How grammar emerges to dampen combinatorial search in parsing

Luc Steels[1,2], Pieter Wellens[2]

[1] SONY Computer Science Laboratory - Paris
[2] Vrije Universiteit Brussel, Artificial Intelligence Laboratory
Pleinlaan 2, B-1050 Brussel

Abstract. According to the functional approach to language evolution (inspired by cognitive linguistics and construction grammar), grammar arises to deal with issues in communication among autonomous agents, particularly maximisation of communicative success and expressive power and minimisation of cognitive effort. Experiments in the emergence of grammar should hence start from a simulation of communicative exchanges between embodied agents, and then show how a particular issue that arises can be solved or partially solved by introducing more grammar. This paper shows a case study of this approach, focusing on the issue of search during parsing. Multiple hypotheses arise in parsing when the same syntactic pattern can be used for multiple purposes or when one syntactic pattern partly overlaps with another one. It is well known that syntactic ambiguity rapidly leads to combinatorial explosions and hence an increase in memory use and processing power, possibly to a point where the sentence can no longer be handled. Additional grammar, such as syntactic or semantic subcategorisation or word order and agreement constraints can help to dampen search because it provides information to the hearer which hypotheses are the most likely. The paper shows an operational experiment where avoiding search is used as the driver for the introduction and negotiation of syntax. The experiment is also a demonstration of how Fluid Construction Grammar is well suited for experiments in language evolution.

1 Introduction

The research reported in this paper is part of a growing body of research that tries to show through careful computational and robotic experiments how communication systems with properties similar to those of human natural languages may emerge in populations of agents. (See recent overviews in [1], [2], and others) Many aspects of language are being studied, ranging from the origins of sound systems, the origins of lexicons, the co-evolution of lexicons with ontologies usable for categorisation, etc. In this paper we focus on issues related to grammar.

We will adopt a functional view on the evolution of language, compatible with cognitive linguistics approaches [3] and construction grammar [4], as opposed to a structuralist view, familiar from generative grammar [5]. Broadly speaking, the functional view argues that syntax is motivated by attempts to solve some aspect of the communication problem, whereas in a structuralist view, syntax is not motivated by communicative function. These two views lead to different models of language evolution. Genetic and cultural transmission models such as those of Nowak, et.al. [6] or most models based on the Iterated Learning framework [7] illustrate a structuralist view. Agents introduce hierarchical structure as they induce (or inherit after mutation) the language of their parent, and this structure is reused when they produce language themselves. But the structure is not motivated by issues that arise when attempting to communicate. Indeed communication itself is not modeled, only the transmission process. The nature of the resulting grammar is therefore solely due to the nature of the learning algorithm (e.g. induction based on minimal description length) and chance factors. Although this is probably a reasonable model for language transmission it makes it hard to understand why language is the way it is and how the intricate structure we observe might have arisen.

In this paper we explore a functional view on language evolution, which means that features of grammar are supposed to emerge because they deal with a particular issue that embodied communicating agents necessarily have to solve. This implies that we must first create situations in which embodied agents encounter certain difficult issues which prevent them from communicating successfully with reasonable cognitive effort, and then formulate repair strategies for dealing with these issues that lead to increased grammaticality and a better communication system.

Our team has already reported several very concrete operational examples of this approach. Steels [8] argued that grammar is needed to link partial meanings introduced by different lexical items and showed computational simulations which use the damping of equalities between variables (which arise when partial meanings are only implicitly linked to each other) as main driver for introducing case grammar [9]. De Beule and Bergen [10] showed how compositional coding (as opposed to holistic coding) emerges when there is a sufficiently large fraction of structured meanings that need to be expressed. When agents reuse existing expressions, communicative success increases more rapidly and cognitive load decreases as they need smaller lexicons. Steels and Loetzsch [11] argued that embodied communication involving spatial relations (like left or right) requires recruiting the ability to adopt different perspectives and communicating explicitly the perspective from which a scene is described because it substantially increases communicative success and decreases the cognitive effort of the agents.

In this paper, we report on another case study, now focusing on the issue of combinatorial explosions in parsing. Multiple hypotheses in parsing arise unavoidably as soon as the same syntactic pattern is re-used as part of a bigger structure. Moreover natural languages re-use the same syntactic pattern with different levels of detail. For example, it is possible to build a noun phrase with just an article and a noun ("the box") but also with an article, an adjective and a noun ("the big box"), or two noun phrases combined with a preposition ("a small box next to the orange ball"), and so on. Unless there is additional syntax, "a" or "the" in the latter example can both be combined with either "box" or "ball", and "big" or "orange" can equally be combined with both nouns and the phrase can also be parsed as "the orange ball next to a small box". Clearly languages introduce syntactic means to restrict the set of possible combinations which otherwise would quickly run out of hand. In English, this additional syntax is usually based on word order, but other languages may use other syntactic devices such as agreement between number and gender.

This suggests that detection of parsing ambiguity can be used as a motor that drives the introduction of syntax. The speaker can re-enter the utterance that he has just produced to detect ambiguity and then add additional constraints if there is a risk of combinatorial explosion. The hearer can parse the utterance produced by the speaker, 'bite the bullet' to arrive at an interpretation even if it involves search, but then use the syntactic sugar that the speaker might have introduced as a way to avoid that search in the future. This is precisely the repair strategy that we have implemented and report on in this paper.

The rest of the paper describes the experimental set-up, how failure or cognitive strain is detected, the repair strategies, and the effect of their application on the communicative success and cognitive effort of language users. The experiments rest on highly sophisticated technical tools contributed by many other members of our team (see acknowledgement). Lexicon and grammar use the Fluid Construction Grammar (FCG) framework, which is a new HPSG-style implementation of construction grammar [12]. An implementation on a LISP substrate has been released for free download through http://arti.vub.ac.be/FCG/. The technical part of this paper assumes some familiarity with FCG, and in particular the way that hierarchy is handled using the J-operator (see [13]). Finally, semantic aspects are handled through grounded procedural semantics based on a constraint language called the Incremental Recruitment Language (IRL) (see [14]).

2 Modeling Communication

It has been well documented that the ability to establish a joint attention frame is an important prerequisite for human-like communication [15]. A joint attention frame is only possible when agents share motives and communicative goals and find themselves in the same (physical) situation in which they can establish joint attention to the same objects or aspects of the situation. We achieve these prerequisites by carefully constructing a language game, which is highly constrained routinised interaction between agents. The game takes place in a physically shared environment and shared motives and communicative goals are part of the scripts through which these robots interact with each other in this environment. For example, two robots are both paying attention to an orange ball that is pushed around by an experimenter and they describe how the current movement of the ball is different from previous events [11] (see figure 1 bottom).

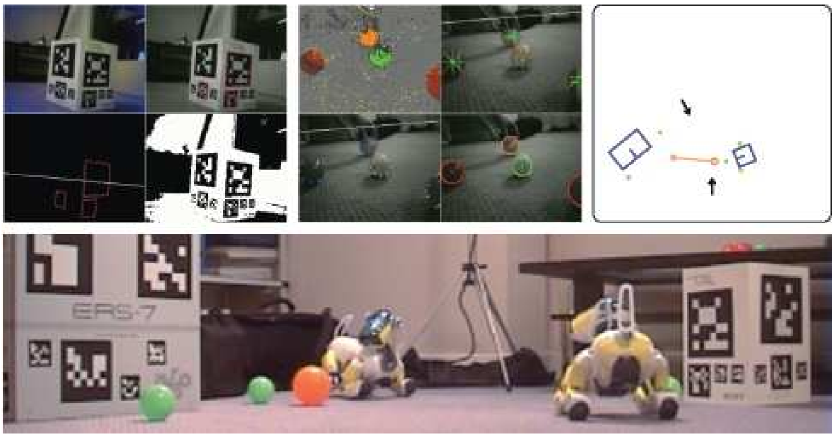

Fig. 1. Typical experimental setup in our language game experiments. The bottom shows two robots moving around in an environment that contains balls and boxes. The robots are equipped with a complex sensory-motor system, able to detect the objects and build an analog world model of their location and trajectories. The top shows objects being detected (left) and a ball trajectory (right) as seen by the robot on the right.

To achieve a communicative goal, the speaker must first conceptualise what to say and this must be based on a perception of the world as experienced through the agent's sensori-motor system. The agent's world models in our experiments are analog and based on direct output of sensors and actuators (as shown in figure 1 top). Often it is assumed that there is a simple straightforward way to transform a non-symbolic world model into a categorical situation model, which is a representation of the world in the form of facts in some variant of predicate calculus however we believe that this assumption is not realistic. Instead we have adopted a 'procedural semantics' view [16] in which the meaning of a phrase is not an expression to be matched against a situation model, but a program to perform the necessary categorisations and conceptualisations in order to achieve specific communicative goals like reference. Conceptualising what to say then becomes a planning process and interpretation is equal to running the programs reconstructed from parsing a sentence. We call these meanings 'semantic programs'.

To operationalise this procedural semantics, our team has designed and implemented a constraint propagation language IRL (Incremental Recruitment Language) [14]. The primitive constraints are cognitive operations like filtering the set of objects in the world model with an image schema, taking the intersection of two sets, checking whether a certain event fits with the dynamical behavior of the objects in a particular situation, etc. A simple example of a constraint network in IRL-notation is as follows (no control flow is expressed):

```
1. (external-context ?s1)    ; ?s1 is the current context
2. (filter-set-prototype ?s2 ?s1 ?p1); retain elements of ?s2 matching ?p1
```

```
3. (prototype ?p1 [box])    ; introduce prototype to be used
4. (unique-element ?o1 ?s2) ; ?o1 is the unique element from s2
```

All symbols preceded by question marks are variables. ?o1 will be bound to an object in the world model, ?s1 and ?s2 will be bound to sets of objects, and ?p1 is bound to a prototype or image schema that is used to filter the set of objects in the context (?s1). The resulting set (?s2) is assumed to contain only a unique element (?o1).

Constraints are exercised until the possible bindings of variables are restricted as much as possible, ideally to single choices. The constraint networks operate in different directions for conceptualisation and for interpretation. For example, in the phrase "the box" the hearer is given a prototype [box] and uses it to classify the objects in the world model, perhaps by a nearest neighbor match with the prototype. But in conceptualising, the speaker must find a suitable prototype, so that if this prototype is used, the hearer will be able to find back a set containing the referent. The constraints are not only able to perform a particular operation over the world model, such as categorising a set of visual stimuli in terms of color categories, but also extend the available repertoire (the ontology) of the agent. In other words, constraints can invent new categories, adjust categories, introduce new prototypes, etc. This way the acquisition of a conceptual repertoire is completely integrated in the process of conceptualising and interpreting language and it is therefore possible to have a strong interaction between the two. IRL features mechanisms to find a network that is adequate for achieving a particular communicative goal and chunking found solutions so that they can be reused later. In multi-agent experiments, each agent builds up his own repertoire of composite constraints and ontologies and they get coordinated due to alignment.

The next task of the speaker is to transform a constraint network into a language utterance using the lexical and grammatical constructions available in his inventory. We have adopted the perspective of construction grammar [17], [18] and our team has designed and implemented a new formalism known as Fluid Construction Grammar (FCG). Construction grammar assumes that every rule in the grammar has both a semantic and a syntactic pole. This contrasts with a (generative) constituent structure grammar that specifies only syntax, and semantics is supposed to be defined separately by translation rules. The semantic pole of a construction specifies how meaning has to be built up in parsing or decomposed in production, and the syntactic pole how the form has to be analysed in parsing or built in production. An important feature of FCG is that rules are truly bi-directional. The same rule can be used both for parsing and production, even if it involves hierarchy ([13]). The syntactic and semantic structure being built during parsing and production takes the form of feature structures and unify and merge are the basic operations that are used for expanding these feature structures through the application of rules, similar to widely used HPSG frameworks [19].

There is a systematic correspondence between constraint networks and grammar (explained in more detail in [20]) in the sense that (1) lexical items introduce the semantic objects used by constraints (for example prototypes, relations, categories, etc.), (2) first order constructions specify how these items are used, and (3) higher order constructions combine these and establish linking of variables between them. We illustrate this with hand-coded examples because they make it easier to understand the underlying ideas, but the agents invent their own words, their own syntactic categories, etc.

The following FCG rule is an example of a lexical rule that associates a semantic object (the prototype [box]) with a stem.

```
(def-lex-stem-rule [box]
  ((?top-unit
    (meaning (== (prototype ?prototype [box]))))
   ((J ?new-unit ?top-unit)
    (context (== (link ?prototype)))
    (sem-cat (prototype ?prototype))))
 <-->
  ((?top-unit
    (syn-subunits (== ?new-unit)))
   (?new-unit
    (form (== (stem ?new-unit "box"))))))
```

The left-pole contains a bit of semantic structure (introducing a prototype [box]) and a semantic category for it, and the right pole a bit of syntactic structure (introducing the stem “box” expressing this prototype).

A first order construction that uses prototypes is as follows.

```
(def-con-rule CommonNoun
  ((?top-unit
    (sem-subunits (== ?prototype-unit))
    (meaning (== (filter-set-prototype ?result-set ?context ?prototype))))
   (?prototype-unit
    (context (== (link ?prototype))))
   ((J ?new-unit ?top-unit (?prototype-unit))
    (context (== (link ?result-set ?context)))))
 <-->
  ((?top-unit
    (syn-subunits (== ?prototype-unit)))
   (?prototype-unit
    (syn-cat (== (lex-cat CommonNoun))))
   ((J ?new-unit ?top-unit (?prototype-unit))
    (syn-cat (== (constituent CommonNoun))))))
```

It handles another bit of meaning, namely a filter-set-prototype constraint and associates it with a Common Noun constituent. The context feature of a unit refers to variables that are linked from pending subunits to other subunits. Thus the ?prototype-unit introduces ?prototype which is used by the filter-set-prototype constraint to filter the set of objects bound to ?context and return a new set ?result-set. The syntactic pole requires that a unit is found whose lexical category is CommonNoun and it creates a CommonNoun constituent.

The next example is a higher order constraint which groups a CommonNoun constituent, as could have been built by the previous construction and an Adjective constituent, into a new CommonNoun constituent. The meaning pole of this construction does not add new meaning, except to link the appropriate variables coming from each of the subunits with each other. The J-operator creates a new unit that has the adjective and common noun units as its children and presents itself as having syntactic category 'constituent CommonNoun'. Therefore this new unit can recursively be combined as if it were a constituent CommonNoun.

```
(def-con-rule AdjNoun
  ((?top-unit
    (sem-subunits (== ?noun-unit ?adj-unit)))
   (?noun-unit
    (context (== (link ?filter-set ?input-set))))
   (?adj-unit
    (context (== (link ?target-set ?filter-set))))
   ((J ?new-unit ?top-unit (?noun-unit ?adj-unit))
    (context ((link ?target-set ?input-set)))))
  <-->
  ((?top-unit
    (syn-subunits (== ?noun-unit ?adj-unit)))
   (?noun-unit
    (syn-cat (== (constituent CommonNoun))))
   (?adj-unit
    (syn-cat (== (constituent Adjective))))
   ((J ?new-unit ?top-unit (?noun-unit ?adj-unit ))
    (syn-cat (== (constituent CommonNoun))))))
```

Constructions like these are systematically built up by agents, as explained in more detail in [20]. Whenever a construction for a semantic object or constructions for constraints that use them are missing, new ones are fabricated and the repertoire of each agent gradually expands. Note that these constructions so far contain virtually no syntax. They only contain very broad semantic subcategorisations (such as 'prototype') and basic syntactic categorisation (lexical categories and constituents).

When the speaker has produced an utterance that completely expresses the meaning, he first re-enters it, in other words he parses the utterance and checks whether the meaning is the same one as he wanted to express and whether no other problems come up (such as combinatorial explosions). Suppose that the speaker is entirely satisfied, then the utterance is transmitted to the hearer. The hearer attempts to parse the utterance and reconstruct a constraint network that can run over his own sensory world model. Both the parser and the constraint network are 'fluid' in the sense that they attempt to arrive at an interpretation even if there are unknown words, rules missing, etc. Based on feedback in the language game and on constraints coming from the language, the hearer reconstructs as well as possible the meanings that are compatible with the joint attention frame (the shared motives, communicative goal, and physical situation) and uses that to reconstruct missing rules. Because speaker and hearer invent new constructions all the time, incompatibilities are bound to arise, but these are flushed out by the lateral inhibition dynamics that we use in all our experiments. It is based on increasing success of winning inventory items (concepts, constraint networks, words, grammatical constructions, etc.) while decreasing competing solutions. The current experiment is based on an operational implementation of all this (reported in more detail in [20]) and we now move beyond these capabilities to focus on the problem of combinatorial explosions.

The bootstrapping of a language system is an extraordinarily difficult undertaking for a group of agents and it is greatly aided if they start simple and then increase complexity as they master basics. This growth in complexity can be regulated by the agents themselves, who monitor success and then increase challenge (based on the 'autotelic principle' described in [21]). In the language game implemented for the current experiment the speaker has to talk about an object in the shared context between the speaker and the hearer. As an example, assume the speaker wishes to talk about a particular object in the context, e.g. a ball. Depending on the shared context the required utterance can range from very simple (e.g. "ball" when there is only one ball) to more complex (e.g. "big ball" when there are multiple balls but this is the only big one). Even spatial relation may need to be expressed to discrimate the object. The most complex utterances that could be construed in the current experiment are of a combination of a spatial relation and Adjective-Noun constructions which renders utterances like "big ball next-to red box". It is this complexity that the agents can regulate, meaning that they won't start talking about " big ball next to red box" until they are confident in talking about more simple scenes.

3 From a lexical to a grammatical language

We now start by considering a lexical language, which is one where no grammatical constructions are built at all. When words are missing, agents execute repair strategies to invent new words (as speaker) or adopt them (as hearer). In the language game constructed for these experiments we speak of communicative success when the speaker can produce an utterance so that the hearer can infer the exact same meaning by interpreting this utterance. In the first experiment as meanings become more complex, lexical items are built for the total meaning as in holistic coding. Results for experiments for 5 runs with 5 agents playing 4000 games are shown in figure 2. We see that quite quickly agents reach a high level of communicative success and the lexicon becomes optimal after about 1500 games. Since there are about 13 basic semantic objects (prototypes, categories, relations) in the example domain, an optimal lexicon just for the semantic objects is around 13 words.

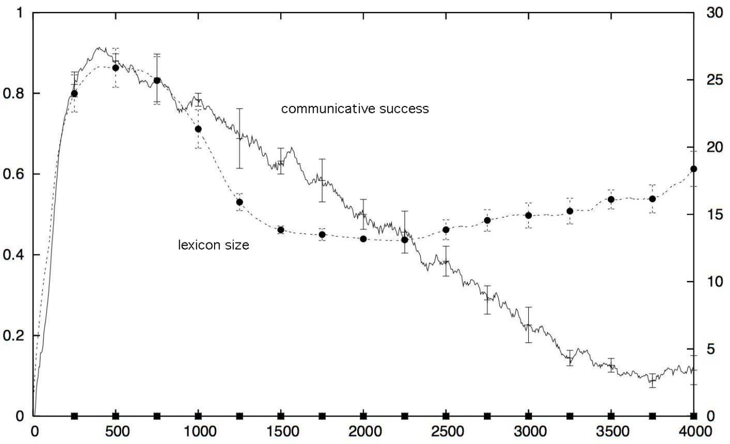

Fig. 2. Experiments where 5 agents use a purely lexical language to bootstrap a communication system. It is initially successful but cannot cope when the complexity of meanings begins to increase.

But challenge is increasing steadily and communicative success starts to drop. In response, the lexicons of the agents begin to increase as they use holistic coding to cope with the more complex meanings. If we continue the experiment we see that communicative success climbs back up, but only at the expense of a much larger lexicon that is slower to get off the ground, more difficult to learn, and requiring more memory. Clearly it would be better if agents recruit strategies based on exploitation of grammar.

This happens in the second experiment (figure 3). In this experiment agents build grammatical constructions, for example to combine adjective-like semantic objects (categories) and noun-like predicates (prototypes) as in “big ball”. The constructions are triggered by the need to express explicitly equalities between variables (as explained in [2]). The lexicon shows the same overshoot in the beginning and then stabilisation around 13 words as competing words are resolved and lexical coherence reached. The necessary grammatical constructions are built early on. They are similar to the Adj-Noun constructions above, i.e. without significant syntax. Only two constructions are needed so far and agents quickly reach agreement on them. The figure also shows ‘grammaticality‘, this is the running average of number of utterances that make use of grammatical constructions. We see that the agents are able to cope with increasing complexity but it comes again at a price. The search space consisting of all applicable grammatical constructions steadily increases during parsing because there are multiple ways in which constructions can be applied. Because the interpretation is no longer guaranteed to be deterministic this also creates the possibility that the hearer has multiple interpretations at the end of the game. In this case we speak of communicative success only when the hearer is able to pick the correct

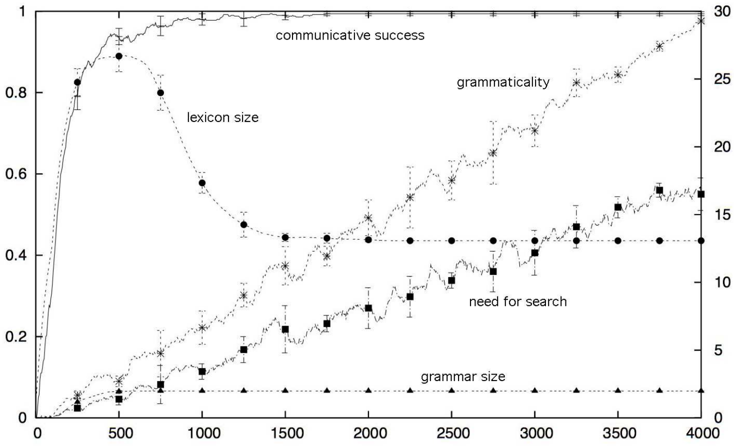

Fig. 3. Experiments where 5 agents use grammatical constructions in addition to a lexicon. They are now able to maintain communicative success even as complexity of meanings increases. But this again comes at a price because the agents have to search through the space of applicable grammatical constructions. This is expressed by the 'need for search', which is a running average of the number of utterances that couldn't be interpreted deterministically. In other words, that required search.

one by inspecting its world model. It is however the need for search that in the current experiment creates the need to recruit mechanisms to dampen the search as shown in the next experiment.

4 Diagnosing and repairing combinatorial explosions

After the speaker produced an utterance he does not immediately utter it for interpretation by the hearer but instead interprets his own utterance himself. The difference with normal interpretation being that the speaker however knows the intended meaning and therefore has a much easier task interpreting his own utterance then the hearer will. We call this special kind of interpretation "re-entrance" [22]. During re-entrance the speaker builds new constructions if his interpreted meaning contains variables that should have been equal but are not. But re-entrance can also be used to diagnose whether search would be taking place in the hearer. For example, for "big small ball box", the Adjective-Noun construction triggers twice, creating a search space with two different possible interpretations: (1) "big box - small ball", and (2) "small box - big ball". The speaker knows which interpretation is intended. He can therefore analyse the choice point where a particular construction could match more than once and introduce additional syntax so as to avoid such choice in the future. In the present experiment, the speaker remedies the situation by imposing word order.

Concretely if a conflict arises between two Adjective-Noun constructions, the speaker knows that the syntactic pole of this construction is not specific enough and chooses (randomly) an order between the noun and adjective units and expands the syntactic pole to become as follows:

```
((?top-unit
   (syn-subunits (== ?noun-unit ?adj-unit))
   (form (== (meets ?noun-unit ?adj-unit))))
 (?noun-unit
  (syn-cat (== (constituent CommonNoun))))
 (?adj-unit
  (syn-cat (== (constituent Adjective))))
 ((J ?new-unit ?top-unit (?noun-unit ?adj-unit ))
  (syn-cat (== (constituent CommonNoun)))))
```

The only difference with the old Adjective-Noun construction is the addition of a form constraint in the top-unit. This form constraint requires that the noun unit 'meets‘ the adjective unit, i.e. has to come right before it. After the speaker has diagnosed and repaired his own inventory of constructions he restarts production. Because he added the form constraint, the speaker can no longer choose any combination of the four lexical entries but can only choose between “box big ball small” or “ball small box big” which both have the same meaning and therefore pose no real conflict.

Because there is no telepathy the hearer is not aware of the diagnosing and repairing the speaker went through. The hearer will parse the utterance and (if all goes well) still arrive at two possible interpretations for “box big ball small”. However, to disambiguate, he can check against his world model which one of these is valid in the current situation. Having recovered the ‘correct‘ interpretation, the hearer goes back to the constructional choice point that gave rise to search and takes the syntactic features used in the speaker's utterance (i.e. the word order) as a clue to tighten up the construction himself. In a community of agents there will be different word-orders competing but the mechanism of lateral inhibition also used to drive the lexicon towards coherence will eliminate those that are less successful and grammatical coherence self-organises.

These repair strategies are seen at work in the next experiment (figure 4). We see again a rapid climb of communicative success in the beginning and overshoot in lexicon size, which becomes optimal. At the same time we see emergence of grammatical constructions. There is also an overshoot (in the sense of more constructions circulating in the population than strictly needed) because there are different ways to add syntax to a construction (e.g. Adj-Noun versus Noun-Adj order). The competing syntax is however flushed out due to lateral inhibition. The most important point, seen in the bottom graph, is that the search space is now completely under control and the grammar becomes deterministic.

5 Conclusion

This paper has argued that grammar is not imposing arbitrary structure on lexical items but that it is motivated by the need to solve certain issues that

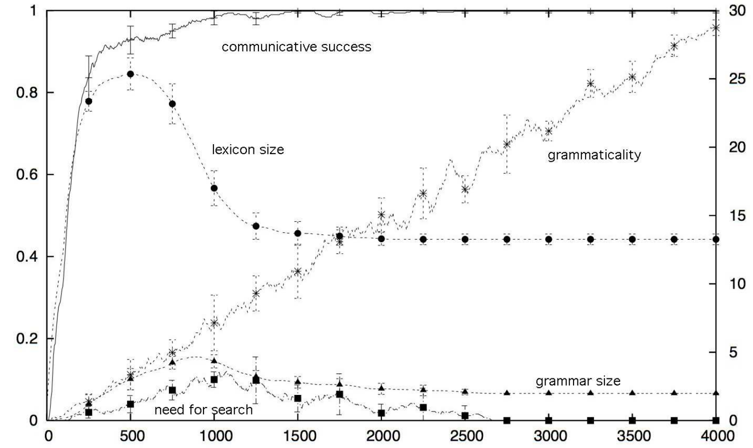

Fig. 4. Experiments where 5 agents now tighten grammatical constructions with additional syntax in order to avoid combinatorial search. We see a drastic reduction in search needed, even till the point (after the 2700^{th} game) that parsing becomes deterministic in all cases.

arise in communication among embodied autonomous agents. One obvious issue is that combinatorial explosions occur during parsing which need to be dampened as fast as possible, otherwise memory and processing time may reach such a level that the hearer has to give up. Additional syntactic constraints help because they provide cues that the hearer can use to cut down parse avenues that are not intended by the speaker. Syntactic constraints can take the form of word order constraints, agreement, or semantic and syntactic subcategorisation. The paper has substantiated this argument by showing a working implementation based on Fluid Construction Grammar.

Acknowledgment

The research reported here has been conducted at the Artificial Intelligence Laboratory of the Vrije Universiteit Brussel (VUB) and at the Sony Computer Science Laboratory in Paris. Partial funding for the Sony CSL work has come from the EU FET-ECAgents project 1940. We are very much indebted to Joachim De Beule for help in the technical foundations for the experiments reported here.

References

1. Minett, J. W. and Wang, W. S-Y. (2005) Language Acquisition, Change and Emergence: Essays in Evolutionary Linguistics. City University of Hong Kong Press: Hong Kong.

2. Steels, L. (2005) The emergence and evolution of linguistic structure: from lexical to grammatical communication systems. Connection Science, 17(3-4):213–230.
3. Langacker, R. (1990) Concept, Image, and Symbol: The Cognitive Basis of Grammar. Berlin and New York: Mouton de Gruyter.
4. Goldberg, A. (2006) Constructions at Work. Oxford University Press, Oxford.
5. Jackendoff, R. (1996) The Architecture of the Language Faculty. The MIT Press, Cambridge Ma.
6. Nowak, M.A., N. Komarova, P. Niyogi (2001) Evolution of universal grammar. Science, 291:114-118, 2001
7. Smith, K., Kirby, S., and Brighton, H. (2003). Iterated Learning: a framework for the emergence of language. Artificial Life, 9(4):371-386.
8. Steels, L. (2005) What triggers the emergence of grammar? A. Cangelosi and C.L. Nehaniv (ed.) (2005) Proceedings of the Second International Symposium on the Emergence and Evolution of Linguistic Communication (AISB 2005 Convention, University of Hertfordshire, Hatfield, UK). AISB.
9. Steels, L. (2002) Simulating the evolution of a grammar for case. Fourth International Conference on the Evolution of Language, Harvard University.
10. De Beule, J. and B. Bergen (2006) On the Emergence of Compositionality. Cangelosi, A., K. Smith and W.Smith (eds.) Sixth Conference on Evolution of Language Conference, World Scientific Pub. New York.
11. Steels, L. and M. Loetzsch (2006) Perspective Alignment in Spatial Language. In: K. Coventry, J. Bateman, and T. Tenbrink (eds.) Spatial Language in Dialogue. Oxford University Press, Oxford.
12. Steels, L. (2004) Constructivist Development of Grounded Construction Grammars Scott, D., Daelemans, W. and Walker M. (eds) (2004) Proceedings Annual Meeting Association for Computational Linguistic Conference. Barcelona. p. 9-19.
13. De Beule, J. and Steels, L. (2005) Hierarchy in Fluid Construction Grammar. In Furbach U., editor, Proceedings of KI-2005, pages 1-15. Lecture Notes in Computer Science. Springer-Verlag, Berlin.
14. Steels, L. and J. Bleys (2005) Planning What To Say: Second Order Semantics for Fluid Construction Grammars. In: Proceedings of CAEPIA 2005, Santiago. LNAI Springer-Verlag, Berlin.
15. Tomasello (1995). Joint attention as social cognition. In Moore, C. and Dunham, P., (Eds.), Joint attention : its origins and role in development. Lawrence Erlbaum Associates.
16. Johnson-Laird, P.N. (1977) Procedural Semantics. Cognition, 5 (1977) 189-214.
17. Croft, William A. (2001). Radical Construction Grammar: Syntactic Theory in Typological Perspective. Oxford: Oxford University Press.
18. Bergen, B.K. and Chang, N.C.: Embodied Construction Grammar in Simulation-Based Language Understanding. In: Ostman, J.O. and Fried, M. (eds): Construction Grammar(s): Cognitive and Cross-Language Dimensions. John Benjamin Publ Cy., Amsterdam (2003)
19. Pollard, C. and I. Sag (1994) Head-Driven Phrase Structure Grammar. University of Chicago Press (1994)
20. Steels, L. and J. De Beule (2006) The Emergence of Hierarchy in Fluid Construction Grammar. submitted.
21. Steels, L. (2004b) The Autotelic Principle. In Fumiya, I. and Pfeifer, R. and Steels,L. and Kunyoshi, K., editor, Embodied Artificial Intelligence, Lecture Notes in AI (vol. 3139), pages 231-242, Springer Verlag. Berlin, 2004.
22. Steels, L. (2003) Language-reentrance and the 'Inner Voice'. Journal of Consciousness Studies, 10(4-5), pages 173-185.