is Q · 1}
s([X, Y, Z], T) → 3, c1(X, P), s(Y,Q), s(Z,R), {T is P · Q · R · 1}
c1(and, X) → dyp, {X is 1}
c1(or, X) → yi, {X is 1}
c1(iff, X) → iaj, {X is 1}
s([X, Y, Z], T) → 1, c2(X,P), s(Y,Q), s(Z,R), {T is P · Q · R · 1}
c2(if, X) → bqi, {X is 1}
Grammar a4
s([not, Y], R) → 2, f, s(Y,Q), {R is Q · 1}
s([X, Y, Z], T) → 3, c4(X, P), s(Y,Q), s(Z,R), {T is P · Q · R · 1}
c4(and, X) → dyp, {X is 1}
s([X, Y, Z], T) → 3, c7(X, P), s(Z,R), s(Y,Q), {T is P · R · Q · 1}
c7(or, X) → yi, {X is 1}
c7(iff, X) → iaj, {X is 1}
s([if, Y, Z], T) → 1, bqi, s(Y,Q), s(Z,R), {T is Q · R · 1}
Grammar a5
s([X,Y],R) → 2, c1(X,P), s(Y,Q), {R is P · Q · 1}
c1(not, X) → f, {X is 1}
s([X, Y, Z], T) → 3, c4(X, P), s(Z,Q), s(Y,R), {T is P · Q · R · 1}
c4(and, X) → dyp, {X is 1}
c4(or, X) → yi, {X is 1}
s([X, Y, Z], T) → 3, c2(X, P), s(Y,Q), s(Z,R), {T is P · Q · R · 1}
c2(iff, X) → iaj, {X is 1}
s([if, Y, Z], T) → 1, bqi, s(Y,Q), s(Z,R), {T is Q · R · 1}

Table 1. Grammars constructed by the agents in a particular simulation run.

agents a4 and a5 have specific grammar rules for expressing implications. There are no alternative words for any connective in the agents' grammars, because the mechanism for forgetting rules that have not been useful in past language games has probably removed such words from the grammars.

Figure 1 shows the evolution of the communicative success, averaged over ten simulation runs with different initial random seeds, for a population of five agents. The *communicative success* is the average of successful language games in the last ten language games played by the agents. We can observe that the agents reach a communicative success of 100% in 20800 language games. That is, after each agent has played on average 4160 language games.

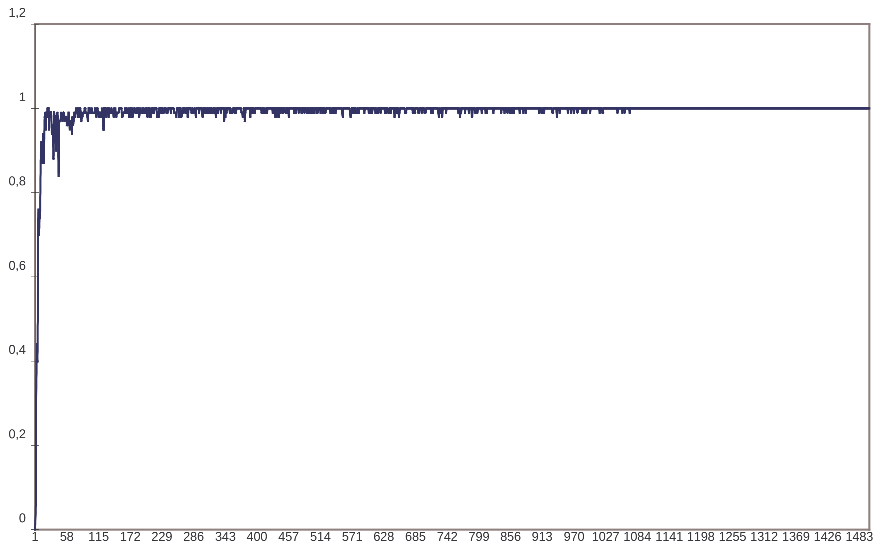

Fig. 1. Evolution of the communicative success in experiments performed using a population of five agents, 10010 language games about propositional constants (not shown), and 15010 language games about formulas of constructed using logical connectives.

5 RELATED WORK

The emergence of recursive communication systems in populations of autonomous agents which have no prior linguistic knowledge has been studied by other authors. The research presented in [15] addresses the problem of the emergence of recursive communication systems in populations of autonomous agents, as we do. It differs from the work described in the present paper by focusing on learning exemplars rather than grammar rules. These exemplars have costs, as our grammar rules do, and their costs are reinforced and discouraged using self-organization principles as well. The main challenge for the agents in the experiments described in [15] is to construct a communication system that is capable of naming atomic formulas and, more importantly, marking the identity relations among the arguments of the different atomic formulas that constitute the meaning of a given string of characters. This task is quite different from the learning task proposed in this paper which focusses on categorizing propositional sentences and connectives, and marking the scope of each connective using the order of the constituents of a string of characters.

The most important difference between our work and that presented in [12] is that the latter focusses on language transmission over generations. Rather than studying the emergence of recursive communication systems in a single population of agents, as we do, it shows that the bottleneck established by language transmission over several generations favors the propagation of compositional and recursive rules because of their compactness and generality. In the experiments described in [12] the population consists of a single agent of a generation that acts as a teacher and another agent of the following generation that acts as a learner. There is no negotiation process involved, because the learner never has the opportunity to act as a speaker in a single iteration. We consider however populations of five agents which can act both as speakers and hearers during the simulations. Having more than two agents ensures that the interaction histories of the agents are different from each other, in such a way that they have to negotiate in order to reach agreements on how to name and order the constituents of a sentence.

The induction mechanisms used in the present paper are based on the rules for chunk and simplification in [12], although we have extended them so that they can be applied to grammar rules which have scores and which mark with a number the position of the connective in the sentence. Finally the meaning space used in [12] (a restricted form of atomic formulas of second order logic) is different from the meaning space considered in the present paper (arbitrary formulas from a propositional logic language), although both of them require the use of recursion.

6 CONCLUSIONS

We have described some experiments which show how a *language* expressive enough to allow the communication of meanings of the same complexity as propositional logic formulas can emerge in a population of autonomous agents which have no prior linguistic knowledge. This language although simple has interesting properties found in natural languages, such as recursion, syntactic categories for propositional sentences and connectives, and partial word order for marking the scope of each connective.

An approach based on *general purpose cognitive capacities,* such as invention, adoption and induction, and on *self-organisation* principles applied to a particular type of linguistic interaction known as a *language game* has been taken.

These experiments extend previous work by considering a larger population and a much larger search space of grammar rules. In particular the agents are allowed to order the expressions associated with the constituents of a logical formula in arbitrary order in the sentence. Previous work assumed that the expressions associated with the connectives should be always placed in the first position of the sentence. The branching factor of the search space of grammar rules that can be used for expressing formulas constructed with binary connectives has been extended thus from two to six.

Another difference is that communication is considered successful in a language game if the meaning interpreted by the hearer is *logically equivalent* to the meaning the speaker had in mind. In previous experiments [1] both meanings were required to be syntactically equal, i.e., the same formula. This has allowed us to observe how a less strict grammar in terms of word order emerges through the self-organisation process, which minimizes the learning effort of the agents by imposing only those order relations among the components of a sentence that are necessary for language understanding.

References

1. Sierra, J.: Propositional logic syntax acquisition. In: Symbol Grounding and Beyond, Lecture Notes in Computer Science, volume 4211. (2006) 128–142
2. McCarthy, J.: Formalizing Common Sense. Papers by John McCarthy. Ablex. Edited by Vladimir Lifschitz (1990)
3. Piaget, J.: The Equilibration of Cognitive Structures: the Central Problem of Intellectual Development. University of Chicago Press, Chicago (1985)
4. Santibáñez, J.: Relación del rendimiento escolar en las áreas de lectura y escritura con las aptitudes mentales y el desarrollo visomotor. Universidad Nacional de Educación a Distancia, D.L., ISBN 84-398-2486-6, Madrid (1984)
5. Santibáñez, J.: Variables psicopedagógicas relacionadas con el rendimiento en E.G.B. Instituto de Estudios Riojanos, ISBN 84-87252-00-1, Logroño (1988)
6. Sierra, J.: Grounded models as a basis for intuitive reasoning. In: Proceedings of the International Joint Conference on Artificial Intelligence. (2001) 401–406
7. Sierra, J.: Grounded models as a basis for intuitive and deductive reasoning: The acquisition of logical categories. In: Proceedings of the European Conference on Artificial Intelligence. (2002) 93–97
8. Sierra, J.: Grounded models as a basis for intuitive reasoning: the origins of logical categories. In: Papers from AAAI–2001 Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems. Technical Report FS-01-01, AAAI Press. (2001) 101–108
9. Steels, L.: The Talking Heads Experiment. Volume 1. Words and Meanings. Special Pre-edition for LABORATORIUM, Antwerpen (1999)
10. Steels, L., Kaplan, F., McIntyre, A., V Looveren, J.: Crucial factors in the origins of word-meaning. In: The Transition to Language, Oxford Univ Press (2002) 252–271
11. Steels, L.: Macro-operators for the emergence of construction grammars. SONY CSL (2004)
12. Kirby, S.: Learning, bottlenecks and the evolution of recursive syntax. In: Linguistic Evolution through Language Acquisition: Formal and Computational Models, Cambridge University Press (2002) 96–109

13. Vogt, P.: The emergence of compositional structures in perceptually grounded language games. Artificial Intelligence **167(1-2)** (2005) 206–242
14. Steels, L.: The synthetic modeling of language origins. Evolution of Communication **1(1)** (1997) 1–35
15. Batali, J.: The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In: Linguistic Evolution through Language Acquisition: Formal and Computational Models, Cambridge U.P. (2002) 111–172
16. Steels, L.: Constructivist development of grounded construction grammars. In: Proc. Annual Meeting of Association for Computational Linguistics. (2004) 9–16