ON THE ORIGINS OF RECURSIVE RULES: A USAGE BASED ACCOUNT

JORIS BLEYS

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

Recursion is considered to be one of the hallmarks of human communication. Many theories have been proposed on how this important feature might once have originated. This paper critically examines previously proposed models and posits a new and clearly defined hypothesis: recursion might originate from language users who try to reuse as much of their previously gained linguistic knowledge as possible. We support this claim by providing results of a multi-agent computer simulation in which the agents invent their own communication system encompassing a recursive syntactic category system.

1. Introduction

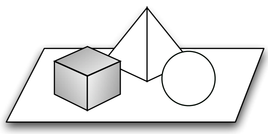
Recursion is widely considered to be one of the hallmarks of human language. Being able to deal with it was once highlighted as the unique capacity that allows humans to maintain such a complex communication system that human language is (Hauser, Chomsky, & Fitch, 2002). Many theories have been devised on how this important feature has come into existence in the evolution of language.

Some researches believe that it became part of an innate Language Acquisition Device (Pinker & Bloom, 1990), while others shift the burden of the origins of recursion to language itself, in particular by focussing on how languages can be learned. Kirby (2002) uses the *iterated learning model* to show the emergence of recursion. In short, the agents in this model try to infer grammatical rules generated by another agent which has to express a subset of a predefined range of meanings. After this process the learner becomes a teacher and a virgin agent takes the role of learner and this learning process is repeated many times. Their model shows that a recursive grammar can emerge, but only if the meanings that the agents have to express themselves are recursive. This shifts the problem of the origins of recursive language to the origins of recursive meanings. Batali (2002) reports on another computational model in which recursion emerges as a result of the interplay between different learning operators and in which the meaning is not recursive in itself. Batali provides an analysis of a communication system that is recursive, but does not provide a clear analysis of how and why this recursion emerges.

The hypothesis we present in this paper is that recursive rules might emerge as a side-effect of agents which try to reuse as much as possible of previously established linguistic knowledge. This fits nicely in our general hypothesis that agents should try to optimise their communicative success while reducing their cognitive effort (Steels, 2006). Reusing previously established linguistic knowledge achieves both: it results in a higher expected communicative success than any new piece of language knowledge invented by a speaker and it decreases the number of rules the agents have to remember and consider during processing. From a population point of view it also limits the number of rules the agents have to reach a consensus about, so it helps the agents in bootstrapping their own language.

2. Grounded semantic networks

The semantics of the utterance in the experiments reported in this paper are not represented in a standard logic, but in an alternative formalism, Incremental Recruitment Language (IRL), which is similar to work on procedural semantics pioneered by Winograd (1972). In this view the meaning of a sentence is a *semantic 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 *semantic operations* which are provided by the experimenter. Each semantic operation has a number of arguments which can be bound to a certain variable. Variables are denoted using a question mark as prefix. If a variable appears as an argument to more than one operation, it means the value for this variable is constrained by more than one operation.



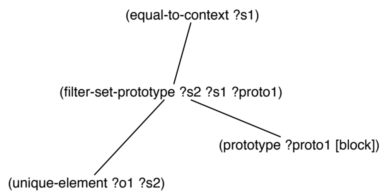


Figure 1. On the left: a hypothetical world. On the right: a valid semantic network to identify the topic (marked in grey for clarity).

Figure 1 shows a valid semantic network on the right to identify the block in the world depicted on the left. The network consists of three operations: (a) EQUAL-TO-CONTEXT, (b) FILTER-SET-PROTOTYPE and (c) UNIQUE-ELEMENT and one *semantic entity*. Operation (a) binds all objects in the current context to its first variable ?s1. Operation (b) filters the contents of this set based on a similarity function using the semantic entity bound to its third argument, in this case the prototype of a block, to only retain those that are similar to that prototype. Variable ?s2 thus is a singleton containing only the internal representation of the block. Finally, operation (c) expects the value bound to its second argument to be a singleton and if so binds the unique element of the variable bound to its first argument ?o1.

In production the speaker has to find a semantic network that is suitable to achieve the communicative goal (e.g. identifying the topic) it selected. The more complex the world (for example by adding a second block), the more complex the semantic 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 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 how semantic networks are constructed, can be found in Steels (2000) and Steels and Bleys (2005).

3. Mapping semantic networks onto language

The next question we have to answer is how the agents are supposed to encode such a semantic network over a serial interface. We use Fluid Construction Grammar (FCG) as our substrate for this mapping (Steels & De Beule, 2006). 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 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).

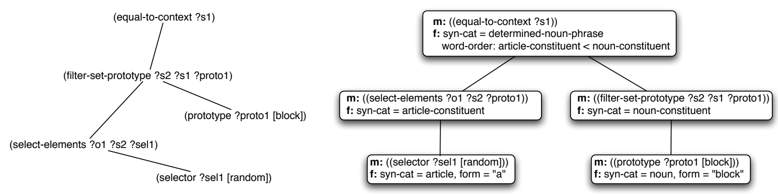


Figure 2. On the left: an example of semantic network. On the right: a complete production/parse tree to encode/decode this semantic network. Each unit contains information on both semantics (m) and form (f). The bottom layer contains the semantic entities, the middle layer contains the semantic operations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

As shown in Figure 2 our basic approach is to divide the semantic network in three layers of units[a]: (a) units containing the semantic entities[b], (b) the func-

[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.

tional units which make direct use of such a semantic entity and (c) contextual units which contain any remaining operations of the semantic network that do not make direct use of any semantic entity. Attentive readers also might have noticed the introduction of a new semantic operation: SELECT-ELEMENTS. This operation constrains, if the value of the third argument is equal to the semantic entity [random], the values of its arguments in such way that the value for ?o1 is a random element selected from ?s2. 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 ordering 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 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 semantic operations to the network but more importantly also connects all semantic operations by introducing variable equalities (for example between the first argument of FILTER-SET-PROTOTYPE and the second argument of SELECT-ELEMENTS) in the network. 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 sake of clarity we choose to focus on the construction of the syntactic category system and we abstract from any 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 optimise their communicative success while minimising cognitive effort.

4. Three steps to the emergence of recursive rules

The construction of the system of syntactic categories is fairly simple: the agents try to reuse any syntactic category available[c]. If no such syntactic category is found, the agent decides to construct a new syntactic category.

4.1. *Starting from scratch*

The first time an agent has to express/interpret a semantic network (similar to the one depicted in Figure 1) it has no syntactic categories and hence it has to invent new syntactic categories. One specifies the syntactical association between the

[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.

entity unit and the functional unit, and the other specifies the association between the functional unit and the contextual unit. This kind of process is shown on the left hand side of Figure 3.

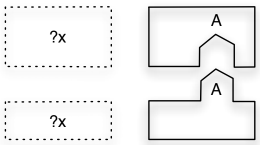


Figure 3. On the left: invention of a new syntactic category. There exists no super- or sub-rule which specifies the syntactic category of any of the rules. Hence a new syntactic category (A) is invented. On the right: reuse of a syntactic category (A) specified by the super-rule.

Let's suppose the agent now has to express/interpret a variation of this semantic 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, it would allow to reuse all other syntactic categories (and rules) it constructed for the previous semantic network. This process is schematised on the right hand side of Figure 3.

4.2. *Substituting a semantic operation*

Let's now consider a semantic network in which a semantic operation which does not take a semantic entity as direct argument is substituted by one that does so (as illustrated in Figure 2). The contextual rule of the previous example is now useless as it contains a semantic operation that is not even part of the semantic 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 rules) if they incorporate the syntactic category they previously used to associate the functional unit with the contextual unit. This process is shown in Figure 4.

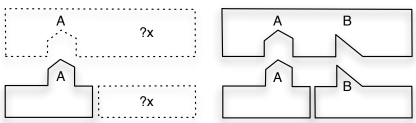


Figure 4. Reuse of one previously known syntactic category (A) and invention of a new one (B). This type of reuse typically occurs in contextual rules (which introduce context units).

4.3. *Adding a semantic operation*

The final semantic network we have to consider is one that is achieved by starting from the one we introduced in Figure 2 and adding an extra semantic operation in between two existing ones[d]. Using the same strategy as introduced in Section 4.2 the agents could learn a new contextual rule which combines three subunits into one new contextual rule as shown in the middle of Figure 5.

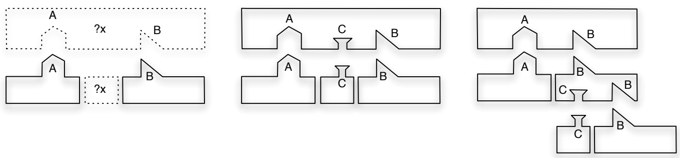


Figure 5. To the left and middle: reuse of two previously known syntactic categories (A and B) and invention of a new one (C) in a similar fashion as in Section 4.2. To the right: another solution which is capable of reusing a contextual rule by adding a truly recursive rule (B $\rightarrow$ CB).

But the agents can do better. By adding another learning strategy which allows agents to combine any number of units into one unit, the agents are able to come up with a rule that allows them to reuse the contextual rule introduced in Section 4.2 (AB). By using exactly the same mechanism to deduce the syntactic category of this new unit as introduced before, the agents decide that it should be of the same category as was used in the previously learned association between the functional unit and the contextual unit (B). As this contextual unit now is a sub-unit of this new unit, the resulting 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 networks increases over time (depicted by the learning stage). Each increase in complexity introduces a period of stress in the communication system which is resolved (as shown by the communicative success). 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.

The most important observation lies in the transitions in which the complexity of the semantic network is increased but the language does not need to be ex-

[d]In this example a FILTER-SET-CATEGORY operation is added in between the FILTER-SET-PROTOTYPE and SELECT-ELEMENTS operations, which acts similar to the former but instead of filtering using a prototype it filters using one channel only. More information on this semantic network and operation can be found in Steels and Bleys (2007).

panded 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 networks the agents have to express, exhibits this phenomenon.

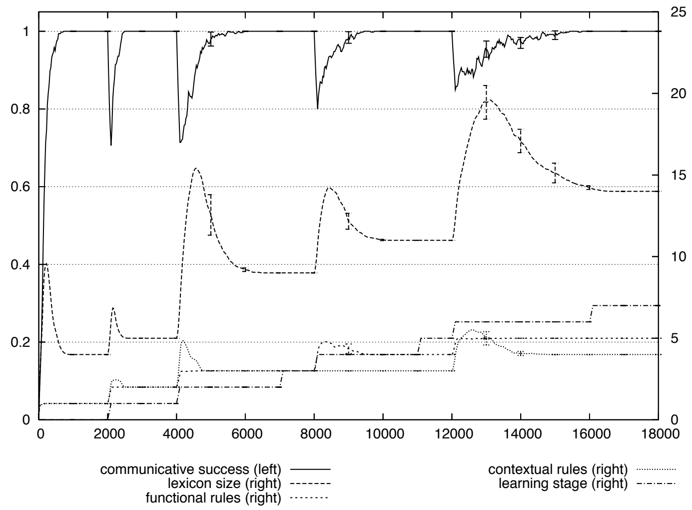


Figure 6. Graph showing the basic measures for our multi-agent simulation. Bottom axis shows number of interactions; left axis shows scores between 0 and 1; right axis shows number of rules.

6. Conclusion

We are able to provide an account for the origins of recursive rules which is completely governed by means of two basic principles: agents trying to optimise their communicative success while minimising the cognitive effort needed to achieve this success. We were able to show the validity of our claim in a multi-agent simulation, which clearly shows that we neither need to resort to the assumption that meaning itself is recursive nor to a specific genetic component. This result also completes other research in which was shown why agents could prefer recursive rules over non-recursive rules, but in which the true origin of a recursive rule was based on an assumption shared by all agents (De Beule, 2007).

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. 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. (2007). *Compositionality, hierarchy and recursion in language.* Unpublished doctoral dissertation, Vrije Universiteit Brussel.

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

Hauser, M. D., Chomsky, N., & Fitch, W. T. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, *298*, 1569-1579.

Kirby, S. (2002). Learning, bottlenecks and the evolution of recursive syntax. In T. Briscoe (Ed.), *Linguistic evolution through language acquisition: Formal and computational models.* Cambridge University Press.

Pinker, S., & Bloom, P. (1990). Natural language and natural selection. *Behavioral and Brain Sciences*, *13*(4), 707-784.

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

Steels, L. (2006). How to do experiments in artificial language evolution and why. In *Proceedings of the 6th international conference on the evolution of language* (p. 323-332).

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 social learning in embodied agents workhop at the 9th european conference on artificial life.*

Steels, L., & De Beule, J. (2006). A (very) brief introduction to Fluid Construction Grammar. In *Proceedings of the 3rd international workshop on scalable natural language (ScaNaLu06).*

Winograd, T. (1972). *Understanding natural language.* NY: Academic Press.