EXPRESSING SECOND ORDER SEMANTICS AND THE EMERGENCE OF RECURSION

JORIS BLEYS

Artificial Intelligence Laboratory, Vrije Universiteit Brussel,
Pleinlaan 2, Brussels, 1050, Belgium
jorisb@arti.vub.ac.be

Although most previous model-based research has not moved beyond first-order semantics, human languages are clearly capable of expressing second-order semantics: the meanings expressed in a sentence do not only consist of conjunctions of first-order predicates but also predicates that take other predicates as an argument. In this paper we report on multi-agent language game experiments in which agents handle second-order semantics. We focus our discussion on how this type of research is able to provide fundamental insights in how properties of human-language-like properties could once have emerged. For recursion, this might have happened as a side-effect of agents trying to reuse previously learned language structure as much as possible.

1. Introduction

Although research on the emergence of communication systems with similar features as human natural language has shown important progress, the complexity of the meanings considered so far remains limited. Experiments either use simple categories (Steels & Belpaeme, 2005), conjunctive combinations of categories (Wellens, 2008) or predicate-argument expressions (Batali, 2002; De Beule, 2008). Natural languages are clearly capable of expressing second order semantics (Dowty, Wall, & Peters, 1981). For example, the adverb “very” in “very big” modifies the meaning of the adjective, it is not a simple conjunction of the predicates ‘very’ and ‘big’. Moreover the same predicate (e.g. ‘big’) can often be used in different ways, for example to further restrict the set of possible referents of a noun (as in “the big ball”), to state a property of an object (as in “the ball is big”), to reify the predicate itself and make a statement about it (as in “big says something about size”), to compare the elements of a set (as in “this ball is bigger than the others”), etc. The specific usage of a predicate in a particular utterance is clearly conveyed by the grammar, so any theory for the origins and evolution of grammar must address second order semantics.

The present paper reports progress on how a communication system could emerge to express second order semantics, building further on previously presented research (Steels, 2000; Steels & Bleys, 2005, 2007) in which we adopted the language game paradigm. In the current paper we do not consider the problem how second order semantics could emerge (discussed in more detail by Van den Broeck (2008)), but rather focus on the question whether recursive structure could arise and how.

2. Grounded semantic constraint networks

The semantics of the utterances in the following experiment are not represented in a standard logic, but in an alternative formalism, Incremental Recruitment Language (IRL). In this view the meaning of a sentence is a *semantic constraint network* that the speaker wants the hearer to resolve in order to achieve the communicative goal selected by the speaker. The basic nodes of these networks are *primitive constraints* which are provided by the experimenter. Each primitive constraint has a number of arguments which can be bound to a certain variable. Variables are denoted using a question mark prefix. If a variable appears as an argument to more than one constraint, it means the value for this variable is constrained by more than one constraint.

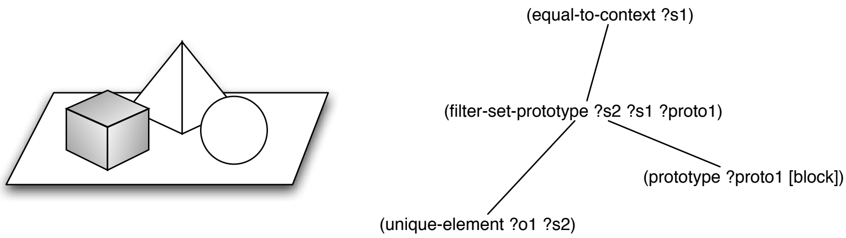

Figure 1. On the left: a hypothetical world. On the right: a valid semantic constraint network to identify the topic (marked in grey for clarity).

In production the speaker has to find a semantic constraint network that is suitable to achieve the communicative goal (e.g. identifying the topic) it selected. An example of such a network, shown on the right in Figure 1, is able to identify the block in the world depicted on the left. The more complex the world (for example by adding a second block), the more complex the semantic constraint network will need to be in order to achieve this goal (for example by adding another filtering operation). This network needs to be encoded in a serial utterance which has to be decoded by the hearer in such a way that when it runs the constraint propagation algorithm over the network, it is able to achieve the communicative goal of the speaker. More details, for instance on the exact inner workings of any primitive constraints reported in this paper, or on how semantic constraint networks are constructed, can be found in Steels (2000), Steels and Bleys (2005) and Van den Broeck (2008).

3. Mapping semantic constraint networks onto language

The next question we have to answer is how the agents are supposed to encode such a semantic constraint network over a serial interface. We use Fluid Construction Grammar (FCG) as our substrate for this mapping. FCG is a computational formalism inspired by the general theory of construction grammar which states that each linguistic rule should be a pairing of syntax and semantics (Goldberg, 2003). In the experiments we report in this paper, the semantics of a rule consist of different parts of the semantic constraint network and/or the variable equalities between these parts. The syntactic side is governed by syntactic categories and/or word-order constraints and/or simple forms (words).

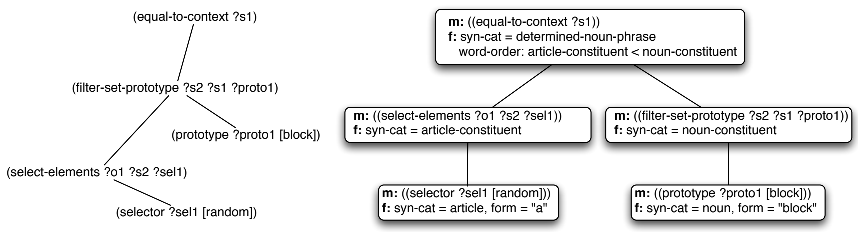

Figure 2. On the left: an example of semantic constraint network. On the right: a complete production/parse tree to encode/decode this semantic constraint network. Each unit contains information on both semantics (m) and syntax (f). The bottom layer contains the semantic entities (entity units), the middle layer contains the primitive constraints that use these entities directly (functional units) and the top layer contains all other operations (contextual unit).

As shown in Figure 2 our basic approach is to divide the semantic constraint network in three layers of units[a]: (a) entity units containing the semantic entities[b], (b) the functional units which make direct use of such a semantic entity and (c) contextual units which contain any remaining operations of the semantic constraint network that do not make direct use of any semantic entity. This program would be useful in a world in which there is more than one block, but the goal is to identify any block.

As a rule of thumb, the reader can assume that each rule introduces one new unit in the production/parse tree. In production, syntactic information is added to the production tree: which words will be used to express certain semantic entities, to which syntactic category does each unit belong and which word order should be applied when this tree is transformed into an utterance. During interpretation, each word in the utterance introduces a new semantic entity. Based on the lexical

[a]Ideally, the agents should come up with this division themselves as they try to reuse as much linguistic knowledge as possible. This is part of our future research agenda.

[b]At this moment the agents assume that each semantic entity is captured in exactly one unit.

categories of these entities, the hearer is able to add the layer of functional units. Finally, the information on the word order constraints augmented with the information of the syntactic categories allows the agent to select the right contextual rule which adds extra primitive constraints to the network, but more importantly also connects all primitive constraints by introducing variable equalities (for example between the first argument of FILTER-SET-PROTOTYPE and the second argument of SELECT-ELEMENTS) in the network shown in Figure 2. We have devised learning operators allowing both speaker and hearer to learn these divisions which are reported in more detail in Steels and Bleys (2007).

For the sake of clarity we choose to focus on the construction of the syntactic category system and we abstract away implementation details which might distract the reader from the main hypothesis we are proposing, namely that recursive rules might emerge as a side-effect of agents trying to reuse as much of their previously constructed language knowledge as possible.

4. Three steps towards the emergence of recursive rules

The construction of the system of syntactic categories is fairly simple: the agents try to reuse any syntactic category which would allow the reuse of a previously learned rule[c]. If no such syntactic category is found, the agent decides to construct a new syntactic category. This basic mechanism results in a one-on-one mapping between syntactic and semantic categories at the lexical level, but at all other levels syntactic categories are only invented when needed and one cannot easily reconstruct a similar mapping unless one takes into account the linguistic development of each agent.

4.1. *Starting from scratch*

The first time an agent has to express/interpret a semantic constraint network (similar to the one depicted in Figure 1, which could represent the semantics of a sentence like "ball") it has no syntactic categories and hence it needs to invent two new syntactic categories. One specifies the syntactic association between the entity unit and the functional unit (e.g. noun), and the other one specifies the association between the functional unit and the contextual unit (e.g. noun-constituent). This kind of process is schematically shown on the left hand side of Figure 3.

Let's suppose the agent now has to express/interpret a variation of this semantic constraint network in which the semantic entity is a prototype of a pyramid instead of one of a block. This provides a first opportunity for the agents to reuse a syntactic category because if the syntactic category of the entity unit for the pyramid would be identical to the one of the entity unit of the block (e.g. noun),

[c]Technically a syntactic category in a rule is first a variable which will get a binding if any other rule requires a specific syntactic category using the unification engine of FCG. This variable will be replaced by the value of this binding before it is added to the actual rule-set of the agent.

Figure 3. On the left: invention of a new syntactic categories A, B and C. On the right: reuse of a syntactic category (A) as expected by the rule (B $\rightarrow$ A).

it would allow reuse of all the other syntactic categories (and rules) it constructed for the previous semantic constraint network. This process is schematised on the right hand side of Figure 3 and typically occurs at the level of syntactic categories linking entity units and functional units.

4.2. *Substituting a primitive constraint*

Let's now consider a semantic constraint network in which a primitive constraint that does not take a semantic entity as direct argument (e.g. UNIQUE-ELEMENT) is substituted by one that does so (e.g. SELECT-ELEMENTS) as illustrated in Figure 2. Let's suppose this network corresponds to the semantics of a sentence like "a ball". The contextual rule of the previous example is now useless as it contains a primitive constraint, namely UNIQUE-ELEMENT, that is not even part of the semantic constraint network at hand. The agents have to invent a new contextual rule, but not all hope is lost, because they can reuse every other previously introduced category (and rule) if they incorporate the syntactic category they previously used to associate the functional unit with the contextual unit (e.g noun-constituent). This process is shown in Figure 4, and typically occurs at the level of syntactic categories linking functional units and contextual units.

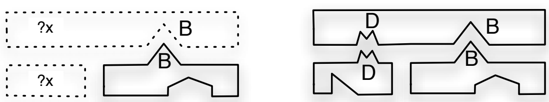

Figure 4. Invention of a new syntactic category (D) while reusing a previously learned syntactic category (B).

4.3. *Adding a primitive constraint*

The final semantic constraint network we have to consider is one that is achieved by starting from the one we introduced in Figure 2 and adding an extra primitive constraint, FILTER-SET-CATEGORY, in between two existing ones, namely

FILTER-SET-PROTOTYPE and SELECT-ELEMENTS, which could represent the semantics of a sentence like “a big ball”. Using the same learning strategy as introduced in Section 4.2 the agents could learn a new contextual rule which combines three subunits into one new unit as shown in the middle of Figure 5.

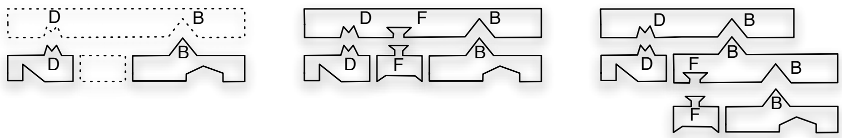

Figure 5. To the left and middle: reuse of two previously known syntactic categories (B and D) and invention of a new one (F) in a similar fashion as in Section 4.2. To the right: another solution which additionally is capable of reusing the contextual rule introduced in Section 4.2 by adding a truly recursive rule (B $\longrightarrow$ FB).

But the agents can do better by exploiting another learning strategy which allows agents to combine any number of units into one unit[d]. In the example shown on the right of Figure 5, the agents are able to come up with a rule that allows them to reuse the contextual rule introduced in Section 4.2. This particular rule combines two units, one belonging to syntactic category F (e.g. adjective-constituent) and the other to B (e.g. noun-constituent). The syntactic category to which this new combination unit should belong, determined by the deduction mechanism introduced at the beginning of Section 4, is syntactic category B (e.g. noun-constituent), as any other category would block the reuse of the contextual rule. As this syntactic category is equal to one of the rule’s constituents, this new rule is truly recursive.

5. Multi-agent simulation

We briefly introduce the results of a multi-agent simulation in which our hypothesis is implemented in Figure 6. The complexity of the semantic constraint networks increases over time (depicted by the learning stage). Each increase in complexity introduces a period of stress in the communication system that is resolved (as shown by the communicative success[e]). Invention and agreement on the level of the word-semantic entity associations, and on the level of specific word order constrains, is shown in the overshoot and stabilisation of the lexicon size and the number of functional and contextual rules, respectively.

[d]At a semantic level, the rule introducing this unit should also take care of the necessary variable equalities between the primitive constraint in the two subunits it combines.

[e]Communicative success is defined as a game in which the hearer was able to achieve the communicative goal selected by the speaker.

The most important observation lies in the transitions in where the complexity of the semantic constraint network is increased but the language does not need to be expanded in order to deal with this extra complexity. The transition to learning stage 4 (between 7k and 8k games) in which an extra filtering operation is added to the semantic constraint networks the agents have to express, exhibits this phenomenon.

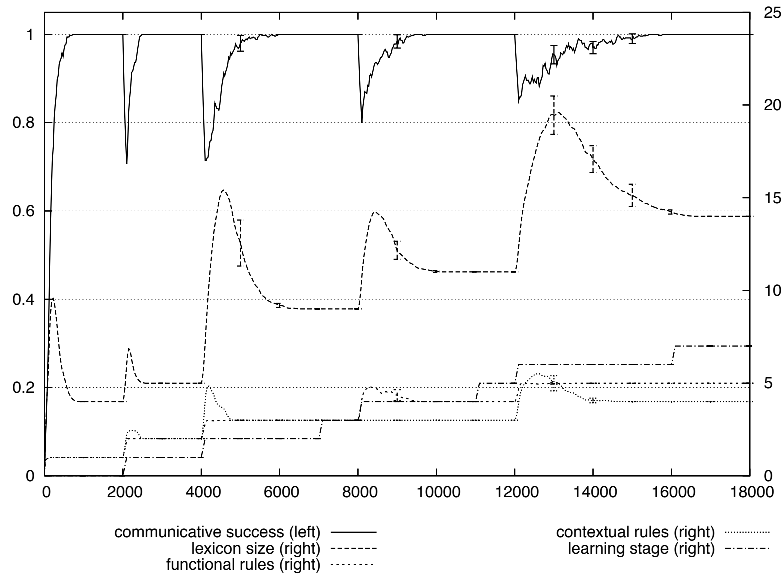

Figure 6. Graph showing the basic measures for our multi-agent simulation. Bottom axis shows number of interactions; left axis shows scores between 0 and 1; right axis shows number of rules.

6. Conclusion

We presented our main hypothesis, namely that hierarchical rules can become recursive as a side-effect of language users who try to reuse as much of their previously gained linguistic knowledge as possible. We supported the plausibility of our hypothesis by showing a clear analysis of how hierarchical rules can become recursive and by showing the results of a multi-agent experiment, which demonstrated that the rules learned by a population of agents are truly recursive.

Although our simulation results depend on many other factors, like for instance the increase of semantic complexity and the specific structure of the semantic constraint networks we have used, we have provided a proof of concept of our main hypothesis. We have shown that it is adequate for the emergence of a recursive syntactic category system for hierarchical rules.

Acknowledgements

I am indebted to my supervisor, professor Luc Steels, as he provided the main idea and hypothesis put forward in this paper. I am also grateful to all members of the Artificial Intelligence Laboratory at the Vrije Universiteit Brussel and the Sony Computer Science Laboratory in Paris for their insightful discussions and comments. I would like to thank the anonymous referees for their thoughtful reviews. Joris Bleys is financed through a fellowship of the Institute of the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

Batali, J. (2002). The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In T. Briscoe (Ed.), *Linguistic evolution through language acquisition: Formal and computational models.* Cambridge University Press.

De Beule, J. (2008). The emergence of compositionality, hierarchy and recursion in peer-to-peer interactions. In A. D. M. Smith, K. Smith, & R. F. i Chancho (Eds.), *The evolution of language: Evolang 7.* World Scientific.

Dowty, D. R., Wall, R., & Peters, S. (1981). *Introduction to Montague semantics.* D. Reidel Publishing Company.

Goldberg, A. E. (2003). Constructions: a new theoretical approach to language. *Trends in Cognitive Sciences*, *7*(5), 219–224.

Steels, L. (2000). The emergence of grammar in communicating autonomous robotic agents. In W. Horn (Ed.), *ECAI2000* (pp. 764–769). Amsterdam: IOS Press.

Steels, L., & Belpaeme, T. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, *28*, 469–529.

Steels, L., & Bleys, J. (2005). Planning what to say: Second order semantics for Fluid Construction Grammars. In A. B. Diz & J. S. Reyes (Eds.), *Proceedings of CAEPIA '05. Lecture Notes in AI.* Berlin: Springer Verlag.

Steels, L., & Bleys, J. (2007). Emergence of hierarchy in Fluid Construction Grammar. In *Proceedings of the SLEA workhop at the 9th European conference on Artificial Life.*

Van den Broeck, W. (2008). Constraint-based compositional semantics. In A. D. M. Smith, K. Smith, & R. F. i Chancho (Eds.), *The evolution of language: Evolang 7.* World Scientific.

Wellens, P. (2008). Coping with combinatorial uncertainty in word learning: A flexible usage-based model. In A. D. M. Smith, K. Smith, & R. F. i Chancho (Eds.), *The evolution of language: Evolang 7.* World Scientific.