Origins of Recursive Phrase Structure through Cultural Self-Organisation and Selection

Emília Garcia Casademont

TESI DOCTORAL UPF / ANY 2018

DIRECTOR DE LA TESI
Luc Steels Departament CEXS

upf. Universitat Pompeu Fabra Barcelona

Acknowledgments

First, I would like to express my gratitude to Luc Steels for allowing me to experience what science was and to meet a lot of extraordinary scientists and people on the way. I also want to thank Katrien Beuls and Pieter Wellens for introducing me to a whole new world with patience from my first days in Brussels; Michael, for encouraging me and seeing the value of my research; a la Maria, per tant de compratit al despatx; Remi and Paul, for always welcoming me in Paris; a Harold, por compartir sus ideas y entusiasmo; a Andrea, por toda la fuerza que tiene y transmite; Eörs Szathmáry and the Insight crew, for shedding so many lights in the fascinating world of evolution, i al Miquel; per compartir i aprendre plegats, del Guinardó a Brussel·les.

I also want to sincerely thank the committee members for accepting and helping me to get to the end of this project.

També vull mostrar el meu agraïment al Projecte Òrbita, per tornar-me la motivació; a la gente de Zapiens, para pensar que la ciencia se puede aplicar en problemas reales y ser nuevos compañeros; a la meva família; i a tots els que hi heu sigut al llarg del camí per tocar, un cafè, una estona a Rovira, Dinar*s* del Mes, ajudes administratives, converses a l'Esterri, escapades encalmades, condividere multiple aventure, glasses of wine, converses inacabables, il·lusionar-me amb nous projectes, acceptar-me com a nou membre de la família, valorar-me i estimar-me.

The research reported here has been funded by a Marie Curie Integration Grant EVOLAN and the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 308943, which is the FET-OPEN Insight project.

Abstract

Many human languages in the world exhibit phrase structure. Phrase structure combines words, phrases, and both, into phrases, and it may empower language systems to exploit recursion. This thesis pushes forward the hypothesis that phrase structure is not an accidental structural property of language, but rather an adaptation of language systems to enable the computation of language. I propose a minimal operational model of communication as a specific language game, which together with concrete learning operators shows how a population of artificial agents is able to self-organise a system exhibiting phrase structure. After demonstrating that phrase structure reduces the complexity of language computation, I propose concrete mechanisms in the form of learning operators whose application introduces variation in the language of the agents and selection on the reduction of the computational cost. The mechanisms are implemented and tested in computer simulations as an evolutionary explanation for the emergence of phrase structure, including cases exploiting recursion.

Resum

Molts llenguatges naturals es basen en gran mesura en gramàtiques sintagmàtiques. Les gramàtiques sintagmàtiques combinen paraules i sintagmes en altres sintagmes, i poden capacitar els sistemes lingüístics a fer ús de la recursió. Aquesta tesi enforteix la hipòtesi que les gramàtiques sintagmàtiques no són una propietat estructural accidental del llenguatge, sinó que són una adaptació dels sistemes lingüístics que permet que el llenguatge pugui ser processat adequadament. Proposo un model minimal de comunicació basat en un joc del llenguatge en concret que defineixo, i que juntament amb operadors d'aprenentatge específics mostra com una població d'agents artificials és capaç d'autoorganitzar un sistema basat en gramàtiques sintagmàtiques. Un cop demostrat que les gramàtiques sintagmàtiques redueixen la complexitat del processament del llenguatge, proposo mecanismes concrets en forma d'operadors d'aprenentatge l'aplicació dels quals introdueix variació en les gramàtiques dels agents i selecció en la reducció del cost de processament. Els mecanismes són implementats i avaluats en simulacions com a una explicació evolutiva de l'emergència de les gramàtiques sintagmàtiques, incloent casos en què es fa ús de la recursió.

Contents

III Conclusions 193

List of Figures

List of Tables

Part I

Introduction

Chapter 1

INTRODUCTION

Human languages are complex systems made of arbitrary and culturally learnt conventions which can be combined to produce infinitely many new meaningful expressions. Moreover, our communicative system is constantly changing [66]. In contrast, other species' communicative systems exploit a limited number of combinations, including the communicative systems of great apes and aquatic mammals, most of which however do change [34, 40].

Human communicative systems are so fundamental to our existence that many scientific endeavors nowadays are devoted to understanding the nature of our languages [13]. Many theories and methodologies are explored today to deepen our knowledge about languages and the human faculty of language, and they deal with a big spectrum of aspects. One of them is the study of language evolution, which following its biological counterpart, puts evolution into the core of the whole understanding of linguistic phenomena.

Within the study of language evolution, various phenomena are studied, from biological aspects related to the language-ready brain [35, 19] or cognitive and cultural transmission constraints [1, 83, 110], to the role of language in the evolution of societies [70], or to the cultural environmental pressures for a language to express certain relevant meanings [65, 43]. The field of evolutionary linguistics is the one concerned with the study of language evolution at all levels, and it grows close to other disciplines, such as artificial intelligence, biology [81, 44], anthropology [23], psychology [46], or physics [6, 84], among others.

This thesis is concerned with research questions related to one of the most studied problems in evolutionary linguistics: the origins and evolution of grammar. The question why human languages exhibit intricate syntax and how this could have arisen in the evolution of our species has motivated many research studies in the last 50 years, some of them stressing the role of biological evolution and general brain functionalities, and some of them stressing the role of cultural evolution.

Here I assume a selectionist approach at the linguistic level, where language is seen as an expression of cultural evolution. This implies that language syntactic structure is not simply seen as an accidental property of language but is motivated by the challenge of collectively building a communication system that can be produced and comprehended by a human brain. This implies in particular that only finite resources in terms of memory and computational power should be required. The thesis focuses on modelling the effect of selective pressures acting on the cultural (linguistic) level on guiding a population of agents in the conventionalization of language systems minimizing the use of such resources. In particular, I study the case of language systems exhibiting recursive phrase structure.

This thesis contributes to the study of the cultural evolution of language, which occurs by the means of communicative actions [52, 71, 76, 89]. The manifestation of cultural evolution doesn't require many human generations, instead, it can be expressed in much less than one's lifespan, as it happens with slang or fashion-related phenomena.

1.1 Phenomena of Phrases and Recursion

Similarly to the study of animal species, linguists have historically tried to classify languages according to their features. However, it is not obvious what are these features, nor what is their level of description (e.g. phonetic, lexical, syntactic, pragmatic, etc.). This thesis is dedicated to the study of the emergence of one of the most well-known sytactic features: phrase structure. The motivation to study phrase structure as one of the fundamental features for language classification comes from three main observations.

The first observation is the phenomenon of fixed word order. That is, given an empirical definition of word as *the smallest element that may be uttered in isolation with semantic or pragmatic content*, fixed word order is the fact that correlations between relative word orders are found among words that are different but behave similarly, in terms of syntax. One of the most common ways in which the syntactic behaviour of a words or any other linguistic elements can be compared is by performing tests of substitution. Two elements behave equally when the results of substituting one for the other in an utterance is always again a well-formed utterance. In most linguistic theories, all the words that behave equally or almost equally are identified as *lexical categories*. Therefore, following these theories, fixed word order is given by sequences of lexical categories.

E.g: English exhibits a lot of fixed word order; in a noun phrase, the order of constituents is always *article-adjective-noun*; on the other side, romance languages exhibit a less strict fixed word order, because in most contexts the order of constituents in a noun phrase is *article-noun-adjective*, but in some cases is *article-adjective-noun*. Many human languages, if not all, exhibit certain systematicity in word order.

The second observation is that fixed word order often implies adjacent words in the utterance, and which are part of the same constituent. A constituent is a central concept in linguistics and it refers to words that behave as a syntactic unit to a certain extend. In the case of adjacent words, the syntactic behaviour of the potential constituent is compared to a single word, which in English could be a pronoun, such as *it* or *they*. Again, if the result of the substitution test is a well-formed utterance, the sequence of adjacent words forms a (syntactic) constituent. Linguists call *phrases* this kind of constituents, which are made of adjacent words.

Moreover, linguists suggested that the linguistic knowledge required to process phrases shares fundamental aspects across phrases, and they formalised this conception in the form of phrase structure rules, and the set of phrase structure rules, as phrase structure grammars [18, 80].

First of all, they reused the same idea described earlier to define lexical categories, but this time for phrases and identified phrases to *phrasal categories*. And second, they explored to what extend the structural properties of language can be described through phrase structure rules. They modelled phrase structure rules as rewrite rules, where a sequence of lexical and phrasal categories is substituted by a unique phrasal category. These rewrite rules are usually represented using the form $A \rightarrow B\ \ C$, meaning that the constituent A is separated into the two elements B and C, or that B and C are rewritten as A. The names used for the constituents and elements are identified to their phrasal or lexical categories. Some examples of rewrite rules modelling English are shown in figure 1.1:

$$S \rightarrow NP\ \ VP$$

$$NP \rightarrow Det\ \ N$$

$$NP \rightarrow AP\ \ NP\ \ PP$$

$$AP \rightarrow Adv\ \ Adj$$

Figure 1.1: Example of phrase structure rules for English. The sequence on the right hand side can get substituted by the element on the left hand side.

Therefore, the feature of *phrase structure* refers to the property of language to be described (and hypothetically processed) using phrase structure rules.

Finally, the third observation that motivated the study of phrase structure had to do with linguistic examples of potentially infinite sentences:

- very very very very ...
- I think you think I think you think ...

- a friend of a friend of a friend of ...

This phenomenon is identified with that of *recursion*, given that one has the intuition that there is a structure that includes itself again and again. Of course, none of the examples above are well-formed phrases or utterances, and they would need at least an adjective, a clause and a noun phrase, respectively, to become so. However, nothing prevents a language user of English to say ”very very big”, and once this is known, say *very* ”very very big”, and keep doing so, which is the idea of identifying the previous examples as recursive.

Actually, this can be found at a very high level of language description, where utterances may include themselves as complements. This is the main mechanism used for the Catalan songs called *cançons de l’enfadós*. These songs are meant to be sung infinitely in order to annoy those that are listening to it, e.g. *el ciclista de pega*.

In terms of phrase structure grammars, recursion is described by rules including the same phrasal category at both sides, and these rules are called *recursive phrase structure rules*, see figure 1.2.

Many authors agree on the fact that *compositionality, along with recursion, is the fundamental feature of human syntax that gives us open-ended expressivity* [62, 59]. In this context, *compositionality* refers to the fact that the meaning of two or more words in combination is a function of the meaning of every one of the words.

$$\mathbf{AP} \rightarrow Adv \;\; \mathbf{AP}$$

$$\mathbf{Pred} \rightarrow N \;\; \mathbf{Pred}$$

$$\mathbf{NP} \rightarrow \mathbf{NP} \;\; PP$$

Figure 1.2: Recursive phrase structure rules are those rules in which the same phrasal category appears at the two sides of the rule representation.

To sum up, the linguistic observations that motivated the study of phrase structure are the phenomena of fixed word order, phrases, and recursive phrases, found in languages.

1.2 Motivation

The research of this thesis is part of a scientific program that studies social and cultural processes involved in the evolution of language by building computational models and performing experiments to reconstruct such processes. The reconstruction is made by instantiating artificial languages, and the research contributes both to the fields of artificial intelligence and linguistics [92].

The main motivation for this thesis is to provide evolutionary explanations of recursive phrase structure based on the hypothesis that one of the functions of phrase structure is to reduce the cognitive effort required to process language, focussing particularly on the effect on memory and computational power.

Concretely, in this thesis I aim to contribute to the understanding of the functions of phrase structure by studying in the context of artificial language evolution how language systems exhibiting phrase structure, and recursive phrase structure, may originate from a lexical language system. In order to do that, I propose cultural mechanisms in the form of operators applied during communicative interactions, and study them on artificial agents as mechanistic explanations supporting the evolutionary explanations.

On the other side, a lexical language system, or simply a lexical system, is a language which doesn't use any grammar. In a lexical system, each individual word expresses meaning but language doesn't encode how all word meaning are combined. The emergence of phrase structure on a lexical system is a particular case of the emergence of syntax, given that phrase structure is a syntactic feature of language. And finally, the emergence of syntax is a particular case of the emergence of grammar.

1.3 Methodology

The field of artificial language evolution has grown a lot since AI techniques became more powerful. Most research so far has focussed on the self-organisation of vocabularies, where the Naming Game [86] emerged as the main model and system of reference to test and falsify hypotheses. The Naming Game is a game of reference meant to study how a population of agents can bootstrap a shared lexical system; in this game, agents negotiate a lexical system to distinguish objects in the world.

Many researchers have proposed language strategies for playing naming games, and studied the semiotic dynamics arising from them [69]. Generalizations of the Nanimg Game have also been proposed in order to allow multiple words and more sophisticated language systems or phenomena, some examples are the co-evolution of concepts and names [104, 113], language systems for the domain of colour [8, 14], or language systems for the domain of space [85, 102]. Moreover, there have been several extensions of computational platforms to implement naming games on real robots [100, 104].

Each language strategy defines a particular way to express some meaning by the means of language. For example, marking the agent of an action verb using a particular morphology, as it happens in the ergative-absolutive languages; or in a lower semantic level, using words for geographical features to indicate directions, such as *mountain* or *sea*, as it happens in the slang spoken in Barcelona. A language system is defined by concrete choices in the realisation of a particular language strategy, for the previous cases that means: concrete morphological elements for ergative cases, and concrete words for geographical features to be used for concrete directions.

Lexical systems are modelled as sets of lexical rules, and each of these sets is called a *lexicon*. On the other side, language systems using grammar, i.e. grammatical systems, are modelled as sets of grammatical rules, and each of them is called a *grammar*. Lexical rules describe the relation between words and meaning, and grammatical rules describe how word meanings are combined in an utterance. The border between lexical and grammatical rules is not necessarily strict, and often the notion of *grammar* is used for the set consisting of all the rules.

The computational models of language strategies include a series of learning operators that are implemented at the agent-level, which guide agents to adopt, extend or align their individual grammars. For example, in the case of the Naming Game, a language strategy defines how agents create words and assign them meanings, e.g. for a whole object, for a property of an object, etc. And language systems are determined by sets of word-meaning pairs.

The overall methodology followed in the work of the thesis consists in hypothesizing, implementing and testing mechanisms capable to drive the emergence of phrase structure and recursive phrase structure. These mechanisms are implemented in the form of learning operators, and as I said earlier they are seen as mechanistic explanations, because they identify components that are sufficient to enable artificial agents to self-organise language systems exhibiting phrase structure.

1.3.1 Language Game Framework

In the fifties, Ludwig Wittgenstein introduced the idea of a language game as a mental model to demonstrate how the meaning of a word is its use in language [114]. A language game is a communicative interaction with at least two interactive agents, one having the role of speaker and the other (or others), the role of the hearer. The speaker has a goal that he wants to convey to the hearer, e.g. an action he wants the hearer to perform such as passing him an object of their surrounding; and he can only make use of language to do so. In the nineties, Luc Steels transferred this idea to the field of artificial intelligence by letting software agents and later robots play language games and motivate them to self-organise language systems from the scratch, using these language game models as tools to study the cultural evolution of language [86].

Following this conceptual framework, I study the emergence of phrase structure as the self-organisation of phrase structure in a community of artificial language users which engage in local communicative interactions. These local interactions are modelled as language games, and differently than in the case of the first language games used by Steels and other researchers where agents started from the scratch, in the thesis I assume a lexical system as a prior linguistic knowledge together with a prior conceptual knowledge from which word meanings are defined.

Evolutionary linguistics aims to provide potential explanatory models for the origins of language systems and their continuous evolution, or language change [92]. These models are usually language game models which study language systems (lexical or grammatical) for specific semantic domains [108, 14, 85], hence the negotiation of a language system is driven by the communicative needs of the agents to understand each other in such domain.

Oppositely, in this thesis I study the emergence of language systems exhibiting phrase structure, so my focus is on a particular syntactic aspect, and not in a concrete semantic domain. That is why agents are already initialized with a lexical system, that I use as a reference system as well. Another difference from most language games is that agents are not only affected by communicative pressures to successfully communicate using a collectively negotiated language system, but they are also affected by computational pressures to decrease the cognitive effort in semantic interpretation. In fact, these computational pressures are in the core of the study, given that agents are initialized with a language that is already partially or fully communicative, and besides the pressures to maintaining or increasing communicative success, more emphasis is made to the pressures to decreasing their cognitive effort.

In evolutionary linguistic studies, linguistic selection is modelled as a self-enforcing causal loop in which the communicative outcome with the help of language strategies influences the development of language systems (see figure 1.3). Language strategies may include operators for expansion (e.g invention, recreation, etc.), adoption, alignment and other multiple subprocesses. I propose unique operators for invention, adoption and alignment. A language user can expand his grammar by using invention operators, a language user can infer grammatical rules from other language users by using adoption operators, and finally a language user can update the preferences within his grammar by using alignment operators after a language game interaction.

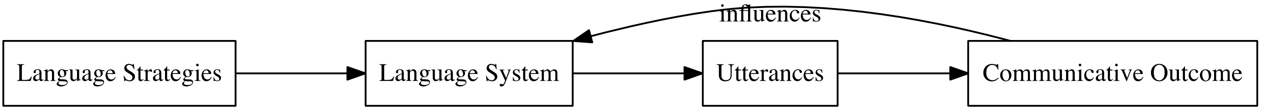


Figure 1.3: The figure represents the self-enforcing causal loop between linguistic selection and the frequency of usage of a language system subscribed by the language strategies considered. Results coming from language game interactions influence the development of the system.

The methodology used to validate the application of a language strategy in a language game model is generally based on multi-agent based simulations, though there are also several analytical studies [32, 69]. Agents interact playing a proposed language game and the results of their communicative interactions feed the model recurrently so that agents can self-organise a language that is adapted to the environment.

The language game framework [86, 75, 88] presupposes a multi-agent setup [37] with at least two agents but generally more. In a typical computational experiment agents have to develop a shared language system by playing language games, and there is no leader or teacher, so that agents can only achieve this goal by engaging in local communicative interactions and align their grammars; and that is why agents are said to *self-organise* a language system. Information in a language game is local to its participants, which means that other agents are not aware of innovations or adaptations that might have come up during an interaction as a result of the application of language strategies by other agents. Only after the innovation is used again in other interactions it can be adopted by other agents and spread through the population. In a language game experiment a population of agents may play thousands of language games.

For all reported experiments in the thesis, I use the multi-agent framework Babel2 [68, 103].

The next section goes further into the cultural selectionist theory of language evolution and its application to language game models.

1.3.2 Theory of Language Evolution by Linguistic Selection

Cultural selection (in this case *linguistic selection*) projects Darwin's original idea of natural selection to the cultural/linguistic level, by describing how variation and selection are instantiated. Applying the replicator dynamics model, which has been successfully applied in many contexts [61], requires first to identify the interactors and the replicators of the system. The interactors interact with the environment or with other units, and carry *traits*. And these traits are the replicators. The replication can happen in multiple ways, in organismic evolution most traits replicate by the multiplication with inheritance of the units so that a whole set of traits gets copied from parent to offspring with some potential random mutation.

In the case of language [95], I identify the interactive agents as the interactors, and every grammatical rule they know as a replicator. In this case, replication will occur by the means of language strategies and their learning operators. Notice that each agent has its own grammatical inventory of rules (grammar), and rules are not copied from one agent to the other, rather inferred by social interaction and it's up to every learning operator application what this inference results in.

Therefore, sources of variations in the system come from the fact that every individual can make their own changes to existing grammatical rules (inheritance), and introduce variation in the language of individuals within a population. On the other side, the choice of which variants of every piece of grammar are retained and become dominant in the population is based on linguistic selection criteria (selective pressures): for example in the cases of the thesis, those variants that allow speakers or hearers to keep or increase communicative success with less cognitive effort are preferred by the agents. Hence they are maintained in their language, used more, and spread faster.

In the approach I follow for the case of language, the units of evolution are grammatical rules and inheritance could be instantiated through the way language strategies reuse information of one rule to create another one. Moreover, not only grammatical rules could be considered as units of evolution but also language strategies themselves [15]. However, in the thesis I consider only fixed strategies for all the agents.

Therefore, learning operators for invention, adoption and alignment of a particular language strategy are the instantiation of the evolutionary mechanisms expressing the sources of variation and the selective pressures, and as they are defined and implemented at the agent level, they act locally but may provoke global behaviours as the emergence of linguistic conventions, e.g. phrase structure.

1.3.3 Computational Experiments

For languages strategies and language games, I introduce effective procedures for the agents as speakers and hearers and possibly run simulations in order to show whether or not a shared successful language system arises in computational experiments. This procedure is known as a computational model.

A computational model is a mathematical model meant to study the behaviour of a complex system. The analyses of such model require often being approached by computer simulations because analytical solutions are usually not reachable. In the thesis, the models are mostly studied through simulations, but I also tried to come up with analytical solutions as much as possible and successfully modelled some aspects, although this is not a frequent practice in the field of artificial language evolution.

Every computer simulation is called a computational experiment (of the model), and the whole experimentation turns around adjusting the parameters or turning on and off certain features of the model in the computer, and studying the differences in the outcome of the experiments. The outcome of an experiment is the product of systematic measurements taken all along the experiment. Finally, operation theories of the model can be derived from the study of these outcomes.

Computational models are used in various scientific fields including biology, sociology, physics and cognitive sciences. As explained earlier, the goal of this research program is ultimately to study the cultural evolution of language, and I propose and study computational models for specific aspects related to the emergence of phrase structure. The requirements for such models vary with respect to what it is the research question that they are meant to answer. In this research enterprise a significant emphasis is put in the communicative need that language solves, and that's why models are framed using the language game paradigm. When setting up a population of agents, the skills they are endowed with have namely to do with three things:

1. processing linguistic information
2. interacting with other agents
3. acquiring and modifying their own linguistic knowledge under certain conditions

The set of skills 3 is therefore modelled using learning operators for specific language strategies, and it is the set of processes where I put the main focus of investigation in the thesis. On the other side, while 2 and 1 remain equal across models and experiments. Skills 3 include all the processes involved in language learning, including how the linguistic knowledge is updated after every interaction.

The agents' skills together with the common assumptions of the language game models [88] conform the assumptions of the computational models. Moreover, these layers of skills are independent to each other in the models studied.

The machinery used to process language, the set of skills 1, i.e. in the models is based on the fully operational computational implementation of construction grammar known as Fluid Construction Grammar [94, 91]. This is a formalism for grammar representation and language processing, which was specially designed for the purpose of evolutionary experiments (see section 2.2).

Moreover, the mechanisms by which agents interact, skills 2, are governed by the language game script of the language game in use. So, agents have to be endowed with the necessary skills to perform the script of a language game. These consists of extra linguistic abilities, such as turn-taking, role assignment and non linguistic feedback, including pointing and nodding [17, 107, 60, 105].

On the other side, agents are not allowed to do things like mind reading or forcing other agents to perform a particular task, and the only means they have to communicate is their language.

Finally, I assume in all the models that all agents are equally likely to interact with any other agent, and that only a pair of agents interact at every time step. See [4, 24] for an study on how the structure of the population of agents affects the output of language game models.

1.3.4 Validation

The validation of empirical research requires access to the experimental data and statistical methods that were used in order to assure reproducible research. However, in the case of computer simulation, what is required is a description of the setups and measurements considered. The full description of the setups is of course in the code, but often when the complexity of the code is high, further demonstrations with illustrative examples are useful.

In this case, the complete code and simulated data are available for download in the site www.biologiaevolutiva.org/lsteels/tesi-emilia/tesi-final.zip, and documentation will be made available soon, so that researchers can keep on studying and improving the model.

Moreover, in www.biologiaevolutiva.org/lsteels/tesi-emilia/acq,
www.biologiaevolutiva.org/lsteels/tesi-emilia/ps
and www.biologiaevolutiva.org/lsteels/tesi-emilia/rps there is access to interactive web demonstrations for several aspects of the models proposed in chapters 4, 5 and 6, respectively.

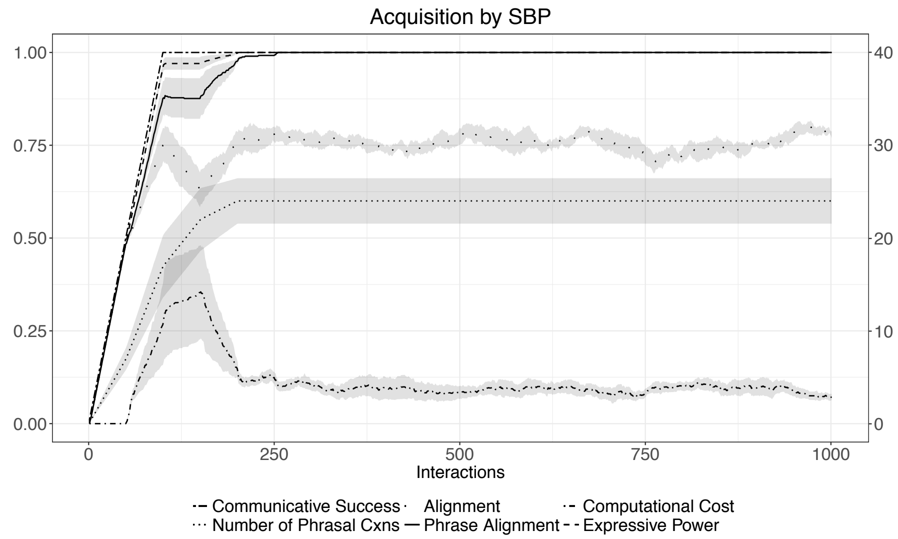


Figure 1.4: An example of the outcome of a computational experiment. Results for several measures are averaged on 5 independent runs and the gray ribbon shows its sample standard deviation.

The data is organised in files where the measurements taken on the experiments are stored. Below I define all the measures considered, and figure 1.4 shows an example of the output of a series of experiments. A sliding window of 100 interactions is used.

The **communicative success** (CS) of an interaction is 0 when the game is a failure and 1 when it is a success.

Alignment measures the consensus between agents. In every interaction it is checked which utterance would have the hearer used to express the same meaning. If both utterances are different, alignment gets the value 0, and if they are equal, it gets the value 1.

Phrase alignment is similar to alignment but instead of comparing whole sentences I compare phrases individually.

The **number of phrasal constructions** (Number of Phrasal Cxns) measures the number of phrase structure rules in a population.

Expressive power measures the ratio of variable equalities that are due to phrasal constructions. Chapters 2 and 3 discuss the necessary information to understand this measure.

Finally, the **computational cost** is derived from the search space to find the solution of a grammar application. It is defined as the ratio between the number of processing steps and the optimal number of processing steps, minus 1. Where the optimal number of steps equals the number of expected phrases. Again, chapter 3 introduces the notion of this measure.

1.4 Research Questions and Hypotheses

The main research questions of the thesis are about the cultural emergence of phrase structure and recursive phrase structure. The specific hypotheses supported follow:

RQ1: Could phrase structure emerge culturally as a result of a functional adaptation in language processing? H1: Yes, one of the functions of phrase structure is to reduce the cognitive effort required for language processing, particularly in the form of memory and computational power for semantic interpretation, and selective pressures on the computational cost of semantic interpretation helps phrase structure to arise in a population of communicative agents.

RQ2: Is there any extra fundamental operator required for recursive phrase structure to arise with respect to those operators required for first-order phrase structure? H2: No, fundamentally the same learning operators with natural extensions lead to successful results.

Generally, the objectives of this thesis are:

- Exploring cultural evolutionary mechanisms that could underlie the emergence of phrases and recursion on a lexical language and gain insights by building computational models.
- Providing evidence from the models and their analyses to support the hypothesis that fundamental features of grammar can be the result of cultural processes, by considering a case study on phrases and recursive phrases and their adaptive function on reducing the computational cost of semantic interpretation.

More concretely, the objectives of the thesis include several steps on the exploration of the research questions as well as on the nature of the mechanisms involved in their related phenomena:

1. Defining a formal framework to approach the investigation on the emergence of phrases and recursion.
2. Analysing the computational complexity of language processing in semantic interpretation in order to justify the choice of selection on the reduction of cognitive effort as a selection criterion.
3. Exploring evolutionary mechanisms for the cultural evolution of artificial languages requires:
    1) *hypothesizing*: Describing specific mechanisms hypothesized to be sufficient for the emergence of phrases and recursion
    2) *implementing*: Implement specific operators as models for mechanisms and integrating them into an operational language game model.

    3) *testing*: Performing computer simulation to validate the behaviour and efficacy of the model, and the hypothesized mechanisms, e.g. the selection criteria.

4. Characterizing the kinds of languages that can emerge given the operators implemented.

5. Analysing scaling properties of the computational models as well as their possible extensions.

1.5 Contributions

The key scientific result of this thesis is

> The demonstration that selective pressures (forces) on the
> computational cost of language processing can trigger
> the conventionalization of phrase structure grammars
> in lexical systems

This thesis contributes to the field of evolutionary linguistics, and particularly to the computational modelling of cultural language evolution, in two different aspects: the technical and the conceptual.

Concerning the technical advances, many of the features developed in order to implement the models proposed provide new possibilities to the field and push the boundaries of its coverage. Moreover, some of them are also coupled to new concepts. Examples of the most relevant ones are:

1. The implementation of two new language games. See chapter 2.

2. The integration of parsing and interpretation processes in one. See chapter 4.

3. The integration of language processing and language learning. See chapter 4.
4. New visualizations for the internal structure of grammars and their usages. See chapters 3, 4, 5 and 6.

Concerning the conceptual contributions, the main contribution is the first computational model of the cultural emergence of recursion where no prior syntactic categories (lexical or phrasal) are assumed. The model is based on the mechanisms proposed for the emergence of first-order phrase structure (non-hierarchical), including computational pressures.

Other important conceptual contributions are:

1. The demonstration that one of the functions of phrase structure is to reduce the computational complexity. See chapter 3.
2. Fully-aligned agents don't necessarily use the same syntactic categorizations.
3. Agents are able to self-organise grammars which use syntactic categorizations that are more efficient than semantic categorizations in terms of computational resources needed to encode a fully communicative grammatical system. See chapters 4, 5 and 6.
4. Agents evolve grammars that induce a notion of syntactic head. See chapter 6.
5. The validation of a cognitive architecture for grammar learning which supports Insight Problem Solving. See chapter 7.

1.6 Outline and Scope of the Thesis

The remainder of this thesis after this introductory section is organised as follows:

Chapter 2 introduces the formal framework that is used along the thesis, including specific formal concepts for grammars and semantic representation, and it describes some examples.

Chapter 3 is the first out of four chapters which overview original contributions of the thesis, and extend them. Within the framework introduced in chapter 2, it compares the complexity of several language strategies in the task of semantic interpretation, and uses it as a measure of cognitive effort along with other measures. Finally, results lead to evidence for the reduction of the computational cost of semantic interpretation as a functional adaptation on phrase structure. This implies a first step towards the construction of a computational model to study the thesis hypotheses. Results of this chapter have been published in [97, 50].

Chapter 4 develops a model which accounts for the acquisition of a language system exhibiting phrase structure. In this model, one agent acts as teacher and speaker, and the other as learner and hearer. Specific learning operators for the hearer are explored demonstrating that phrase structure can be acquired by the means of adoption operators for contextual inference, syntactic coercion and ordering variation. The two articles resulting form this chapter have been published in [51, 98].

Chapter 5 demonstrates that a population of artificial agents such as the learner in the previous chapter are capable to self-organise a language system exhibiting first-order phrase structure. Specific mechanisms in the form of communicative and computational pressures are identified and their performances assessed through computational experiments. Moreover, the results and assumptions are compared to the case of lexicon formation using Naming Game models, finding that the mechanisms to self-organise a lexical system are not enough for a language system exhibiting phrase structure, and that computational pressures on the reduction of semantic ambiguity gives a solution for the formation of the latter. Results of this chapter have been published in [97, 98, 50, 47].

Chapter 6 builds further on the model proposed in chapter 5 to explore how higher-order and recursive phrase structure can be achieved. It shows that no fundamental additional mechanisms are required to achieve that goal. The article resulting from this chapter is not yet published while some results have been partly published in [48], and more will be published soon in [99].

Finally, chapter 7 discusses the main results of the thesis in the light of the research questions, it concludes this work by summarizing results presented in chapters 3 - 6 and contrasts or complements them to previous studies. It discusses strengths as well as drawbacks and limitations of the proposed approach, and provides guidelines and perspectives for future research, together with some final remarks.

Chapter 2

FORMALISATION

In this chapter I formalise the approach I follow in the thesis to study the problem of conventionalization of a language system exhibiting phrase structure.

I study the problem using language game models and language strategies to solve them, possibly leading to resulting language systems exhibiting phrase structure. In the first section, I introduce the syntax games, which are the language games that are used. These language games are studied all along the thesis and tested together with a particular language strategy that solves them forming a solution to the problem of conventionalization of phrase structure.

In the second section I present the grammar formalism and grammar engine that are used to model language processing and discuss the particular cases of phrase structure grammars and recursive phrase structure grammars. Finally, I sum up all the concepts introduced and explain how the grammar formalism is integrated to a syntax game interaction, illustrating it with an example.

2.1 Syntax Game

Syntax games are characterized by the fact that they assume a given lexical system already conventionalized by the population of language users, and build grammatical systems on top of this lexical system.

Concretely, I propose syntax games as language games meant to explore the origins of phrase structure. I define the games, putting special emphasis to the meaning representations and to the lexicon that is given to the agents.

The Syntax Game is a game of reference, either in the form of a multi-referential game (description) or a discrimination game. In a description game the speaker expresses a partial description of the situation to the hearer, and in the discrimination game the aim of the speaker is to draw the attention of the hearer to an entity or event in the situation. The situation is shared by both agents and a common ontology is initially provided. Therefore, the situation, i.e. the communicative environment, and the given lexicon and ontology play a central role in a syntax game. See figure 2.1 for an example of a discrimination game.

The Syntax Game involves the following steps:

1. **Communicative goal:** The speaker selects a communicative goal from his situation model to convey it to the hearer. In the case of discrimination games, he selects an entity or event, which will be referred to as the topic of the situation, and in description games, a partial description of the situation.

2. **Formulation:** The speaker identifies what meaning distinctively describes the topic (discrimination), or conforms the desired partial description (description). The speaker uses his own lexicon and grammar to translate this meaning into an utterance (production). This utterance is then transmitted to the hearer.

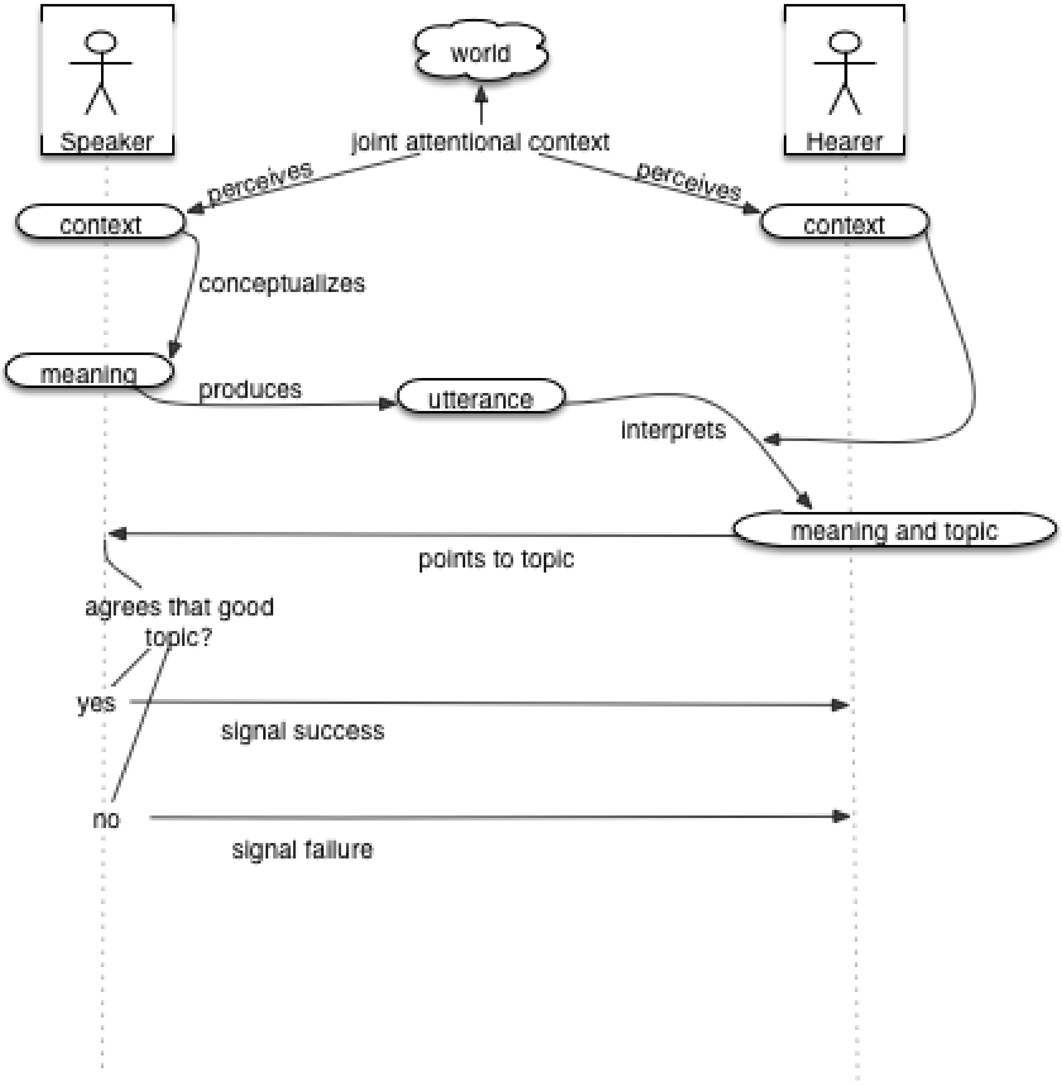


Figure 2.1: Diagram of a discrimination language game. Agents can perceive only the elements represented in the middle, i.e. the shared context, the utterance, the pointing and the feedback.

3. **Comprehension:** The hearer parses this utterance using his own lexicon and grammar in order to reconstruct a possible meaning to interpret it in terms of his own situation model, and to find out what topic or partial description the speaker intended.

4. **Pointing:** In a discrimination game, the hearer signals to the speaker which is the topic he interpreted. In a description game (multi-referential), he signals the object or objects in the situation that are described in the partial description he interpreted.

5. **Feedback:** The speaker signals success if the topic/topics identified by the hearer is/are the same as the topic originally chosen by the speaker. If they differ, the speaker signals failure and also points to the original goal.

6. **Learning:** Both speaker and hearer then expand and align their lexicon and grammar based on the outcome of the game.

Therefore, formulation includes the conceptualisation of the meaning to produce, and the production of an utterance expressing this meaning. And comprehension includes the parsing of an utterance to identify a possible meaning (situation independent) and the interpretation of this meaning in the situation (situation dependent). As it is shown in chapter 4, both processes in comprehension are combined.

A language game models the interaction between two individuals of the same language community. In a syntax game interaction, both agents are assumed to maintain a model of the current situation. In computer simulations, this situation model is synthesized based on an ontology of possible predicates, in which both situation models and utterance meanings are based. Particularly, an agent's lexicon directly maps word meanings to predicates of the ontology. The motivation for this design choice is to use a minimal model for word meanings with potential semantic compositionality, where semantic compositionality implies that the meaning of a complex expression (more than one word) is a function of the meaning of its constituents. The remaining of this section describes the ontology in detail, and its connection to word meanings in the lexicon.

2.1.1 Ontologies, Situations and Meanings

In the models of the thesis, ontologies are represented using a variant of higher-order typed predicate calculus, where every predicate has at least one argument. Arguments get the form of variables, which are represented as symbols with a question mark in front, whenever they represent abstract meanings in language processing, e.g. *?ball-1*; and they get the form of constants whenever they represent facts in the situation, e.g. *ball-1*. Therefore, semantic interpretation (or simply interpretation) is modelled as the binding of variables to constants, i.e. linking abstract meanings to actual facts. For example, given the situation: {(a-1 p-1-1 o-22), (a-2-p-2-1 o-30), (a-1-p-1-2 o-14)}, the meaning {(a-1 p-1-1 ?o-1), (a-2-p-2-1 ?o-2), (a-1-p-1-2 ?o-3)} has one unique interpretation, which can be represented by the set of bindings {(?o-1 o-22), (?o-2 o-30), (?o-3 o-14)},where each variable is bound to the constant on its right.

I use symbols as *ball-1* to illustrate examples in the ontology. However, I use arbitrary symbols in the models (see figure 2.2) so that I can modify the number of predicates.

Predicates are represented as predications in the form *type(predicate, entity)* or as triples as *(type predicate entity)*. For example, *physical(ball, o-1)* or *(physical ball o-1)* represents that *physical* is the type of the property *ball*, *ball* is a predicate and *o-1* is an object in the situation. Therefore, object properties are represented as predicates of one argument. This is the same ontological model that was used in [11] in the design of a computational model for the emergence of agreement systems.

Moreover, predicates which have more than one argument are models for relations, and they are represented as sets of predications, one for each argument in the relation, and one for the relation. For example, *spatial(on, rel-9) on-arg-1(rel-9,ball-3) on-arg-2(rel-9, table-5)* represents that *spatial-relation* is the type of the predicate *on* and *rel-9* is the object denoting that relation. The relation *on* has two arguments: *on-arg-1* and *on-arg-2*. The types of the arguments are derived from the symbol representing the relation, e.g. *on*, and the fillers are denoted again with predictions using the constant (or variable) for the relation as a predicate itself, so that relations are reified [72] to predicates.

Situations are represented by sets of predications of constants, where each constant is identified to an instance of an object or event in the situation. Abstract meanings are represented by sets of predications of variables, and these variables are made equal when they have to be interpreted as the same constant in the situation.

Hence, interpreting a meaning {*(a-3 p-3-4 ?o-2)*} in the situation in figure 2.2 (bottom) results into two possible sets of bindings, {(?o-2 o-68698)} and {(?o-2 o-68697)}, where each of them consists of one element because a single predicate of one argument was considered). Whereas interpreting a meaning {*(a-3 p-3-4 ?o-2), (r-4 r-4-3 ?r-1), (r-4-3-arg-1 ?r-1 ?r-3*, (r-4-3-arg-2 ?r-1 ?o-2), (r-2 r-2-1 ?r-3), (r-2-1-arg-1 ?r-3 ?o-4), (r-2-1-arg-2 ?r-3 ?r-2), (b-2 q-2-1 ?r-2), (q-2-1-arg-1 ?r-2 ?o-5)} results into a single interpretation given by the set of bindings {(?o-2 o-68698), (?r-1 0-68693), (?o-4 o-68695), (?r-2 o-68696), (?o-5 o-68697), (?r-3 o-68694)}.

2.1.2 Lexicon

Every agent in a syntax game model is initialized with a lexicon. This lexicon has to account for semantic compositionality. Using the ontology introduced in the previous section, this happens when word meanings are properties, i.e. predicates of one argument, or relations, i.e. predicates of two or three arguments. Notice that predicates of two arguments are models for unary relations and predicates of three arguments are models of binary relations, this is because the relation itself is represented as an argument.

The examples in figure 2.3 are lexical constructions represented using the grammar formalism of Fluid Construction Grammar (FCG), which is

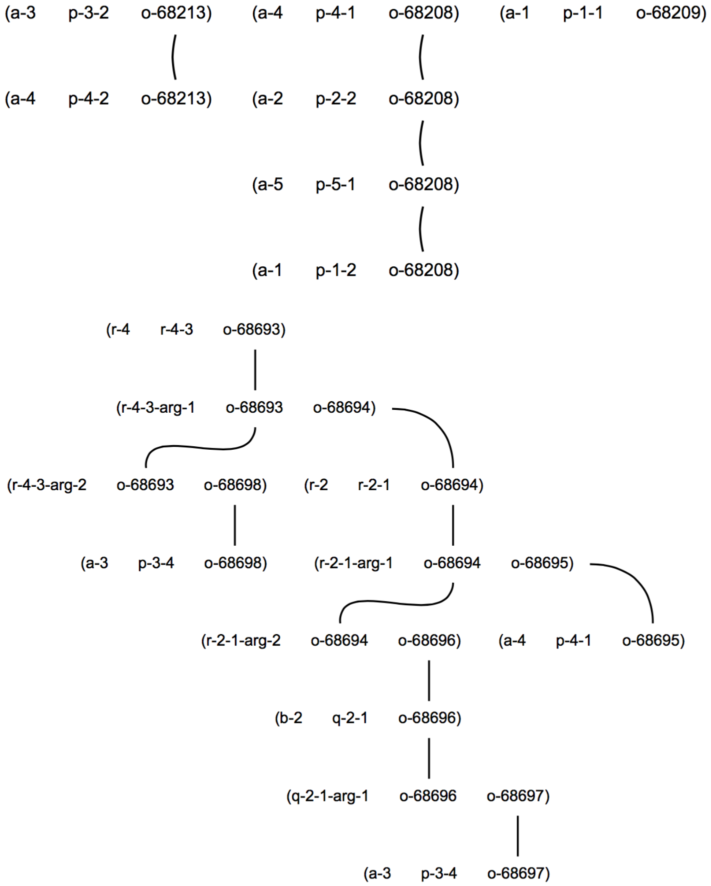


Figure 2.2: Each node is a predicate and edges represent constant (or variable) equalities. Top: Part of a situation including uniquely predications of one argument. Bottom: Part of a situation including predications of one to three arguments.

$$\left[\begin{array}{l} \textbf{?rova-1} \\ \hline\hline \text{referent: ?o-1} \\ \text{potential-syn-cat: } \{\} \end{array}\right] \leftarrow$$

$$\left[\begin{array}{l} \textbf{?rova-1} \\ \hline\hline \text{\# meaning: } \{\text{a-2(p-2-4, ?o-1)}\} \\ \hline \text{\# form: } \{\text{string(?rova-1, rova)}\} \end{array}\right] \qquad \text{(rova-cxn)}$$

$$\left[\begin{array}{l} \textbf{?fareza-1} \\ \text{args: [?o-194, ?o-195, ?o-196]} \\ \text{referent: ?ref-35} \\ \text{potential-syn-cat: } \{\} \end{array}\right] \leftarrow$$

$$\left[\begin{array}{l} \textbf{?fareza-1} \\ \hline\hline \text{\# meaning: } \{\text{r-4(r-4-1, ?o-194), r-4-1-arg-1(?o-194, ?o-195), r-4-1-arg-2(?o-194, ?o-196)}\} \\ \hline \text{\# form: } \{\text{string(?fareza-1, fareza)}\} \end{array}\right] \qquad \text{(fareza-cxn)}$$

Figure 2.3: Examples of lexical constructions represented using FCG (formalism introduced in the coming section). Top: Lexical construction for the property *a-2(p-2-4,?o-1)*. Bottom: Lexical construction for the binary relation *r-4(r-4-1, ?o-194), r-4-1-arg-1(?o-194, ?o-195), r-4-1-arg-2(?o-194, ?o-196).*

introduced in the coming section. Lexical constructions are models for the lexical rules in the lexicon.

Agents playing syntax games require supporting language processing for production (meaning to utterance) and for parsing (utterance to meaning). Therefore, agents' grammars should support rules able to map aspects of the meaning to aspects of the utterance, and viceversa.

2.2 Fluid Construction Grammar

Many linguists model mappings of meaning, form, and lexical or grammatical categorizations as constructions which pack relevant information for the mapping. This is the central idea of the linguistic theory of Construction Grammar [39, 55, 54, 56], and it is the one I adopt in the thesis. In this theory, constructions include information about both syntax and semantics, differently than in the generative grammar where the focus is only on syntax, and in which semantics is assumed as being introduced separately by translation rules [18].

Several theories of Construction Grammar have been proposed [22, 9], and also several computational formalisms have been developed to operationalize language processing in terms of such constructions [73]. The linguistic framework I follow in the thesis is the one defined by Fluid Construction Grammar (FCG), a variation of Construction Grammar which counts with a fully-operational implementation. FCG is a unification-based grammar formalism, as it is Head-Driven Phrase Structure Grammar (HPSG) [80].

Language processing studies how to derive a meaning from an utterance or how to build an utterance to express a meaning. Most linguistic theories are only concerned with the latter, i.e. parsing an utterance, to get a syntactic structure and deriving a meaning from it. In contrast, FCG is concerned as well with the inverse process, i.e. production, where the goal is to build an utterance to express a meaning.

I use FCG for the representation of grammars as sets of constructions [94] and for the grammar engine that allows the use of constructions to map between symbolic meanings and utterances, in the process of production; and between utterances and symbolic meanings, in the process of parsing. FCG is specially designed to implement experiments for cultural language evolution, although because:

- FCG constructions can be used in both directions (production and parsing), and therefore, the same construction can be used by an agent acting as speaker and acting as a hearer

- construction applications are open, meaning that FCG accounts for constructions that can be used in more than one way, which is needed in order to facilitate finding the syntactic structures required
- and finally, because it allows the co-existence of competitor constructions, implying that during the process of conventionalization it can keep as much information as possible even if the linguistic knowledge is incomplete or inconsistent.

In this section I explain some relevant properties of FCG and describe the kinds of grammars used in the thesis.

2.2.1 Language Processing in FCG

FCG views language processing in terms of a search problem to find a solution syntactic structure. Nodes (syntactic structures) are represented as transient structures and transitions as construction applications, i.e. transformations of transient structure due to the application of constructions (one per each transformation). A transient structure captures all that is known about a particular utterance being parsed or produced, and it is used as an extended model for a syntactic structure. Therefore, in language processing, transient structures are expanded by the application of constructions, and their applications consist of two sub-operations, the operation of matching, to see whether the construction can apply, and the operation of merging, to add information from the construction to the transient structure.

FCG represents transient structures in terms of feature structures, similar to many other computational formalisms in use today, such as HPSG. These feature structures are organised in units, and every unit encapsulates different information. See figure 2.4.

In constructions, FCG distinguishes between two types of units according to whether they are used for matching or for merging, and they are called conditional and contributing units, respectively. Constructions are

$$\left[\begin{array}{l}\textbf{?unit-1}\\ \hline\hline \text{feature-1: value-1}\\ \text{feature-2: }\{\text{value-2, value-3}\}\\ \text{feature-3: value-4}\end{array}\right] \quad \left[\begin{array}{l}\textbf{?unit-2}\\ \hline\hline \text{feature-1: value-5}\\ \hline \text{feature-2: }\{\text{value-6, value-2}\}\end{array}\right]$$

Figure 2.4: In FCG, transient structures are organised in units, and every unit has a set of features with values encoding relevant information for language processing. The figure shows an example of a transient structure (TS) of two units.

written so that contributing units come first, followed by an arrow to the left, and conditional units are found in the right side of the arrow.

The order in which features appear within units in a transient structure is irrelevant to the application of a construction.

The constructions used in the thesis always include a contributing unit which creates a new unit in the transient structure, and possibly other contributing units which add information to existing units. Lexical constructions create lexical units, and phrasal constructions (models for phrase structure rules) create phrasal units. Here I illustrate examples of construction applications for an agent parsing the utterance *zegue tequale*:

The two lexical constructions below...

$$\left[\begin{array}{l}\textbf{?zegue-1}\\ \hline\hline \text{referent: ?o-1}\\ \text{potential-syn-cat: }\{\text{syn-cat-36, syn-cat-33}\}\end{array}\right] \leftarrow$$

$$\left[\begin{array}{l}\textbf{?zegue-1}\\ \hline\hline \text{\# meaning: }\{\text{a-1(p-1-2, ?o-1)}\}\\ \hline \text{\# form: }\{\text{string(?zegue-1, zegue)}\}\end{array}\right] \qquad \text{(zegue-cxn)}$$

$$\left[\begin{array}{l}\textbf{?tequale-1}\\ \hline\hline \text{referent: ?o-1}\\ \text{potential-syn-cat: }\{\text{syn-cat-35, syn-cat-38}\}\end{array}\right] \leftarrow$$

$$\left[\begin{array}{l}\textbf{?tequale-1}\\ \hline\hline \text{\# meaning: }\{\text{a-3(p-3-2, ?o-1)}\}\\ \hline \text{\# form: }\{\text{string(?tequale-1, tequale)}\}\end{array}\right] \qquad \text{(tequale-cxn)}$$

apply to create the following two lexical units:

tequale-6

referent: ?o-89356
potential-syn-cat: {syn-cat-38, syn-cat-35}
meaning: {a-3(p-3-2, ?o-89356)}
form: {string(tequale-3, tequale)}

zegue-6

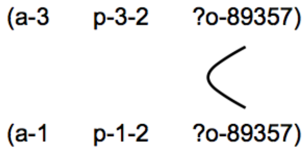
referent: ?o-89357
potential-syn-cat: {syn-cat-36, syn-cat-33}
meaning: {a-1(p-1-2, ?o-89357)}
form: {string(zegue-3, zegue)}

Then, a phrasal construction applies to make the variable *?o-89356* in unit *tequale-6* equal to *?o-89357* and to create a phrasal unit on top of the two previous lexical units with the following features:

np-unit-18

form: {meets(zegue-6, tequale-6)}
subunits: {zegue-6, tequale-6}
referent: ?o-89357
boundaries: {zegue-6, tequale-6}

Therefore, given the variable equalities, the resulting meaning of the structure can be represented as:



Recall that in an FCG construction representation, the left side of the arrow is called the contributing side, and the right side of the arrow is called the conditional side. The line in the units of the conditional side separates the production lock and the comprehension lock. The production lock is used to match in production and the comprehension lock is used to match in comprehension (i.e. in parsing). Moreover, the features in the production lock are merged to the corresponding unit in the parsing process when the comprehension lock matches, while the features in the comprehension lock are merged to the corresponding unit in the production process when the production lock matches.

Notice that both production and comprehension locks of the lexical constructions above have the sign # at the beginning of the features. This is a special notation to represent that the following features shouldn't be found in the unit itself but in the set of features computed at the beginning of a grammar application and put into a special unit which encodes information on the meaning to produce (in production) or on the utterance to parse (in parsing). This is just an implementation issue and it is done by the FCG-interpreter. In the context of FCG, this special unit is called *root unit*. Hence, according to the previous lexical construction representations, when a lexical construction applies in parsing:

1. it targets to match a certain string in the *form* feature of the root unit. In the previous example, one lexical construction targets the string *zegue* and the other, the string *tequale*.

2. if it matches, it removes that information from the root unit and uses it to create a new lexical unit. In the previous example, each lexical construction creates a lexical unit with the corresponding string information in the *form* feature.

3. the content in the production lock is also added (merged) to the new unit. In the previous example, the *meaning* feature of the production lock is added to the new unit.

4. finally, the content of the contributing units (left side) is also added to the new unit. In the previous example, the features *referent* and *potential-syn-cat* are added to the new unit.

When grammar application in parsing is finished, the FCG-interpreter translates FCG structures into a meaning expression by using information from the units.

Inversely, when a lexical construction applies in production:

1. it targets to match certain meaning and it looks for it in the root unit

2. if it matches, it removes that meaning from the root unit and uses it to create a new lexical unit

3. the content in the comprehension lock is also added (merged) to the new unit

4. finally, the content of the contributing units (left side) is also added to the new unit.

And when grammar application is finished, the FCG-interpreted translates FCG structures into utterances by using information from the units.

Grammar applications for both production and parsing are accomplished by the sequential application of constructions until the final transient structure that the FCG-interpreter uses to get an utterance or a meaning, respectively, is computed.

As mentioned before, there are two types of constructions in the cases studied in the thesis: lexical constructions (see figure 2.3), that create lexical units out of words or meaning predicates; and phrasal constructions, that create phrasal units out of phrases in an utterance or variable equalities in meanings. These constructions are defined in the next sections and some examples are examined.

Therefore, in the context of this thesis agents' grammars consist of lexical constructions, phrasal constructions and a grammar engine to allow language processing as explained before.

2.2.2 Phrase Structure Grammars in FCG

In the framework described before, first-order phrase structure is motivated to express the co-referentiality of properties of the same object.

In the models of the thesis, the grammar engine that agents use has to support phrase structure grammars, similarly than agents playing naming games have to be able to store lists of word-meaning pairs.

In the previous section I presented an example of the application of a phrase structure grammar in FCG without explaining the details, in the present and next sections I describe in depth the kinds of constructions I use to represent these grammars, and their features. First, I study the case of first-order phrase structure, and after it, the case of higher-order phrase structure and recursive phrase structure.

As I said earlier, the way I represent phrase structure grammars in FCG consist of two types of constructions: lexical and phrasal. Lexical constructions essentially associate information between meaning predicates and word form, and they have a potential use of categorizations, and phrasal constructions combine lexical and phrasal units by the means of their categorizations, and make equal some specific variables in their meanings (in parsing) or impose a particular order to the phrasal constituents (in production).

All lexical constructions are constructions mapping word forms to predicates on the ontology, and they support the use of lexical categorizations through the feature *potential-syn-cat*. In the *zegue-cxn* used earlier (see figure 2.5), two lexical categories were introduced into the new lexical unit: *syn-cat-36* and *syn-cat-33*. And these categories, in turn, allow or prevent the application of phrasal constructions matching on those units.

First-order phrasal constructions associate a pattern of lexical units by

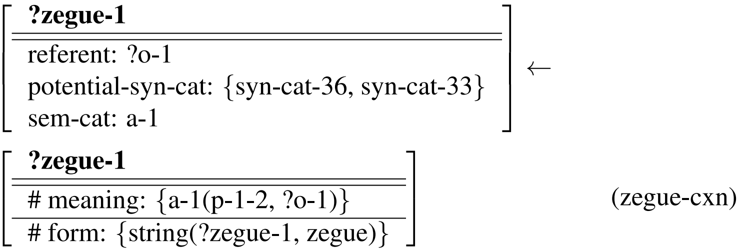
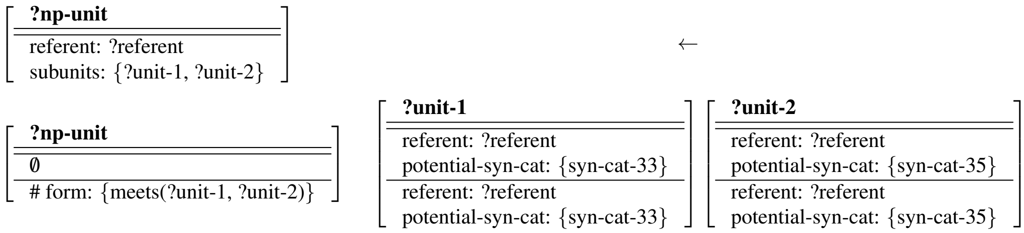


Figure 2.5: Example of a lexical construction using the FCG formalism. Lexical constructions match on meaning in production an on words in parsing.

their lexical categories and impose a relative word order when they are co-referential (in production) or impose co-referentiality when they are placed in the order required for the construction matching (in parsing). Lexical categorization is represented by the means of the feature *potential-syn-cat*); relative word order, by the means of the value *meets* in the feature *form*, which indicates that the word forms (i.e. strings) on the two lexical constructions involved are placed in the corresponding order in the utterance; and co-referentiality is represented by the means of the feature *referent*, which indicates the variable or constant that the corresponding unit refers to.

Therefore, first-order phrasal constructions can be identified to a sequence of lexical categories. E.g. in the construction in figure 2.6, the most left unit of the pattern targets a *syn-cat-33* category, and the sequence *(syn-cat-33 syn-cat-35)* can identify the construction. I refer to this sequence as construction form.

Finally, there are two features in the contribution side of the first-order phrasal constructions, the feature *referent*, to state that the referent of the whole phrase indeed equals the referent of all of its constituents, and the *subunits* feature, which identifies the lexical units of its value as children


?np-unit
referent: ?referent
subunits: {?unit-1, ?unit-2}

←

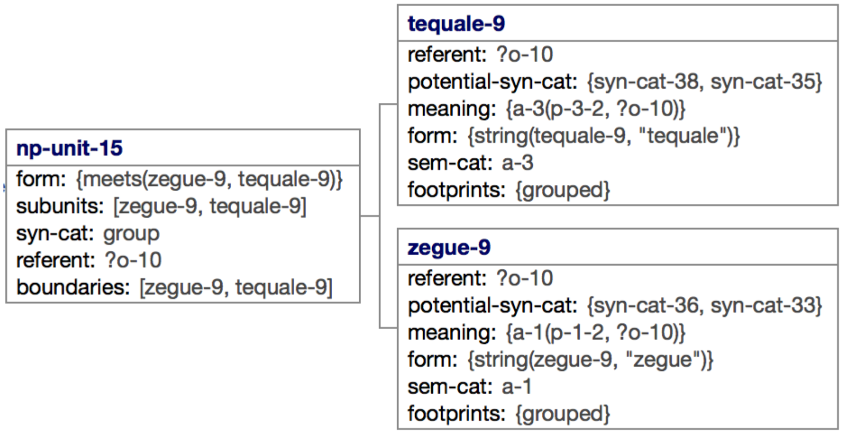
?np-unit
∅
form: {meets(?unit-1, ?unit-2)}

?unit-1
referent: ?referent
potential-syn-cat: {syn-cat-33}
referent: ?referent
potential-syn-cat: {syn-cat-33}

?unit-2
referent: ?referent
potential-syn-cat: {syn-cat-35}
referent: ?referent
potential-syn-cat: {syn-cat-35}

Figure 2.6: Example of a single-noun phrasal construction using the FCG formalism and the features that are relevant for the grammars explored in the thesis.

units of the phrasal unit. See figure 2.7 for an example.


np-unit-15
form: {meets(zegue-9, tequale-9)}
subunits: [zegue-9, tequale-9]
syn-cat: group
referent: ?o-10
boundaries: [zegue-9, tequale-9]

tequale-9
referent: ?o-10
potential-syn-cat: {syn-cat-38, syn-cat-35}
meaning: {a-3(p-3-2, ?o-10)}
form: {string(tequale-9, "tequale")}
sem-cat: a-3
footprints: {grouped}

zegue-9
referent: ?o-10
potential-syn-cat: {syn-cat-36, syn-cat-33}
meaning: {a-1(p-1-2, ?o-10)}
form: {string(zegue-9, "zegue")}
sem-cat: a-1
footprints: {grouped}

Figure 2.7: The feature *subunits* introduces a hierarchy in the transient structure by identifying the lexical units bound in the value of the feature as children units of the corresponding phrasal unit. This figure shows the resulting transient structure after parsing the utterance *zegue tequale* discussed in the text.

Hence, in this case the conception of phrasal constructions can be simplified as a sequence of syntactic categories for matching. Figure 2.8 illustrates this idea in a manner that is used along the thesis. Syntactic categories are represented as shapes in the sides of a constructions and I use the term binding side for their combination, as a metaphor for the binding sites in proteins.

So, phrasal constructions can be represented as a set of context-free rules as $S \rightarrow cat_1 cat_2 cat_3$ each rule mapping onto a construction. However, this formalisation is not sufficient to state the construction meaning, the variable bindings provoked (required) after (before) construction applications.

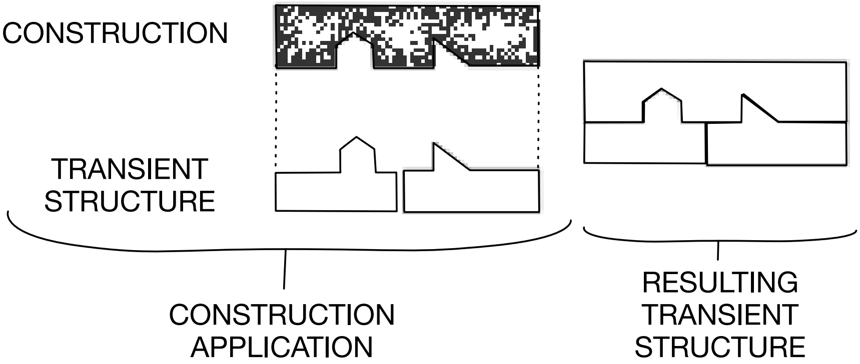


Figure 2.8: Illustration of a phrasal construction application. Lexical categories are represented as binding sides from a planar shape and they need to fit into slots of the phrasal construction (top). When this happens, a new unit containing the two original lexical units is created.

2.2.3 Recursive Phrase Structure Grammars in FCG

Higher-order phrase structure rules are phrase structure rules for which at least one of the conditional units is a phrasal unit itself. Phrasal constructions associate a sequence of units by their *syntactic categories* (lexical or phrasal categories), impose/require constituent orders or argument co-referentiality, and assign a phrasal category to the unit that they create.

Moreover, every lexical construction maps a word either into the predication of a property, in the case of an attribute word, or into the set of predicates of a relation and its arguments in the case of a relation word. Therefore the only lexical construction that looks different than before is the one corresponding to a relation.

$$\begin{array}{l}\left[\begin{array}{l}\textbf{?word-unit}\\ \hline\hline \text{args: [?x, ?y, ?z]}\\ \text{referent: ?x}\\ \text{potential-syn-cat: }\{\text{transitive-verb}\}\\ \text{sem-cat: spatial-rel}\end{array}\right] \leftarrow \\ \left[\begin{array}{l}\textbf{?word-unit}\\ \hline\hline \text{\# meaning: }\{\text{spatial-rel(on, ?x), on-arg-1(?x, ?y), on-arg-2(?x, ?z)}\}\\ \hline \text{\# form: }\{\text{string(?word-unit, on)}\}\end{array}\right] \quad \text{(on-cxn)}\end{array}$$

The value of the feature *args* consists of the variables introduced by the predications in a particular order.

Every phrasal construction application introduces a phrase on top of other units, and these units become the constituents of the new phrase. I model higher-order phrasal constructions as patterns where there is only one lexical constituent among the constituents.

Figure 2.9 gives an example of schematic phrasal construction applications.

In parsing, the phrasal constructions match on constituent order constraints and constraints on the categories. And in production, they use argument bindings between the arguments introduced by the lexical item and the referents of the phrasal constituents.

In figures 2.10 and 2.11 two phrasal constructions are represented, again highlighting the differences with respect to the case of first-order phrase structure. The values of the feature *boundaries* are used to express the units that are placed at the leftmost and rightmost sides of a phrase.

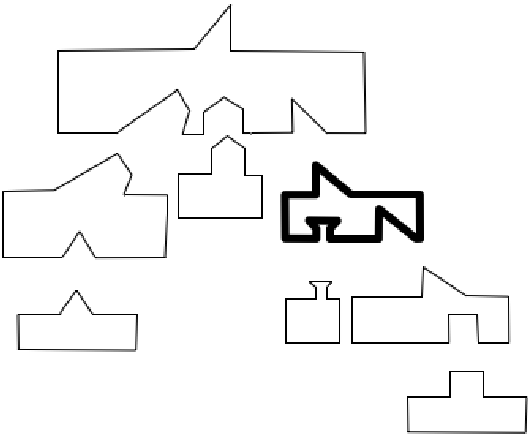


Figure 2.9: An illustrative example of the application of phrasal constructions. In this case argument bindings and referents are omitted. Each shape represents a construction, where the bottom line has spaces for syntactic categories to match on it and the top line has the phrasal category added by the construction. Constructions with a straight line in the bottom represent lexical constructions and the highlighted construction is an example of a recursive phrasal construction.

$$\left[\begin{array}{l} \textbf{?phrasal-unit-1973} \\ \hline\hline \text{referent: ?var-2891} \\ \text{syn-cat: np-2} \\ \text{boundaries: [?unit-2891, ?unit-2891]} \\ \text{subunits: [?unit-2891]} \end{array}\right] \leftarrow$$

$$\left[\begin{array}{l} \textbf{?unit-2891} \\ \hline\hline \text{referent: ?var-2891} \\ \text{potential-syn-cat: }\{\text{noun}\} \\ \hline \text{potential-syn-cat: }\{\text{noun}\} \end{array}\right]$$

Figure 2.10: An example of a single phrasal construction in FCG, i.e. the phrasal unit that is created has only a single subunit. Hence, the two units in boundaries are the same.

This time a simplified representation for phrasal constructions can be given by

$$\begin{gathered}\text{syn-cat-14} \leftarrow \text{+syn-cat-21}_1 \\ \text{syn-cat-32 syn-cat-23}_2\end{gathered}$$

for a phrasal construction which builds a phrase on top of three constituents. Every slot corresponds to a constituent, and they are placed in the imposed relative order. The constituent from which the new phrase will inherit the referent here is the first constituent and it is indicated by the + sign in the front, and the argument bindings are represented using subindexes.

Recursion occurs when the new phrasal unit has the same category as one of its constituents, then the corresponding construction would be a recursive phrasal construction. However, recursion have other meanings also in the context of phrases, which are discussed in section 6.1.

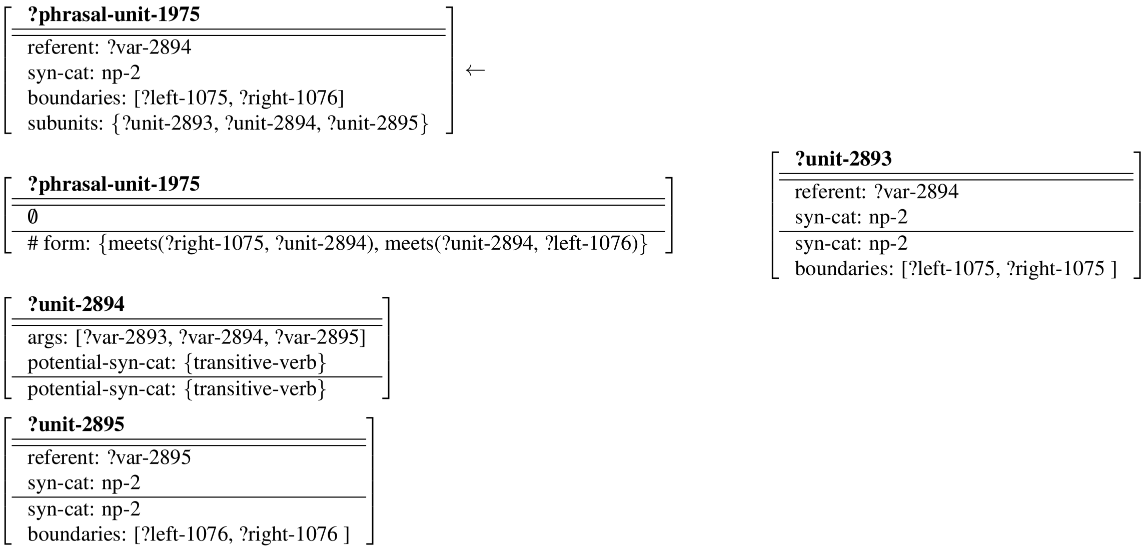


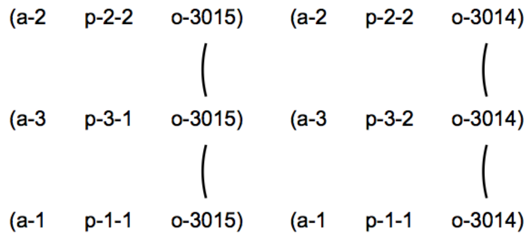
Figure 2.11: Every lexical unit has one or more arguments and one referent; and every phrasal unit, one referent and no arguments. In the case of phrasal units the referents are bound to the entity or relation which the phrasal constituent refers to; and in the case of lexical units for relations, they equal the referent of the phrase when this is assigned. The second and third values in *args* of unit **?unit-2894** are equal to the values in *referent* of the units **?unit-2893** and **?unit-2895**, which are phrasal units. Moreover, the referent of the former is chosen as the referent of the created phrasal unit.

2.3 Experimental Framework

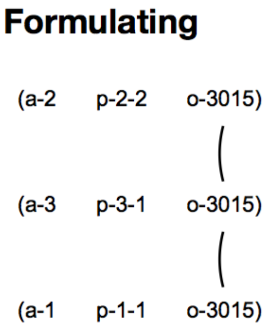
In this section I overview a syntax game interaction between two agents endowed with a first-order phrase structure grammar as the one described in section 2.2.2. An interactive web demonstration with the interaction can be found in the site www.biologiaevolutiva.org/lsteels/tesi-emilia/syntax-game-interaction

As I mentioned earlier, I use the Babel2 Framework for the implementation of the language game script and further functionalities for visualizing and storing resulting data.

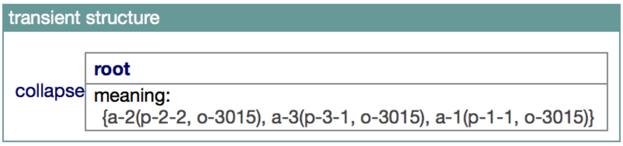
First, the two agents are placed in a shared context,



as they are playing a description game, the agent assigned the role of speaker selects a part of the meaning to express.



The formulation (production) process starts with the translation of this meaning to the initial transient structure by the FCG-interpreter

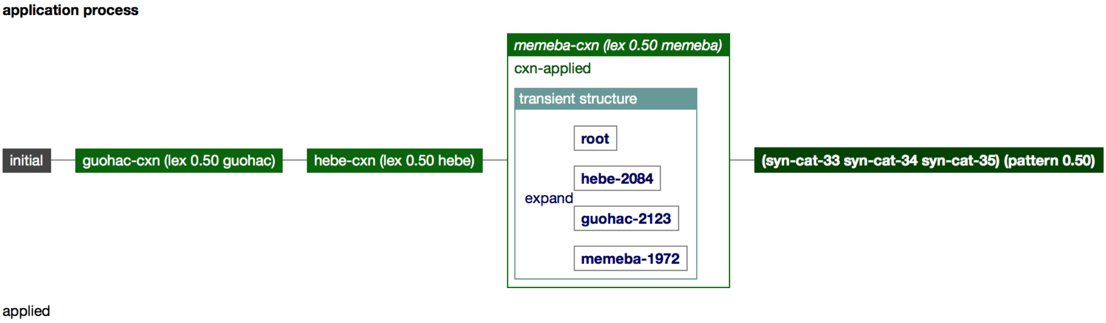

transient structure

collapse

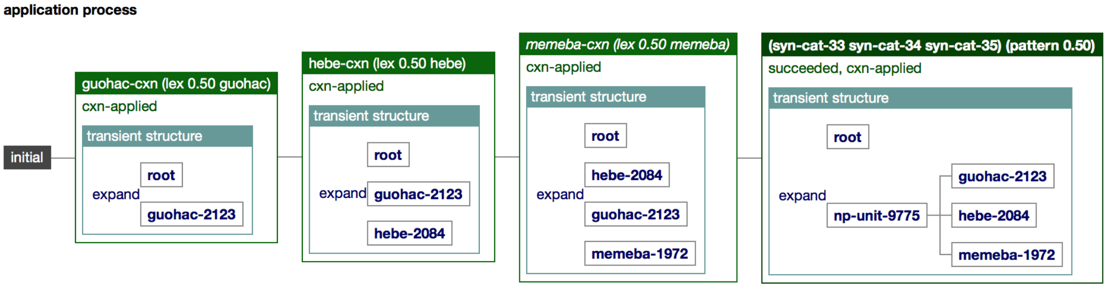
root

meaning:
{a-2(p-2-2, o-3015), a-3(p-3-1, o-3015), a-1(p-1-1, o-3015)}

and then it consists of a set of processing steps where constructions are applied to transform the previous transient structures



until they reach an interpretable structure



from which the FCG-interpreter can derive an utterance

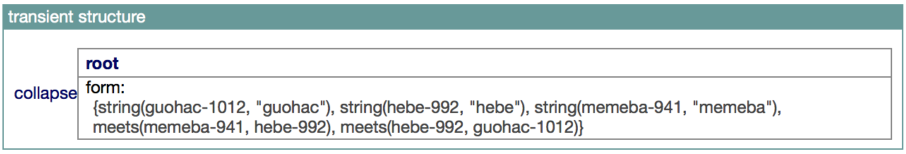
Utterance: memeba hebe guohac

Structure: "(memeba hebe guohac)"

Then the hearer starts the inverse process by comprehending (parsing) the resulting utterance

Comprehending "memeba hebe guohac"

and translating it to the first transient structure



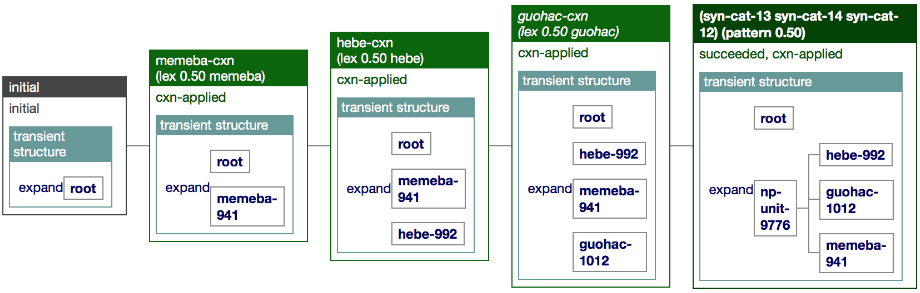
transient structure

collapse **root**

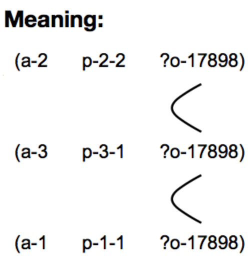
form:
{string(guohac-1012, "guohac"), string(hebe-992, "hebe"), string(memeba-941, "memeba"), meets(memeba-941, hebe-992), meets(hebe-992, guohac-1012)}

and processing until finding a solution

application process



from which a meaning can be derived



Meaning:

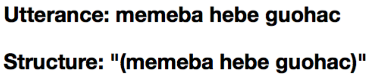
```
(a-2    p-2-2    ?o-17898)

(a-3    p-3-1    ?o-17898)

(a-1    p-1-1    ?o-17898)
```

Finally, in order to measure how well aligned the two agents are, and only for the sake of observation, the hearer formulates the meaning that he has derived

application process

initial — hebe-cxn (lex 0.50 hebe) — guohac-cxn (lex 0.50 guohac) — *memeba-cxn (lex 0.50 memeba)* — **(syn-cat-10 syn-cat-11 syn-cat-12) (pattern 0.50)**

and although a different construction applied, the utterance is the same as the one uttered by the hearer.



Utterance: memeba hebe guohac

Structure: "(memeba hebe guohac)"

Chapters 5 and 6 discuss results on the alignment between agents in detail.

To conclude, the table below makes a comparison between the main characteristics of the naming game and the syntax game.

	NG	SG
Initialization	$\emptyset$	Lexicon and ontology
Challenge	Lexical system	Language system exhibiting phrase structure
Topic	Object	Entity or event / Description
Topic Meaning	Single category	Meaning network
Utterance	One word	Multiple words organised in phrases

Table 2.1: The same way as naming games aim to study of the emergence of lexical systems, syntax games are designed to study the emergence of language systems exhibiting phrase structure.

2.4 Conclusions

In this chapter, I described the formal framework I use to approach the investigation on the emergence of phrases and recursion, including the ontology used for meanings and situations, the space of language systems exhibiting phrase structure and recursive phrase structure, and how lexicon is modelled and maps to predicates in the ontology.

Moreover, the chapter introduced central concepts and machinery involved in the models, such as syntax games and FCG, as well as described how they are integrated into the Babel2 Framework.

Part II

Results

Every chapter in this part corresponds to the work done in response to specific objectives of the thesis to defense the hypotheses proposed. Part of the work has been published in related original publications. The list of the corresponding publications is provided at the beginning of every chapter.

The first chapter uses analytical arguments to argue for functional adaptations driven by computational needs as one of the main functions for phrase structure and it introduces some of the building blocks needed for the language strategies studied in the other three chapters.

On the other side, the next three chapters mainly explore through computational experiments the behaviour of the computational models including such language strategies. The first chapter studies a model of the acquisition of phrase structure; in the second, the same model is extended and language strategies involving concrete selectionist criteria are proposed to show how phrase structure could be motivated by the need to reduce the computational cost in semantic interpretation, while always keeping communicative accuracy, and how first-order phrase structure can be achieved and acquired at the individual and collective level. Finally, the same machinery is tested for higher-order phrase structure in the last chapter, leading to recursive grammars.

Chapter 3

LANGUAGE STRATEGIES AND THEIR COGNITIVE EFFORT

Luc Steels and Emília Garcia Casademont.
Published [97]
Emília Garcia Casademont and Luc Steels.
Published [50]

The goal of this chapter is to introduce the study of the computational functions of phrase structure in the framework of the thesis. Ultimately I aim to frame phrase structure as a functional adaptation of some language systems which is driven by computational needs. Concretely, the first hypothesis of the thesis is that one of the functions of phrase structure is to reduce the cognitive effort required to process language.

In the first section of the chapter, I explain how I model this cognitive effort as the computational cost of semantic interpretation, and the motivation for the definition. In the second section, I measure the computational complexity of semantic interpretation for several language strategies progressively more similar to phrase structure grammars. Following the ontology introduced in the previous chapter, I first analyse first-order phrase structure grammars motivated to express multiple properties of an object, and after it, I analyse higher-order phrase structure grammars motivated to express relations and their arguments.

3.1 Determining Cognitive Effort

There is a widespread consensus that grammar has the function to express how word meanings are combined. Grammar helps knowing how the meaning fragments contributed by words are semantically connected to each other to give rise to a combined meaning. In particular, the syntactic structure that grammar computes gives cues on how meanings have to be combined. For example in the case of ambiguous words, grammar can derive different structures depending on the position of the word with respect to its adjacent words and this way helping to find the right structure, and therefore, the correct meaning.

In the present case, language strategies define how grammar expresses the way in which word meanings combine, i.e. in which ways the variables introduced by lexical constructions can be made equal to other variables.

3.1.1 Semantic and Syntactic Ambiguity

While it is true that grammar helps disambiguating ambiguous words, there are cases where it cannot disambiguate all the words. The corresponding utterances are said to have semantic ambiguity. The term semantic ambiguity usually refers to cases where the structure remains the same, but the individual words are interpreted differently. E.g. *he is in a terrible state*, where the word *state* has two interpretations. Either the state of the person, i.e. *how* he is, or a political state, *where* he is at this moment. In the models of the thesis, this kind of ambiguity cannot occur given that the mappings between word forms and meaning predicates are bijections, in other words, I don't consider polysemy.

On the other side, syntactic ambiguity occurs when an utterance has more than one interpretation due to ambiguous sentence structure. E.g. *I saw her duck*, where *her* can be interpreted as a possessive or as a pronoun, and consequently, *duck* can be interpreted as a noun or as a verb. Obviously, syntactic ambiguity generally leads to semantic ambiguity.

Finally, human languages may incorporate other kinds of ambiguities, such as phonological ambiguities, for example the one between *I scream* and *Ice Cream*, which are not studied in the thesis because I don't consider the phonological level of segmentation in the models.

Therefore, the only kind of ambiguity that is relevant for the phrase structure grammars modelled here is syntactic ambiguity. That is why the complexity measure that I use to compare language strategies is based on syntactic ambiguity. Concretely, per every utterance of a particular number of words, I define the computational complexity of a language strategy as the number of possible syntactic structures that the strategy supports.

In the computational experiments, I don't allow two or more objects with the exact same description in the situation, as it is explained in the description of the *Generation of situations* in sections 4.3.1, 5.3.1 and 6.4.1.

Consequently, the only source of ambiguity in the models is syntactic ambiguity, and hence each syntactic structure is identified either to a unique semantic interpretation in the situation or it has no semantic interpretation in case there doesn't exist a set of bindings consistent with the meaning found. Recall that semantic interpretation is represented as sets of bindings of variables in the utterance meaning to constants in the situation. Figure 3.1 illustrates on one side possible utterance meanings and on the other their corresponding syntactic structures, which in the case of the thesis are mapped one onto the other and each pair characterise a possible semantic interpretation.

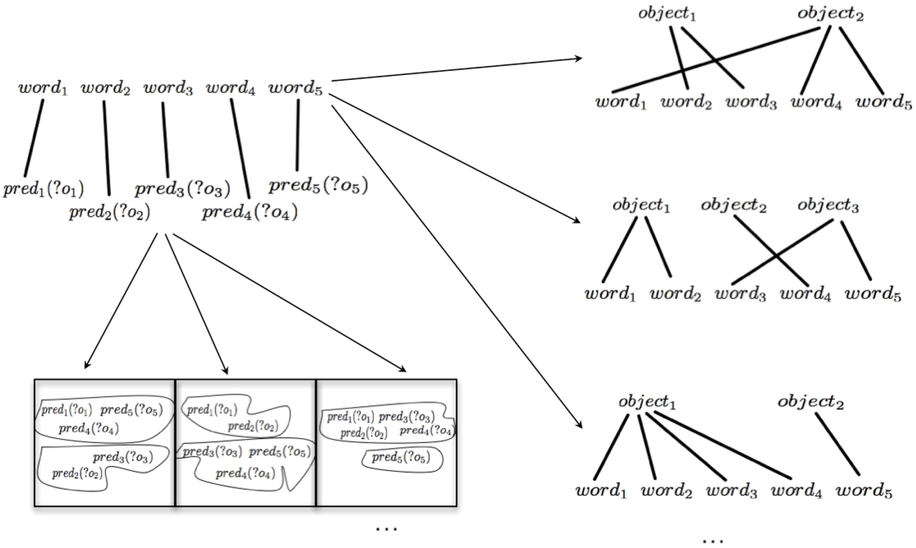


Figure 3.1: Illustration of the space of possible semantic interpretations associated to different syntactic structures.

Syntactic ambiguity is the complementary of phrase alignment, in the experiments I measure phrase alignment because I can do it explicitly and in the same conditions for all the interactions.

In the next section I compare several language strategies in terms of their computational complexity, but before I introduce few other measures of complexity for language processing related to the size of the grammar.

3.1.2 Size of the Grammar

In the computational models proposed in the thesis, language users are a community of artificial agents, and their power of computation is limited by the way in which language strategies and linguistic processing devices are represented. One of the most important measures of computational power is storage, so I define the size of an agent's grammar as the amount of information that the agent needs in order to store (and retrieves) information from his grammar. A phrase structure grammar is a combination of syntactic categories and phrasal constructions based on these syntactic categories. The basic measures for the size of the grammar are therefore the number of constructions and the number of categories. However, these two measures don't capture all the information encoded in the grammar. In figure 3.2 a phrase structure grammar is represented as a network, where edges go from lexical constructions (words) to lexical categories, and from lexical categories to phrasal constructions. And the number of constructions and categories gives no information on these relationships.

Moreover, in a population of agents there are as many grammars as agents, which makes categories difficult to compare across agents. In [49] I define several measures and methods to study the grammars of agents in a population.

Nevertheless, in the thesis I consider only the basic measures: the number of grammatical constructions, and the number of syntactic categories (lexical and phrasal categories).

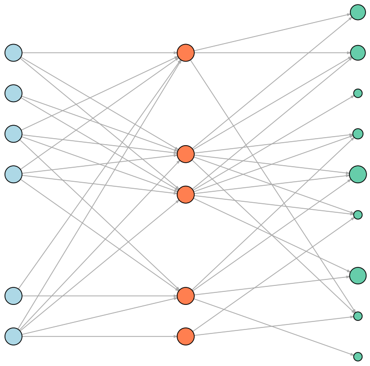


Figure 3.2: First-order phrase structure grammar that an agent evolved represented as a network where lexical constructions are represented as blue nodes, lexical categories as red nodes, and phrasal constructions as green nodes. Given that constructions are represented explicitly (and not as procedures, for instance), symbol equalities are required to account for the complete structure of the grammar, e.g. which phrasal constructions can apply with a lexical construction. In the network representation, this is represented using edges.

In the next section, the computational complexity defined above is measured for several language strategies progressively similar to phrase structure, and it is validated that the computational complexity of the strategies decreases as they get closer to phrase structure.

3.2 Computational Complexity of Language Strategies

I compare several language strategies progressively more similar to a language strategy exploiting phrase structure. As explained in the previous chapter, I consider two examples of ontologies, one consisting uniquely of predicates with one argument, and another one consisting of predicates with one to three arguments. The present section studies these two cases respectively, and for their corresponding syntax games.

3.2.1 First-Order Phrase Structure

In this section I study the first case, so I assume a lexicon consisting of a set of words $\{word_1, ..., word_m\}$, where each word $word_i$ introduces a predicate and a variable argument. The computational complexity of a language strategy equals the set of possible meanings whose corresponding syntactic structure is supported by the strategy (see figure 3.1), and syntactic structure combines words by making equal the variables of their predicates. When two variable predicates are made equal, I say that the variables (or the words) co-refer, because an equality of variables implies that in interpretation both predicate arguments are bound to the same constant, i.e. both words express properties of the same object.

I compare four different strategies: from the lexical strategy, where no grammar is considered, to the syntactic-based pattern strategy, where phrase structure is considered. The lexical strategy (L) doesn't impose any form constraint to the utterances and therefore the space of syntactic structures gets as big as it can be. The grouping strategy (G) imposes that only adjacent words can co-refer. The word-based pattern strategy (WBP) imposes that only sequences of words can co-refer. And finally, the syntactic-based pattern strategy (SBP) imposes that only sequences of syntactic categories can co-refer. Table 3.1 summarizes the language strategies studied.

	Two variables can co-refer
L	always
G	if the words are adjacent
WBP	if the words are in the same position as in a stored sequence of words
SBP	if the lexical categorization of the words are in the same position as in a stored sequence of lexical categories

Table 3.1: Language strategies for first-order phrase structure classified according to how they represent the co-referentiality of properties.

Lexical Strategy

In this strategy, only lexical constructions are used, and therefore the relative order between words is random. See figure 3.3 below for an example of its usage.

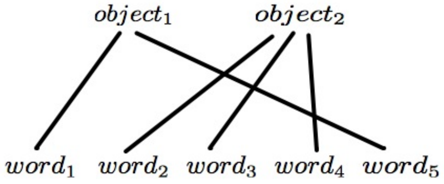


Figure 3.3: Lexical Strategy. The utterance doesn't encode which words refer to the same objects.

In this strategy, a speaker looks up the set of words that expresses all the object descriptions that he wants to convey (lexical constructions) and utters these words in any order to the hearer (no grammar). So the computational complexity of the semantic interpretation for the hearer of an utterance of length n equals the number of partitions of the set of words. A partition of a set is a set of pairwise disjoint subsets whose union is the original set, and the total number of partition for a set of n elements is B_n, the Bell Number for n [7].

$$B_n = \sum_{i=0}^{n-1} \binom{n-1}{i} B_i, \ B_0 = B_1 = 1 \tag{3.1}$$

So, in this case, semantic interpretation scales double exponentially with respect to the number of words.

However, if the ontology definition has attribute incompatibilities, this value can reduce considerably. For example this is the case when it is taken into account that words that are values of the same attribute and are different cannot refer to the same object, e.g. *red* and *blue*. The attribute compatibility can be constrained by compatibility sets limiting the attributes that can be combined in the same object. For example, while *colour* could go with any other attribute, *material* could be incompatible with *person*.

Figure 3.4 compares the complexity of the lexical and grouping strategies and also plots the complexity for these cases.

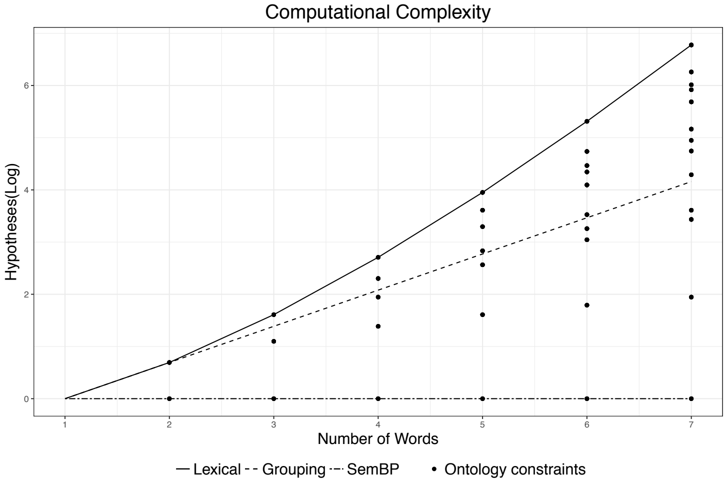


Figure 3.4: The complexity of the lexical strategy (with and without ontology constraints), the grouping strategy and a fully-aligned SemBP system are plotted in logarithmic scale against the number of words.

Grouping Strategy

In this strategy the speaker groups together the words referring to the same object, and in this case the complexity of semantic interpretation is of one degree less than in the lexical strategy. Figure 3.5 provides an example of it.

$$2^{n-1} \tag{3.2}$$

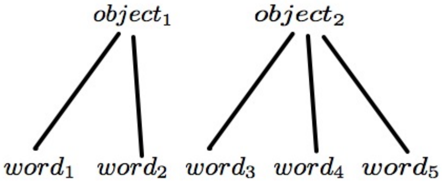


Figure 3.5: Grouping strategy. Only adjacent words can refer to the same object.

Word-Based Pattern Strategy

In the Word-Based Pattern Strategy words referring to the same object are also put together but with a particular order (see figure 3.6). Therefore, differently than in the previous cases, one particular object is always expressed in the same way, so there is linguistic consensus on phrases representing objects, although not always across utterances due to the free order between two objects.

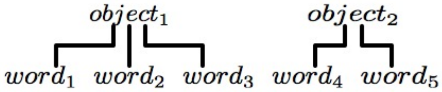


Figure 3.6: Word-based Pattern Strategy. Object descriptions are uttered in fixed sequences of words.

The computational complexity of semantic interpretation of the word-based pattern strategy is the same as the one of the grouping strategy restricted by the ontology, given that only adjacent phrases whose corresponding meanings can be identified to an object are hypothesized as solutions. This is a slightly better result than the pure grouping strategy.

Syntactic-Based Pattern Strategy

The syntactic-based pattern strategy leads to language systems based on the use of patterns of categories (see figure 3.7). Every category is a set of words and all categories are different from each other. Hence, this is the strategies for first-order phrase structure grammars that I consider.

In this case linguistic consensus is not necessarily assured for all the objects; and while the computational complexity of semantic interpretation is bounded by the one of the word-based pattern strategy and the grouping strategy, the grammar inventory size may differ a lot between language systems.

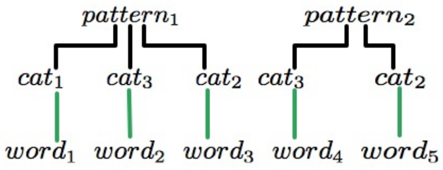


Figure 3.7: Syntactic-Based Pattern Strategy. Object descriptions are uttered according to sequences of lexical categories, and therefore, previous correspondences from words and lexical categories are required.

Although the space of language systems that this strategy opens is big, I test in the coming chapter how a population of agents endowed with certain learning operators is capable to self-organise syntactic-based pattern systems, that improve the complexity of the grouping and word-based pattern strategies.

Notice that in the extreme cases, one could conceive on one side a

Language Strategies	Computational Complexity
L	B_n
G	2^n
WBP	$K \cdot 2^n$ for $0 < K < 1$
SBP	$\leq 2^n$
(fully-aligned) SemBP	1

Table 3.2: Comparison of the computational complexity of language strategies to express co-referentiality.

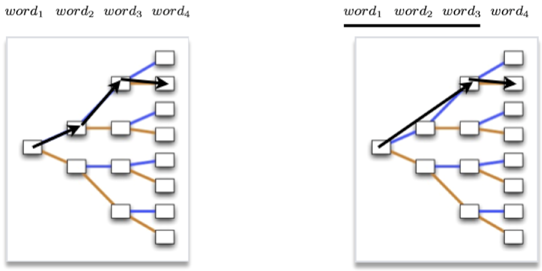
syntactic-based pattern system with only one category, which would be equal to the grouping strategy; and on the other side, one system with as many categories as words, which would be equal to a word-based pattern system. However, the behaviour of intermediate systems in terms of computational cost (i.e. processing steps) is not so easy to predict, nor it is its relation to the size of the grammar, although as it can be seen from the extreme cases at some point there is a trade-off between the two.

A special case of SBP systems are the SemBP systems, i.e. semantic-based pattern systems, whose syntactic categories are equal to semantic types. Morever, a system is fully aligned when only a single phrase can be produced to express any connected meaning, and given that the semantic types defined in the ontologies here used are a partition of the set of object properties, it is assured that a fully-aligned SemBP exists. Figure 3.4 and table 3.2 also include this case.

Chapters 4, 5 and 6 study SBP systems derived from agent-based collective algorithms based on the language games and grammars discussed in chapter 2.

Processing Steps and Search Space

The space of linguistic structures is explored by the application of grammatical constructions. When looking at language processing it is found that the word-based pattern strategy reduces the number of steps required to finalise the process of finding a solution with respect to the grouping strategy. Recall that these steps are the base for the definition of computational cost that is used in chapters 4, 5 and 6.



Grouping Strategy

Word-Based Pattern Strategy

Figure 3.8: While the grouping strategy (G) needs a construction application (processing step) for every pair of words, the word-based pattern strategy (WBP) requires the same number of steps than the number of objects in the hypothesis.

In table 3.3 I include the average number of processing steps for optimal computations in terms of the search space for each of the language strategies, and it is shown that in WBP it decreases with respect to L and G. See figure 3.8 for an illustration.

Language Strategies	Computational Complexity	(Average) Processing Steps
L	B_n	$\frac{\sum_{i=1}^{n} \left\{ {n \atop i} \right\}(n-i)}{B_n}$
G	2^n	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1}(n-i)}{2^n}$
WBP	$K \cdot 2^n$ for $0 < K < 1$	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1} i}{2^n}$
SBP	$\leq 2^n$	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1} i}{2^n}$
(fully-aligned) SemBP	1	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1} i}{2^n}$

Table 3.3: Comparison of the computational complexity and the average number of steps for the language strategies studied.

Grammar Size

In the case of the lexical strategy only one construction is used, namely the one making equal the variables introduced by two different predicates (words). So, the number of constructions can be computed as 1, and so does the number of lexical categories.

In the case of the grouping strategy, the grammar size can be computed again as 1 for both constructions and categories, because that is the minimum number required.

On the other side, in the case of WBP, the grammar size equals the number of objects, which scales exponentially with respect to the number of attributes. For an ontology with a attributes assuming v values for all of them, the number of objects is $(v+1)^a - 1$.

Finally, language systems derived from the SBP strategy can reduce the number of constructions with respect to WBP due to the use of syntactic categorisations instead of the words themselves.

Table 3.4 below summarizes all the results discussed in this section.

Language Strategies	Computational Complexity	(Average) Processing Steps	Grammar Size
L	B_n	$\frac{\sum_{i=1}^{n} \genfrac{\{}{\}}{0pt}{}{n}{i} (n-i)}{B_n}$	1
G	2^n	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1} (n-i)}{2^n}$	1
WBP	$K \cdot 2^n$ for $0 < K < 1$	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1} i}{2^n}$	$(v+1)^a - 1$
SBP	$\leq 2^n$	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1} i}{2^n}$	$< (v+1)^a - 1$
(fully-aligned) SemBP	1	$\frac{\sum_{i=1}^{n} \binom{n-1}{i-1} i}{2^n}$	$2^a - 1$

Table 3.4: Comparison of the language strategies studied for the three measures: computational complexity, processing steps and grammar size (number of constructions).

3.2.2 Higher-Order Phrase Structure

In this section, I make a similar comparison as before, but for several language strategies progressively closer to a language strategy exploiting higher-order phrase structure. In this case, I explore meanings including relations, and I study both a description and a discrimination syntax game with a lexicon consisting of a set of words $\{w_1, ..., w_m\}$. Words are divided into attribute words and relation words, introducing each of them a predicate. For the attribute words the predicate is introduced together with one variable argument, and for the relation words, together with at least two variable arguments, e.g. $w = p_a(?v)$ or $w = p_r(?v_1, ?v_2, ?v_3)$.

Actually, in order to simplify the approach, the meaning of the utterances considered in the analysis is not exactly derived from the ontology, and instead it is formed by several attribute words describing a set of objects and one relation word whose arguments are all the objects in the utterance, see figure 3.9 for an example. I consider an utterance with n attribute words and a relation word that introduces m variable arguments, and I impose $n \geq m \geq 1$. Although in the ontology used for the experiments I use only relation words introducing 2 and 3 variables, here I make a generalization for m in order to approach the analysis of the complexity of the strategies studied.

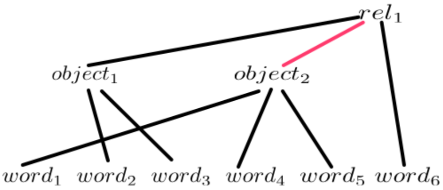


Figure 3.9: The meaning utterances used to analyse the strategies consists of several attribute words describing a set of objects and one relation word (*word*$_6$) whose arguments are all the objects in the utterance.

Concerning the dichotomy description/discrimination (language) game, in the case of first-order phrase structure the game could not be a discrimination game because the interpretation task would be trivial, meaning that all words would refer to the same object. That's why only a description (or multi-referential game) was considered.

Moreover, in general in a description game the process of pointing looks more unreal that in a discrimination game. However, for higher-order phrase structure I also include the setup of a description game in the analysis because it is a kind of game that has been used in most previous similar studies [26, 58], even though in the model in chapter 6 I consider only a discrimination game.

Lexical Strategy

In this case the computational complexity of semantic interpretation for the kind of utterances considered is, in the case of a description game:

$$m!\left\{ n \atop m \right\} \tag{3.3}$$

where $\left\{ n \atop m \right\}$ is the stirling number of the second kind for n and m. This number computes the number of partitions of a set of n elements into m non-trivial subsets.

And in the case of a discrimination game:

$$m!\left\{ n \atop m \right\}(m+1) \tag{3.4}$$

because a choice of a reference has to be made.

In this case the size of the grammar can be computed similarly than in the case of first-order phrase structure including a few more constructions, due to the assignment of relational arguments. So the size can be bound by $m+1$ for both constructions and categories.

Grouping Strategy

The grouping strategy groups together the words referring to the same object and a relation word with its object arguments, being the latter useless for the current case because there is only one relation so all the words that would be grouped are all the words in the utterance, which of course are already grouped.

I define n_{left} as the number of attribute words in the left of the word relation and n_{right} as the number of attribute words in the right of the word relation. The computational complexity in this case is:

$$\sum_{\substack{(i,j)\\ i+j=m\\ i\leq n_{left}\\ j\leq n_{right}}} m!\binom{n_{left}-1}{i-1}\binom{n_{right}-1}{j-1} \tag{3.5}$$

for the description game, and:

$$\sum_{\substack{(i,j)\\ i+j=m\\ i\leq n_{left}\\ j\leq n_{right}}} m!\binom{n_{left}-1}{i-1}\binom{n_{right}-1}{j-1}(m+1) \tag{3.6}$$

for the discrimination game, where in both cases the effect of the grouping strategy on reducing the space with respect to the lexical strategy is again more than exponential.

The grammar size is again similarly than in the previous case.

Word-Based Pattern Strategy

The idea of the word-based pattern strategy as grouping words of the same relation or arguments of that relation cannot be used here unless a sequence of words is stored for every single utterance. So, the word-based pattern strategy doesn't support hierarchical semantic structures.

Syntactic-Based Pattern Strategy

Syntactic-based pattern systems are linguistic systems based on the use of patterns of categories (see figure 3.10). As explained in chapter 2, here there are two types of categories:

- categories defined as sets of attribute words or as sets of relation words that I call *lexical categories*. All *lexical categories* are different from each other.

- categories added in the unit created by a pattern. These categories are called *phrasal categories*, and they can be shared across patterns.

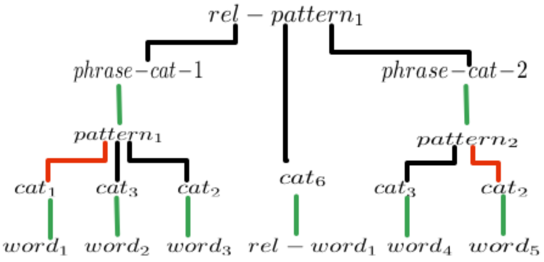


Figure 3.10: Schematic representation of how phrasal categorisation acts.

The complexity in this case is much less than previously:

$$\binom{n_{left}-1}{i-1}\binom{n_{right}-1}{j-1} \tag{3.7}$$

for the description game, and

$$\binom{n_{left}-1}{i-1}\binom{n_{right}-1}{j-1}(m+1) \tag{3.8}$$

for the discrimination.

Let v_r be the number of values per relation type, and v_p, the number of values per property type.

Language Strategies	Computational Complexity	(Average) Processing Steps	Grammar Size[1]
L	B_n	$\frac{\sum_{i=1}^{n} \left\{ {n \atop i} \right\} (n-i)}{2^n}$	$m+1$
G	$\sum_{\substack{(i,j)\\ i+j=m\\ i\leq n_{left}\\ j\leq n_{right}}} m! \binom{n_{left}-1}{i-1} \binom{n_{right}-1}{j-1}$	$\frac{\sum_{i=1}^{n} \left\{ {n \atop i} \right\} (n-i)}{2^n}$	$m+1$
WBP	this strategy doesn't support higher-order phrase structure		
SBP	$\binom{n_{left}-1}{i-1} \binom{n_{right}-1}{j-1}$	$\frac{n(n+1)/2}{2^n}$	$\leq v_r \cdot v_p^m$
(fully-aligned) SemBP	$\binom{n_{left}-1}{i-1} \binom{n_{right}-1}{j-1}$	$\frac{n(n+1)/2}{2^n}$	$r \cdot a^m$

Table 3.5: The complexity measures discussed are given by four language strategies considering an utterance with n words, including one binary relation.

[1] In this case, grammar size computes the additional grammatical constructions to process relation words, so that in order to get the total number of constructions they should be add up to the ones computed for first-order phrases.

3.3 Discussion

In this chapter, I compared language strategies leading to grammars each time progressively closer to grammars exhibiting phrase structure, and I showed how they progressively reduced their computational complexities. This gives evidence to hypothesize computational functions for phrase structure, and concretely a function to reduce the use of memory and computational power in language processing. This grounds the idea to use the reduction of the computational cost of semantic interpretation as a selective pressure. In chapters 5 and 6, this is analysed studying resulting grammars from computational experiments where this computational pressure is considered.

Moreover, the difficulties of choosing a system exhibiting phrase structure (i.e. syntactic-based pattern strategy) which minimizes its computational resources and computational cost in language processing have been made explicit. This is the main problem that populations of artificial agents are capable to solve in the models proposed in the thesis.

Therefore, phrase structure helps agents avoid combinatorial explosions and ambiguity, both for semantic interpretation, where combinatorial search and ambiguity unavoidably arise when multiple words are used without signaling how these words are semantically related, and for parsing, because words or patterns tend to have multiple possible functions, generating combinatorial search.

Chapter 4

ACQUISITION OF PHRASE STRUCTURE

Luc Steels and Emília Garcia Casademont.
Published [98]
Emília Garcia Casademont and Luc Steels.
Published [51]

The acquisition of phrase structure is a prerequisite for its maintenance and transmission across generations of agents. In this chapter I start the study of language strategies that allow agents to build and maintain a shared language system exhibiting phrase structure, and I do that by describing and testing a model for the acquisition of phrase structure.

The model is a syntax game model whose communicative interactions occur between a tutor agent, which is endowed with a fully formed syntactic-based pattern system, and a learner agent that gets the same lexical constructions as the tutor, but no phrasal constructions yet. The tutor agent always gets the role of the speaker and the learner, the role of the hearer, so that he gets exposed to a language with a systematic use of word order. Moreover, the learner applies as well a syntactic-based pattern strategy to acquire a phrase structure grammar.

Remember that while word-based pattern systems (WBP) are sets of grammatical constructions where specific word orders are directly mapped to sets of bindings between predicate variables, in syntactic-based pattern systems (SBP) these mappings require lexical constructions introducing lexical categories, because word orders are assigned through these lexical categories instead of directly through words.

I propose a syntax game based on an ontology consisting of unary predicates and co-referentiality of arguments for object properties, as well as specific adoption operators that are tested using simulations.

Detailed description of the model can be found in the interactive web demonstration www.biologiaevolutiva.org/lsteels/tesi-emilia/acq

In the first section I hypothesize the adoption operators; in the second, I describe and study the performance of a baseline experiment; in the third, I explain the design of the main acquisition experiment and present its results, and finally, the fourth section discusses the implications of the main results of the chapter.

4.1 Learning Operators for the Acquisition of Phrase Structure

The model proposed consists of a description syntax game which uses an ontology consisting of properties, i.e. predications with a single variable. Remember that this ontology is made of unary predicates as *(color blue ?o-1)*, where variables are made equal to each other when they co-refer to the same object in a situation model. *color* is the semantic type (so-called attribute), and *blue* is the property (so-called attribute value).

Given that the aim of the experiments in this chapter is to study the acquisition of phrase structure, the population of agents in the model is made of two agents: a tutor and a learner. And the roles of speaker and hearer are always given to the tutor and the learner, respectively. While the tutor is initialized with a full grammar, the learner is initialized only with a lexicon. Such lexicon maps every first-order predicate of meaning into a word, but has no information on how predicate variables could be bound to each other, so it is equivalent to a lexical system as the ones given by the lexical strategy in chapter 3. The role of grammatical constructions is precisely to relate the relative word orders in the utterances with the linking between variables of the predicates in the processed meaning. The system then turns to a pattern system when a consistent set of grammatical constructions is added. A pattern system is one as the ones given by pattern strategies (word-based or syntactic-based) in chapter 3.

If agents don't share enough grammar, greater is the cost of the hearer (i.e. in this case the learner) to process an utterance. So the challenge of the learner is to acquire enough knowledge in the form of grammatical constructions so that both agents can reach full communicative success, minimizing their computational resources.

In order to do that, the learner is endowed with a set of learning operators. Learning operators are models of the learning strategies, which ultimately are models of cognitive mechanisms.

The question addressed in this chapter is how constructions empowering a language system with phrase structure can be acquired, which is a first step towards the understanding of phrase structure, given that acquisition is in the core of the mechanisms responsible for cross-generational transmission of cultural or linguistic traits.

The remaining of this section describes the learning operators that have been tested, and some of their effects to language processing. The implementation of the operators is based on the learning component of the Babel2 platform [12] and examples of the application of these learning operators can be also found in the site www.biologiaevolutiva.org/lsteels/tesi-emilia/acq

4.1.1 Contextual Inference Operator

The contextual inference operator (CI) allows the creation of phrasal constructions as agents make hypotheses of the language of their interlocutors. Hence, this is the operator which potentially generates variation in the grammars of the agents. However, in this chapter the focus is on the acquisition and therefore only the learner uses the learning operators. On the other side, the tutor agent is endowed with a consistent grammatical system, meaning that every object description is uttered in a unique way. This means that the learner is exposed always to the same descriptions and that the application of the learning operator won't lead to competitor constructions. Moreover, this implies that agents are expected to reach linguistic consensus, which is not obvious in the case of a population of more than two agents with no grammar yet (see chapter 5).

The operator applies in two steps of the game: in language processing (flexible grammar) and at the end of the interaction when they have been successfully applied (learning). In language processing it helps to find linguistic structures beyond those computable by the grammar of the agent, and at the end of the interaction, it stores in the agent's grammar the new construction that encodes the transformation the operator has introduced, in case this one helped to find the solution.

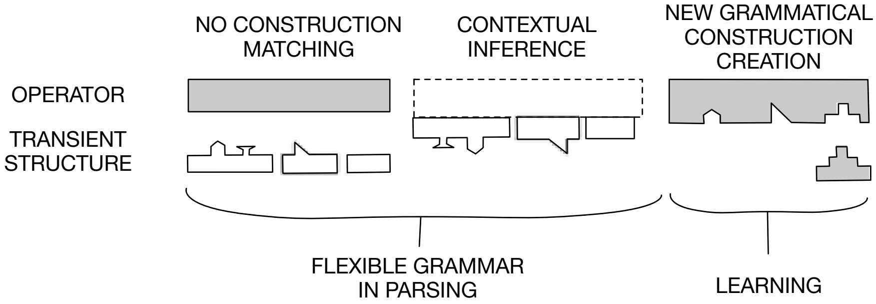


Figure 4.1: The contextual inference operator applies in parsing when no phrasal constructions can apply. Such operator builds a phrasal unit containing the corresponding lexical units in the order they are given, and when it lead to the solution, a new pattern construction mimicking the operator is created. The category slots of the new construction are assigned according to the lexical units: if a unit had one lexical category, this one is taken; if it had more than one, a random one from them is taken; and if it had no lexical categories yet, a new one is created and the corresponding lexical construction updated, as it happens in the example with the rightmost lexical unit.

So, the application of this learning operator ultimately adds a new grammatical constructions to the grammatical inventory of constructions of the learner. The learner applies a syntactic-based pattern strategy, so it can be that new lexical categories have to be created. The operator reuses as much information as possible from the grammar of the agent, so if the lexical units in the structure had already some potential syntactic categories, the operator selects one of them at random; on the other side, if the lexical units had no potential syntactic categories, the operator creates a new category and also makes sure that the corresponding lexical constructions will add it next time, see figure 4.1 for an illustration of its application.

In terms of cognitive abilities, this operator can be justified by two reasons: one, the fact that humans use semantics independently of language to complete an hypothesis about the meaning of an utterance, when they don't know that language; and two, the adjacency bias, i.e. the bias towards favouring the combination of local elements in language utterances, which is supported by many psychological [42], analytical [38] and empirical [45] studies.

4.1.2 Syntactic Coercion Operator

The syntactic coercion operator (SC) allows the hearer to look for a construction which is only partially matching before applying the contextual inference operator. This is done by the syntactic coercion operator. The syntactic coercion operator applies in language processing (flexible grammar application) and at the end of the interaction (learning), similarly than the contextual inference operator, when the application of the operator leads to a successful solution.

This operator is inspired by the phenomenon of coercion, a phenomenon that it occurs for example when a word is used as an unexpected lexical category e.g. *she googled me on the Internet*, where a noun (*google*) is used as verb (*googled*). I model coercion as the expansion of a lexical construction by expanding the list of syntactic categories that the introduced lexical unit can use.

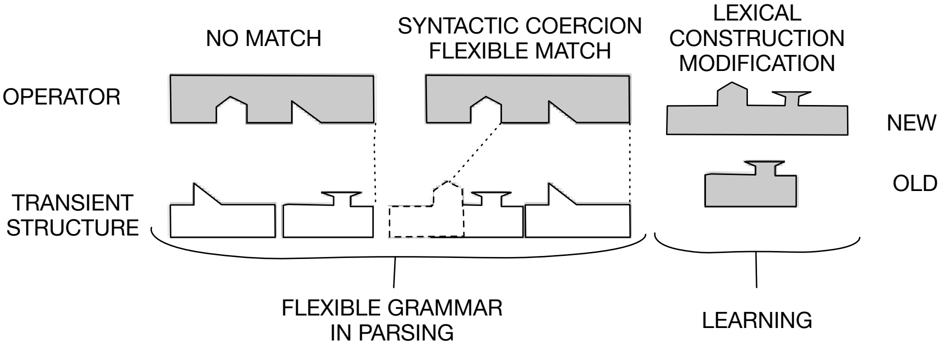


Figure 4.2: Syntactic coercion is the procedure that allows to do a flexible matching on a lexical unit by assuming it has a lexical category that it is missing, and at the end of the interaction it makes this modification effective by modifying the original lexical construction that created the lexical unit. Syntactic coercion grammar updating always consists of extending the lexical categories of the corresponding lexical construction.

In language processing, syntactic coercion applies after the fully matching of grammatical constructions is tried (see section 4.1.4). A construction is found that is partially compatible when one (or more) words do not have the appropriate lexical category (as in the example of "googled" where a noun occurs in a context where a verb is expected). The operator then omits that constraint on the lexical unit in the transient structure and the construction applies normally. The resulting structure is again transformed by the grammar and operators until a solution is found. At the end of the interaction, the corresponding lexical construction is modified to account for the lexical category that was missing, in case syntactic coercion helped to find the solution.

Coercion is a way to minimize the number of patterns that are used, and therefore it subscribes to one of the main tenets of the thesis, which is favouring the minimization of computational resources. Moreover, it has cognitive justifications from outside the context of language, and even from non-human behaviour, e.g. experiments shown already in 1956 that chimpanzees could *coerce* objects to have functions that they normally do not have [64]. For example, to view a shoe as a hammer so that it can play the role of the instrument in a hitting action.

Concerning the representation of grammars, coercion flexible matching is therefore modelled as a flexible matching at the level of lexical categories in lexical units and not at the level of lexical categories in phrasal construction slots. See figure 4.2 for an illustration if its application. The reason for this design choice is that the focus of the thesis is on the phrasal constructions, which are the ones carrying out the competition that is under study by underlying the usage of potentially variate constituent orders.

4.1.3 Ordering Variation Operator

The ordering variation operator again applies in language processing (flexible grammar) and at the end of the interaction (learning) after being successfully applied. In language processing it is very similar than syntactic coercion. In this case it accepts a construction to apply by allowing different word orderings, instead of by accepting missing lexical categories. On the other side, at the end of the interaction when it comes to updating the agent's grammar, it is similar to the contextual inference operator, given that it creates a new phrasal construction accounting for the new order and the same syntactic categories of the original pattern.

Figure 4.3 illustrates the application of this operator.

Summing up, the three adoption operators imitate the application of a phrasal constructions but are less restrictive in choosing the units to be used as constituents. SC and OV apply a concrete construction with softer constraints; SC is less restrictive with syntactic categorizations and it may choose a unit (or possibly more) that doesn't have the expected syntactic category; and OV is less restrictive in the ordering of constituents (units) and it may choose a set of units not satisfying the corresponding orderings. On the other side, CI applies resulting in a phrasal unit on top of other units (constituents) but it has no more restrictions than the general ones for a phrase structure, i.e. in parsing, the constituents have to be adjacent in the utterance; and in production, the referents of the constituents have

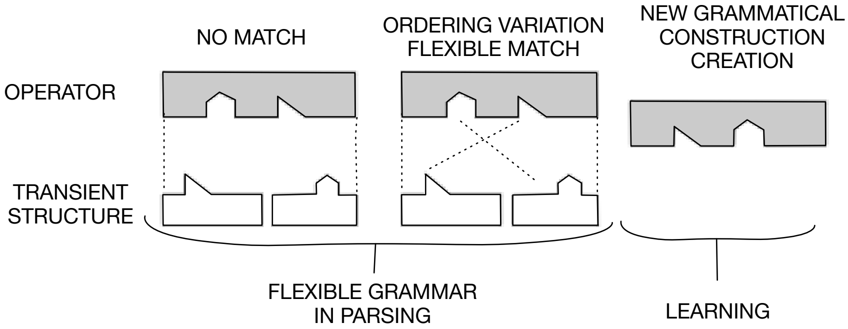


Figure 4.3: Ordering variation allows flexible matching as it comes to the relative order between constituents. Therefore, at the end of the interaction, in consolidation, it stores in the agent's grammar a phrasal constructions with the same set of syntactic category than an existing one but ordered differently.

to be the same. Table 4.1 compares the three operators according to their requirement in flexible grammar and implications in learning.

	SC	OV	CI
Grammar Application: Flexibility in	syntactic categorizations	phrasal ordering constraints	both
Learning: Grammar Update	a syntactic category is added to a lexical construction	a new phrasal construction is built	a new phrasal construction is built

Table 4.1: The three adoption operators proposed have similar methods to enable flexible grammar. And moreover, if their application leads to the solution structure, new grammar is learned to account for the result of the application of the operator.

4.1.4 Bias in Language Processing

The main effect of the three operators in language processing is that when the learner tries to parse an utterance he prioritizes the application of grammar over learning operators, and the syntactic coercion and ordering variation operators over the contextual inference operator.

Moreover, in order to guarantee the proper application of the operators, those phrasal constructions that have a larger number of constituents have to be tried before than those having a smaller number of constituents, which applies also for constructions trying to be used in syntactic coercion. Furthermore, this has to coexist with the a mechanism of *online interpretation*. The online interpretation is a systematic check of the consistency of bindings introduced by each construction or operator with respect to the current situation, where meaning hypotheses including inconsistent bindings, i.e. including bindings which have no interpretation in the situation, are not pursued.

Besides, the hearer stops searching for a syntactic structure when he has no more grammar or operators to apply. Without online interpretation he would stop after computing a first random syntactic structure and the computational cost would be minimal. Only when online interpretation is considered, the space of computable solutions is reduced and there is variability in the computational cost across interactions.

4.2 Baseline Experiment

This section provides details of the model and computational experiments, and the description and results of the baseline experiment against which the model performance is compared in the coming section.

Baseline experiments are used to compare the behaviour of a model against a simpler model that has less functionalities. In this case, the main interest lies on the effect of the learning operators.

4.2.1 Experiment: No Adoption Operators

In order to test the learning operators, which are the adoption operators described in the previous section, I compare them against a baseline experiment where such operators are inhibited so that their direct effect can be identified.

In this case, at the beginning of the experiment the language system of the learner is similar than a grouping system as the ones discussed in chapter 3 but including also the ability of inhibiting inconsistent hypothesis (online interpretation) and prioritizing the hypotheses composed by longer phrases (groups) first.

The baseline experiment involves two agents, one tutor endowed with a SBP and one learner endowed only with a lexicon, but no grammar nor learning operators.

In terms of the implementation, this means that the first experiment is run with the same initialization as the main acquisition experiment, but without learning operators.

Ontology Size

The ontology consists only of predicates that have one argument and is given by 3 semantic types and 2 properties each of them.

Semantic Types *A*	Properties: *P*
a-1	p-1-1 p-1-2
a-2	p-2-1 p-2-2
a-3	p-3-1 p-3-2
a-4	p-4-1 p-4-2
a-5	p-5-1 p-5-2

There are no incompatibilities between properties and the repetition of attributes in the same object description are not allowed.

Utterance length

words = $\{2, 3, 4, 5\}$

Population Size

2 agents.

Situation size

The situation size equals the number of attributes in the situation. So, it is limited by the ontology size given that two objects with exactly the same attribute values are not allowed. I set up the situation size as two times the utterance length.

Generation of the situation

The set of attributes in the situation is randomly generated and conditioned by the utterance length and the situation size (here set to 2·*utterance length*). The situation is generated as follows:

- When utterance length is set to 2, the situation generated consists of a set of two objects of two attributes each (2 2).
- When utterance length is set to 3, the situation generated consists of two to four objects whose attributes sum up to 6 after two random selections from (2 1) and (3); an object of two attributes and an object of one attribute, or an object of three attributes.
- When utterance length is set to 4, the situation generated consists of two to four objects whose attributes sum up to 8 after two random selections from (2 2), (3 1) and (4).
- When utterance length is set to 5, the situation generated consists of two to four objects whose attributes sum up to 10 after two random selections from (2 3), (4 1) and (5).

Moreover, in any of the cases it is allowed to generate situations with two objects being described exactly by the same attributes nor with an object having the same attribute more than once in its description.

As I discuss in the next results section, the behaviour of the model can be fully predicted analytically.

4.2.2 Results

The results of the experiment measure the *communicative success*, *the alignment* and *the computational cost*. The *number of phrasal constructions* here is not a relevant measure given that no learning operators apply and hence agent grammars don't change, and *phrase alignment* influences *alignment* in a clear and stable way so only *alignment* is plotted.

In this experiment, agent grammars don't change because no learning operators apply. Results in figure 4.4 are from 5 independent runs and the average is plotted using a sliding window of 100 interactions with a gray ribbon for the sample standard deviation. The number of words in the utterance increases every 3000 interactions, starting from 2 and ending with 5.

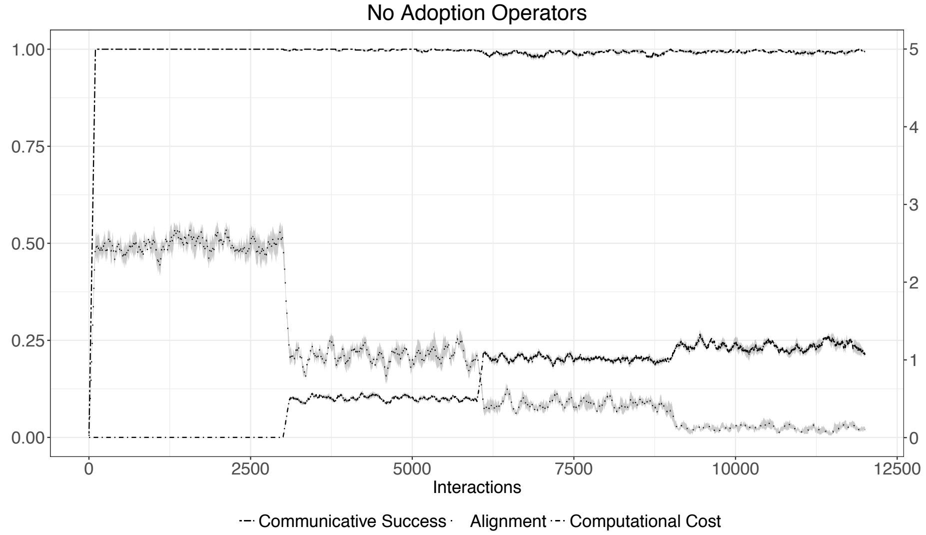


Figure 4.4: Agents reach full CS in most of the interactions. On the other side the alignment across agents is never reached and it decreases as the number of words in the utterance increases. The computational cost is measured on y-axis in the right side and it increases as the number of words increases.

The remaining of this section discusses in detail the reasons behind the observed results.

Communicative Success (CS)

Agents have full communicative success in most of the interactions, but not in all, as it is shown by the variation of its value, which isn't always 1. This is because the hearer doesn't consider only the hypotheses which include equalities between predicate variables that are not consistent with the situation thanks to the online interpretation. Remember that online interpretation is crucial to get variability, though. Otherwise, the hearer would consider only the first solution he computes.

The failure in communicative success comes from two cases:

1. Cases where the situation includes three objects whose properties can be represented as (A) (B) (A B) (or (A B) (C) (A B C), etc.) and a configuration where (A) and (B) are part of the selected partial description to convey, and are found in an unambiguous order in the produced utterance.

   For example, a situation including the predications *(colour red o-1) (physical-object table o-1) (colour red o-2) (physical-object table o-3)* where the last two are selected as part of the partial description and are adjacent in the utterance.

2. Cases where another linguistic structure is compatible with the current situation and equally likely to occur than the correct one due to random choices in the search tree.

   What it occurs is that the hearer doesn't compute the right boundaries between object descriptions in the utterance, e.g.: (A B C D E) in the utterance, where the partial description that the utterance describes includes (A B) and (C D E), and the structure computed by the hearer is (A B C) and (D E), where all the four objects are in the situation (otherwise this hypothesis would be cancelled by the online interpretation).

Alignment

In terms of alignment and phrase alignment, the results are the same as they would be if both agents were initialized as the learner, given that the tutor always produces the same object description, and the learner always interprets random object descriptions. That is why the results for alignment can be computed analytically:

- for the first 3000 interactions, where utterances have 2 words, in half of the cases the two words refer to the same object in and the other half, they refer to two different objects. However, for both cases, the probability to get fully aligned equals to $\frac{1}{2}$ given that the utterances refers only to these one or two objects. $A = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$

- for the second 3000 interactions, where utterances have 3 words, the utterance could refer to one object (the three words refer to the same one), to two objects (two words refer to one and the third word refers to another one) or to three objects (each word refers to a different object). However, the last case is not considered as possible in the experimental setup given that it doesn't motivate the usage of first-order phrase structure to disambiguate the expressed meaning. Therefore, the alignment is given by:
  $A = \frac{1}{2} \cdot \frac{1}{2^2} + \frac{1}{2} \cdot \frac{1}{6} = \frac{1}{2} \cdot \frac{5}{12} = \frac{5}{24} \approx 0.21$
- following the same reasoning, for the third interval of 3000 interactions, where utterances have 4 words, the alignment is given by:
  $A = \frac{1}{3}\frac{1}{2^3} + \frac{1}{3}\frac{1}{12} + \frac{1}{3}\frac{1}{24} = \frac{1}{3}\frac{1}{4} \approx 0.08$
- and finally, for the last interval of 3000, where utterances have 5 words, the alignment is given by: $A = \frac{1}{3}\frac{1}{2}\frac{1}{2}\frac{1}{6} + \frac{1}{3}\frac{1}{2}\frac{1}{24} + \frac{1}{3}\frac{1}{120} = \frac{1}{3}\frac{17}{240} \approx 0.024$

So the simulated data satisfy the predictions.

Computational Cost

Recall that the computational cost is the ratio between the number of nodes in the search space of grammar application and the number of phrases in the utterance, minus one. For example, if the targeted utterance structure is ($word_1$ $word_2$) ($word_3$ $word_4$ $word_5$) and grammar application explores 4 nodes (because it considers incorrect hypotheses) the computational cost is $\frac{4}{2} - 1 = 1$ whereas if it explores precisely the two nodes where the two phrases are built, the computational cost is $\frac{2}{2} - 1 = 0$.

In this experiment the computational cost increases as the number of words in the utterance increases.

- In the case of sentences with 2 words the computational cost is only 0. It could be different than 0 for two cases. The first one happens when the same situations where communicative success fails occurs, i.e. situations with three objects having the description of attributes (A B), (A) and (B), where the last two are the partial description topic of the interaction. In this case the computational cost would be $\frac{1}{2} - 1 = -0.5$. And the second occurs when the partial description topic consists of two objects (A) (B) and (A B) is not in the situation, however the hypothesis consisting of a phrase expressing (A B) is considered, so the computational cost would be: $\frac{3}{2} - 1 = 0.5$ in $\frac{1}{2}$.

- 3 words: besides cases where the computational cost is 0, it also occurs that it is 1 when the partial description consists of two objects, one with one attribute, e.g. (A), and another one with 2, e.g. (B C), and before computing the right phrases the hypotheses including the phrases (A B C) and (A B) are considered. Then the computational cost is $\frac{4}{2} - 1 = 1$. This is the source of the deviation of the cost from 0 in the second 3000 interactions.

- 4 words: besides cases where the computational cost is 0, it also occurs that it is 1 similar than before when the partial description is (A B C) (D) and the phrases (A B C D) and (B C D) are explored before, CC = $\frac{4}{2} - 1 = 1$; 2 when the partial description is (A B) (C D) and the phrases (A B C D), (A B C), (B C D), (B C) are explored before the correct ones $\frac{6}{2} - 1 = 1$; 3 when (A B) (C D) and for example the phrases (A B C D), (A B C), (D), (B C D), (A) and (B C) are explored, which happens when the hypotheses (A B C) and (B C D) are consistent with the situation

- 5 words: There are cases where CC equals 2.5, 1 and 0. And the possible configurations are (A B C), (D E); (A B C D), (E); and (A B C D E) respectively.

So, the computational cost is negative only if communicative success fails, and it is greater than 0 when the optimal path to the solution is not expanded.

4.3 Acquisition Experiment

This section describes the acquisition experiment and presents and discusses the results.

4.3.1 Main Experiment: Acquisition by SBP

This is the experiment that evaluates the main model. The proposed learning operators are tested for the task of acquiring a phrase structure grammar by the syntactic-based pattern strategy (SBP), i.e. by acquiring a syntactic-based pattern system (SBP).

In the main experiment the learning operators proposed are tested in a setup where the tutor agent is again endowed with a SBP and the learner acquires a SBP.

Ontology Size

The ontology consists only of predicates that have one argument and is given by 5 semantic types and 2 properties each of them.

Semantic Types A	Properties: P
a-1	p-1-1 p-1-2
a-2	p-2-1 p-2-2
a-3	p-3-1 p-3-2

There are no incompatibilities between properties and the repetition of attributes in the same object description are not allowed.

Utterance length

words = 3

Population Size

2 agents.

Situation size

The situation size equals the number of attributes in the situation. So, it is dependent on the ontology size given that two objects with exactly the same attribute values are not allowed, and then the size has a limit. Moreover, the situation size can be set up as a dependent value of the utterance length, in this case, I set up the situation size as two times the utterance length.

Generator of the situation

The situation consists of two to four objects whose attributes sum up to 6 after two random selections from (2 1) and (3); an object of two attributes and an object of one attribute, or an object of three attributes.

Moreover, the values of attributes (i.e. properties of semantic types) are introduced sequentially as:

- Interaction 1: p-1-1 and p-2-1
- Interaction 25: introduction of the attribute value p-3-1 to be selected to take part in a situation.
- Interaction 51: introduction of the attribute value p-1-2 to be selected to take part in a situation.
- Interaction 101: introduction of the attribute value p-2-2 to be selected to take part in a situation.

- Interaction 151: introduction of the attribute value p-3-2 to be selected to take part in a situation.

Besides the parameters determining the ontology, the utterance length and the situation size, two more parameters have to be set to determine the application of syntactic coercion:

- the number of categories that can be omitted
- the criterion to choose between more than one construction partially matching.

In the design of the model I consider the application of syntactic coercion when only one category is missing; and if more than one construction is partially matching, I prioritize the one that would result in adding a category to the lexical construction that currently introduces more categories.

4.3.2 Results

The results of the experiment measure all the measures defined in section 1.3.4. In comparison with section 4.2 here the measures of *number of phrasal constructions*, *phrasal alignment* and *expressive power* are also considered. Results are averaged from 5 independent runs, and the sample standard deviation is shown in a gray ribbon.

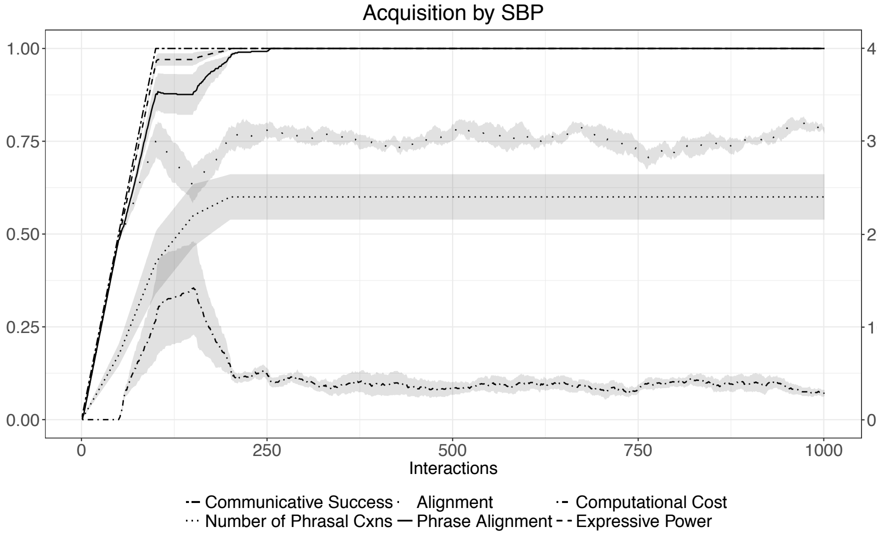


Figure 4.5: The SBP language strategy successfully guides the learner agent towards the complete knowledge of the language of the tutor. After 200 interaction approximately all the phrasal constructions are acquired and expressive power, phrase alignment, alignment and computational cost stabilize.

Communicative Success (CS) and Expressive Power (EP)

CS is 1 from the very beginning and EP is 1 once grammar is acquired. CS is always 1 because it is 1 once grammar is acquired and the amount of interactions before this happens is not large enough to encounter situation where it could be a failure as the ones described in section 4.2.2.

Computational Cost (CC)

Moreover, in this case the CC is optimal in the vast majority of situations. The cases of negative values are the same as the cases of failure in communicative, because they are cases where the situation includes the objects (A B C) and (A) (B C), the topic descriptions are the latter, and the utterance transmitted by the tutor happens to be the correct order for the description of (A B C).

The other cases of value different than 0 has to with case where the hypotheses (A B) is considered in (A) (B C) and consistent with the situation.

Number of phrasal constructions

The number of phrasal constructions equals 2 for some runs, and 3 for some other runs, that's why the resulting curve lies between 2 and 3.

Actually, in the optimal cases the learner acquires only two phrasal constructions of more than one constituent. Simply one for 2-word phrases, and one for 3-word phrases.

In the remaining of the section I take a closer look to the resulting grammars in order to understand these results.

Resulting Grammars

In most cases the resulting system equals a system where lexical categories are directly identified with predicate types (semantic types), but eventually also improves it and acquires only 1 construction for 2-word phrases, see figure 4.8.

Agents find systems based on categorizations which are more abstract than semantic categorizations, this is a very good result because they require less categories, and because this categorization is motivated by syntax which is closer to the definition of lexical categories provided in the first chapter of the thesis, where lexical categories are sets of words behaving similarly in terms of syntax. Figure 4.6 compares the distribution of categories for one of the grammars resulting from the experiment against a semantic-based pattern system (SemBP).

Figure 4.7 compares again a resulting grammar against a SemBP, in this case, the distribution of lexical categories over phrasal constructions is shown.

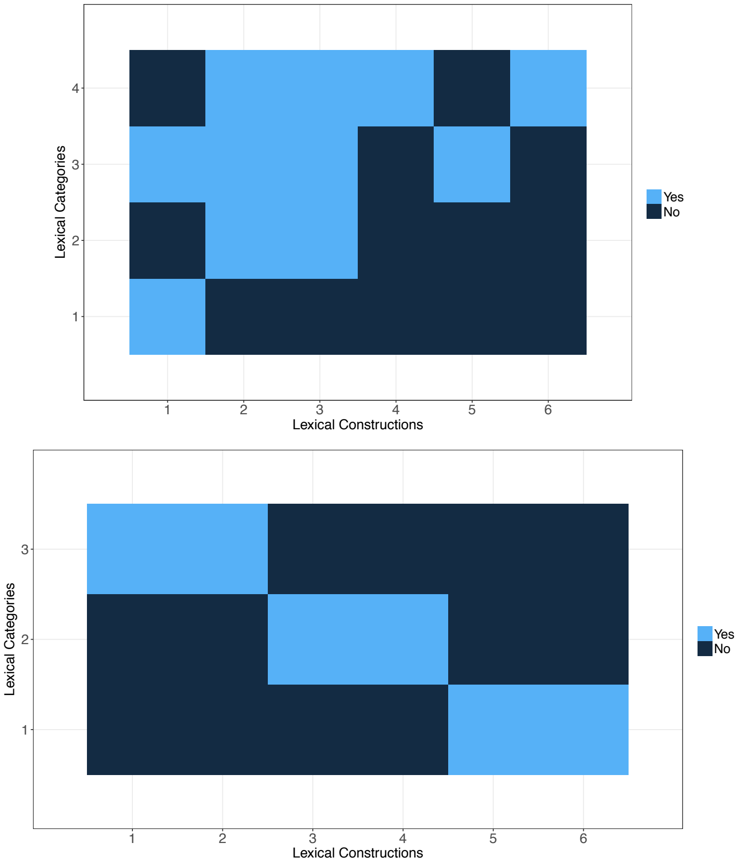


Figure 4.6: The distribution of lexical categories over lexical constructions varies from that of SemBP. Top: distribution of a system acquired by the learner. Bottom: distribution of a SemBP system.

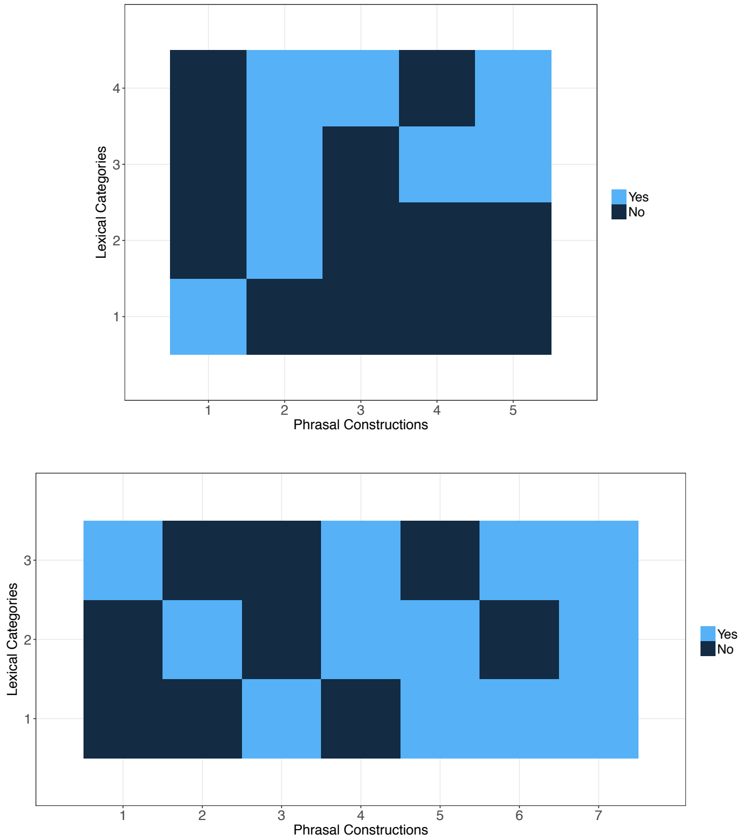


Figure 4.7: The number of patterns is inferior than in SemBP systems. Top: distribution of a system acquired by the learner. Bottom: distribution of a SemBP system.

In this case notice that the SBP grammar includes also the single noun-phrases.

For fully-connected attribute compatibilities it doesn't improve much with respect to a system based on semantic types (SemBP), although for other configurations does it. The learner eventually finds a system that requires only two patterns and 3 categories, depending on the sequence of sentences he is exposed to. Moreover, SBP systems may scale constantly for an increasing number of semantic types if their combinations are configured in particular topologies, as it is illustrated in figure 4.9, although in general they scale exponentially as WBP and SemBP, and lay slightly below SemBP.

Assuming that all lexical constructions mapping predicates of the same type have the same lexical categories, given that 3 different semantic types (attributes) were considered, the grammar found by the learner agent can be illustrated as follows in figure 4.8:

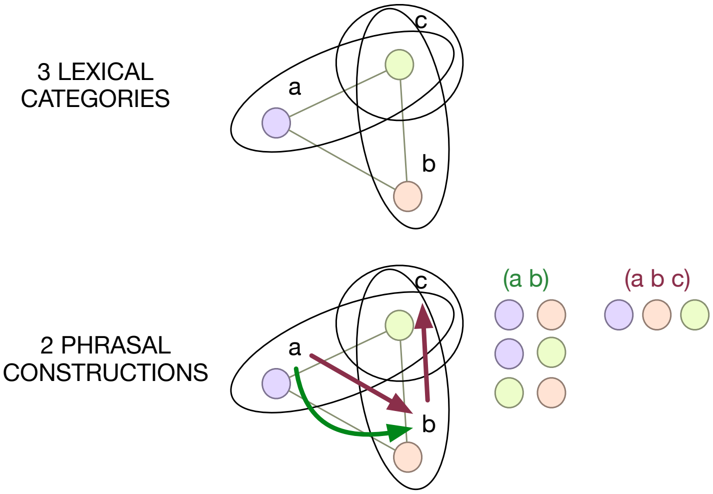


Figure 4.8: The learner is capable to find a system which if fully aligned with the system of the speaker and requires less resources than a system based on semantic categories.

Therefore, in the previous representation of SBP systems, a fully aligned SBP is represented as a set of sets (categories) covering all the nodes, and a set of category sequences, such that for all paths in the graph up to the maximum size of objects/patterns, there exists only one sequence.

Reducing the number of resources with respect to a SemBP system is more difficult as the number of semantic types increases. However, when ontologies include semantic incompatibilities (see figure 4.9) is feasible although the scaling of number of phrasal constructions and lexical categories is still exponential.

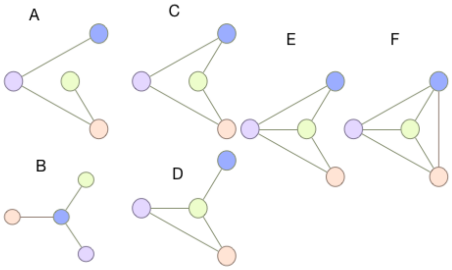


Figure 4.9: Representation of the attribute compatibility configurations for 4 attributes. Only the pairs of attributes that have an edge in between are compatible.

Figure 4.10 shows in more detail the grammar found by the learner agent in 5 different runs where the ontology includes only the 3 semantic types described above and with fully compatibility among attributes. The grammar in the first row is the same as the one represented in figures 4.6 and 4.7.

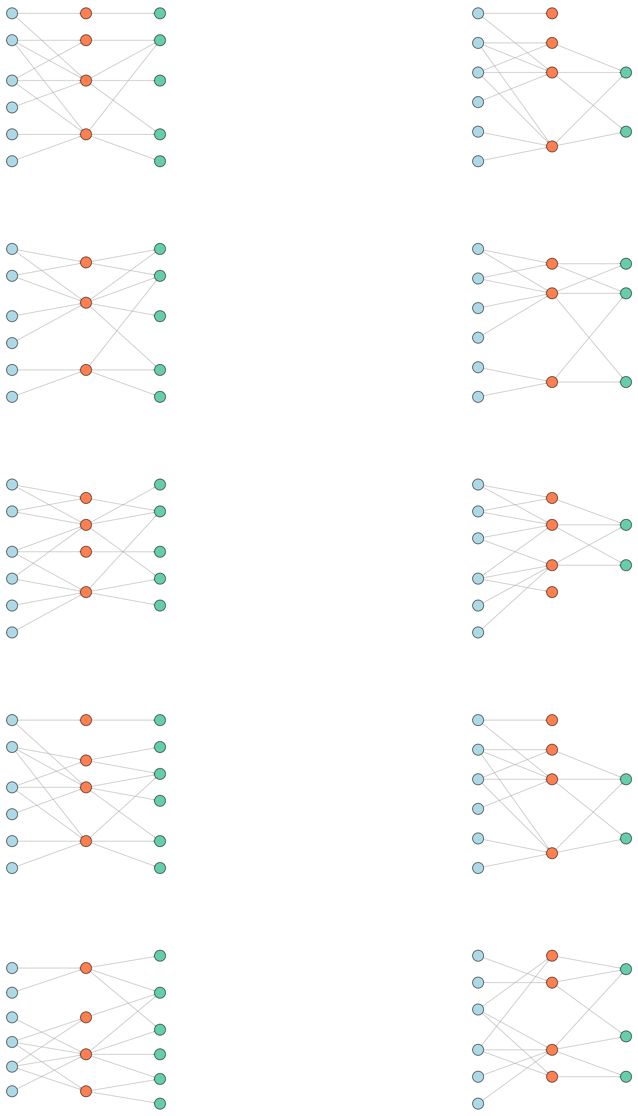


Figure 4.10: Representation of the learner grammar for 5 different runs. Blue: lexical constructions, red: lexical categories, green: phrasal constructions. Left: phrasal constructions with a single constituent are included. Right: without including phrasal constructions with a single constituent.

Finally, I show some examples of FCG constructions from a resulting grammar:

?np-unit
referent: ?referent
subunits: {?unit-1,?unit-2}

←

?np-unit
∅
form: {meets(?unit-1, ?unit-2)}

?unit-1
referent: ?referent
potential-syn-cat: {syn-cat-198}
referent: ?referent
potential-syn-cat: {syn-cat-198}

?unit-2
referent: ?referent
potential-syn-cat: {syn-cat-199}
referent: ?referent
potential-syn-cat: {syn-cat-199}

(syn-cat-198 syn-cat-199)

?np-unit
referent: ?referent
subunits: {?unit-1, ?unit-2, ?unit-3}

←

?np-unit
∅
form: {meets(?unit-1, ?unit-2), meets(?unit-2, ?unit-3)}

?unit-1
referent: ?referent
potential-syn-cat: {syn-cat-198}
referent: ?referent
potential-syn-cat: {syn-cat-198}

?unit-2
referent: ?referent
potential-syn-cat: {syn-cat-200}
referent: ?referent
potential-syn-cat: {syn-cat-200}

?unit-3
referent: ?referent
potential-syn-cat: {syn-cat-199}
referent: ?referent
potential-syn-cat: {syn-cat-199}

(syn-cat-198 syn-cat-200 syn-cat-199)

?fared-1
referent: ?o-1
potential-syn-cat: {syn-cat-198}

←

?fared-1
meaning: {a-1(p-1-1, ?o-1)}
form: {string(?fared-1, fared)}

(fared-cxn)

$$\left[\begin{array}{l}\textbf{?gaisa-1}\\ \hline\hline \text{referent: ?o-1}\\ \text{potential-syn-cat: }\{\text{syn-cat-199}\}\end{array}\right] \leftarrow$$

$$\left[\begin{array}{l}\textbf{?gaisa-1}\\ \hline\hline \text{\# meaning: }\{\text{a-3(p-3-1, ?o-1)}\}\\ \hline \text{\# form: }\{\text{string(?gaisa-1, gaisa)}\}\end{array}\right] \qquad \text{(gaisa-cxn)}$$

$$\left[\begin{array}{l}\textbf{?jejap-1}\\ \hline\hline \text{referent: ?o-1}\\ \text{potential-syn-cat: }\{\text{syn-cat-198, syn-cat-199, syn-cat-200}\}\end{array}\right] \leftarrow$$

$$\left[\begin{array}{l}\textbf{?jejap-1}\\ \hline\hline \text{\# meaning: }\{\text{a-2(p-2-2, ?o-1)}\}\\ \hline \text{\# form: }\{\text{string(?jejap-1, jejap)}\}\end{array}\right] \qquad \text{(jejap-cxn)}$$

$$\left[\begin{array}{l}\textbf{?nobo-1}\\ \hline\hline \text{referent: ?o-1}\\ \text{potential-syn-cat: }\{\text{syn-cat-199, syn-cat-201}\}\end{array}\right] \leftarrow$$

$$\left[\begin{array}{l}\textbf{?nobo-1}\\ \hline\hline \text{\# meaning: }\{\text{a-3(p-3-2, ?o-1)}\}\\ \hline \text{\# form: }\{\text{string(?nobo-1, nobo)}\}\end{array}\right] \qquad \text{(nobo-cxn)}$$

4.4 Discussion

This chapter reports on computational experiments in which a learning agent incrementally acquires a grammar from a tutoring agent through situated language game interactions.

Most models of language learning use a form of Bayesian unsupervised grammar learning operating over large amounts of data [16, 29]. However, the present model supports a complementary approach based on mechanistic explanations for language acquisition. Hence, I propose the implementation of specific learning operators as models for mechanisms sufficient for the acquisition of phrase structure.

The agent-based model of this chapter is focussed on the learner, which has always the role of hearer. Therefore, the learning operators studied are adoption operators, given that they allow the learner to infer constructions from the tutor, and hence, to adopt them. On the other side, invention operators are those operators allowing agents to invent new constructions, and these operators have not been considered yet here.

Finally, the model includes the first example of an artificial language evolution model, where adoption operators are divided between flexible grammar and learning actions, a trend that is followed in the coming chapters also for invention operators. In other models, learning agents build new constructions to enable their grammars to explore hypotheses beyond their previous scope.

A direct consequence of the adoption operators design is that all the information needed to build or expand a construction is taken directly from the final linguistic structure. This has some important conceptual implications when it comes to the application of the replicator dynamics theory for grammar variants, because it provides a potential low-level implementation for a mechanism for replication.

Finally, recall that an online web demonstration available through www.biologiaevolutiva.org/lsteels/tesi-final/acq provides further details on the model.

Chapter 5

EMERGENCE OF PHRASE STRUCTURE

Luc Steels and Emília Garcia Casademont.
Published [97]
Luc Steels and Emília Garcia Casademont.
Published [98]
Emília Garcia Casademont and Luc Steels.
Published [50]
Emília Garcia Casademont
Published [47]

In the field of artificial language evolution, the emergence of a particular language system, in the case of the thesis a language system exhibiting phrase structure, is seen as a process of self-organisation of the agents in the population. The term self-organisation here captures the idea of producing global phenomena under the use of local interaction, where the local interactions are communicative interactions modelled as language games.

In this chapter, I report on experiments for the emergence of first-order phrase structure by expanding the model discussed in the previous chapter. The goal of the model is to demonstrate how phrase structure can emerge as a result of selective pressures on the computational cost of language processing.

This hypothesis contrasts other hypotheses that have been studied in the context of the emergence of syntactic structure. One widespread hypothesis (following from research in Iterated Learning) is that syntactic structure is motivated by overcoming the transmission bottleneck from one generation to the next [83].

In contrast, here I argue that syntax arises from the need to *avoid combinatorial explosions in parsing and reduce semantic ambiguity in interpretation*, being the first the source for computational cost and the latter, the source for computational complexity of a language strategy. Moreover, the way structure arises is not through a transmission bottleneck but by language strategies and operators for the stepwise invention, adoption and alignment of linguistic conventions in a population, based on a *cultural selectionist dynamics* [87],[93]. So I seek a functionalist explanation where phrase structure emergence is motivated by selective pressures on the reduction of the computational cost and computational complexity.

Further details on the model can be found in the interactive web demonstration on the website www.biologiaevolutiva.org/lsteels/tesi-emilia/ps

5.1 Learning Operators for the Emergence of Phrase Structure

For the model proposed I consider again the same syntax game used in chapter 4, but in this case all agents are initialized as the learner agent, without a given language system. Therefore agents need to be able to use a CI operator to explore and create phrasal constructions and apply SC operator also in production because they will also get the role of speaker. The invention operators are the operators proposed to fulfill this role. Moreover, the application of these invention operators do lead to competing variants in grammar in contrast to the model in chapter 4. Therefore, linguistic consensus is not assured and alignment operators are also required to regulate the priorities among constructions to use.

The aim of the experiments is to build and test computational models to validate the hypothesized selective pressures. With this respect, the core of the model is found precisely in the alignment operators, which reinforce constructions requiring less computational power and punish constructions requiring more.

I chose this language game because it is the minimal task that generates communicative ambiguity given an ontology containing only first-order predicates.

5.1.1 Invention Operators

The application of the adoption operators CI and SC presented in chapter 4 is now available also for agents in the role of speakers as invention operators. For OV operator this is not the case because it can only be applied in parsing.

The only difference in the case of the CI operator is that agents select a random order instead of using the order found in the utterance, given that there is no utterance yet. Figure 5.1 shows an illustration of the CI and SC operators in invention.

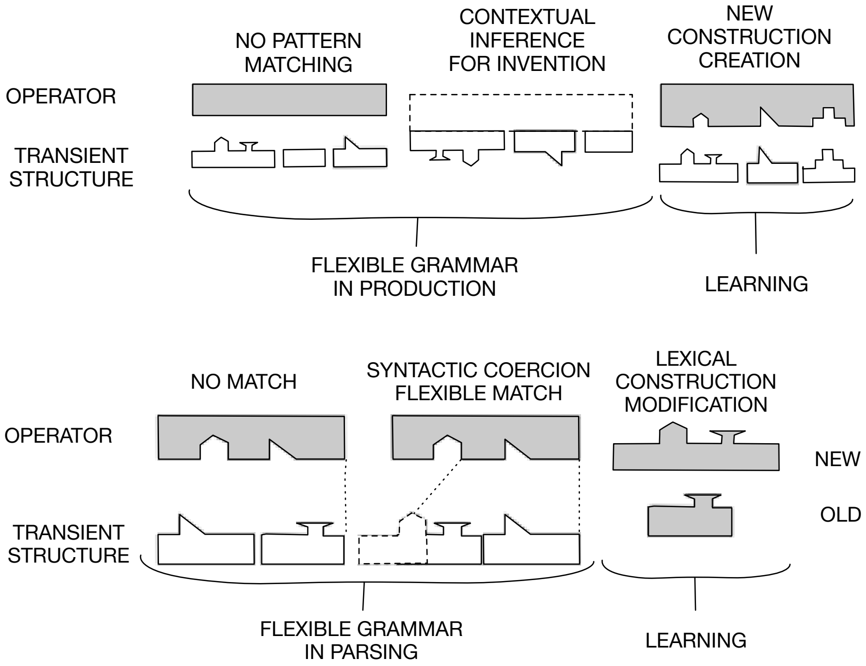


Figure 5.1: Invention Operator (top): The steps are similar to the contextual inference used for adoption, but this time the operator is used in production (formulation) instead of in parsing, and the constituents are ordered in a random order. There are no differences with respect to the creation of a new pattern.
Syntactic Coercion (bottom): In this case the difference is that the operator imposes the order according to the corresponding construction, instead of requiring it; and as it happens in the case of flexible grammar in parsing, the constraint of one syntactic category can be omitted.

In the case of SC the only difference is that any phrasal construction is capable to combine the corresponding lexical units omitting one of the syntactic constraints given by a lexical category. As SC acts in production, it imposes the constituent order of the corresponding construction, instead of requiring it.

Invention and adoption operators in combination with the lexicon design define a space of all the possible grammatical constructions that they can create. A question of investigation is how this space is explored along the evolution of a grammar inventory. Given that invention and adoption operators may reuse information from other constructions in the inventory to build new ones, I started to study the dynamics of such transmission of information in [49].

5.1.2 Alignment Operator

In the case of the acquisition experiments reported in the previous chapter, alignment operators were not required given that the adoption operators already guided the learner to the acquisition of the desired grammar. In the present case the situation is different because I model the formation of a shared grammar, and all agents can have the role of speaker or hearer. This means that they all can put new grammatical construction in circulation resulting into different phrases being used for the same meaning. Hence *variation* occurs due to the effect of invention operators.

All agents in the population are endowed with invention operators and any agent can interact with any other agents, which results into the unavoidable generation of competing phrases for the same meaning. Agents when having the role of speakers create new language or use language they've previously acquired, and agents when having the role of hearers infer the meaning of the utterances in interactions and potentially acquire new language.

Alignment operators in artificial language evolution models usually use alignment strategies where weights, scores or usage frequency [78, 28, 101, 27, 96] are read as priorities for constructions to apply in language processing. These priority orders are modified at the end of every interaction according to the communicative outcome. It is a design choice whether both speaker and hearer update the priorities in their grammars or only the hearer does. For the case of lexical systems it has been shown that updating only the hearer improves the results of convergence [3]. Usually in models of grammar evolution constructions successfully applied in the solution branch of the search tree are strengthen and their competing constructions are weaken.

Competing constructions, or simply called competitors, are grammar variants of another construction generally either because they use a different linguistic form to express the same meaning (meaning competitor), or because they use the same construction form to express another meaning (form competitor).

The computation of competing constructions is trivial for some grammars, e.g. in the case of WBP, where construction form equals a sequence of words, meaning competitors are constructions whose construction form is a permutation of this sequence, and form competitors don't exist. For example for the phrase *red ball* the only meaning competitor in WBP would be *ball red* which can be computed based on the construction form.

Therefore, there is no problem in identifying them offline the search space.

However, in the case of SBP they are not trivially given by the construction forms of phrasal constructions, because the applicability of a construction relies on the rest of the grammar, i.e. lexical and phrasal constructions. E.g. given two lexical constructions whose potential syntactic categories are *syn-cat-1* and *syn-cat-2* for the first one, and *syn-cat-3* and *syn-cat-4* for the second one; the grammatical constructions *(syn-cat-1 syn-cat-4)* and *(syn-cat-2 syn-cat-3)* are meaning competitors although their construction forms cannot be trivially derived from one to the other.

In order to compute competing constructions in SBP I propose instead an alignment operator which actually implements selective forces on the computational cost by favouring those constructions reducing it, and weakening those that have increased it for the current interaction.

I do it by determining competitors online. Given a construction application, when the resulting branch on the search tree of language processing has not lead to a solution, such construction will be weakened and treated as a competitor of the construction that applied in that node which is part of the path to the solution. And viceversa, those constructions that were used on the path towards the final transient structure are strengthen. See figure 5.2 for an example.

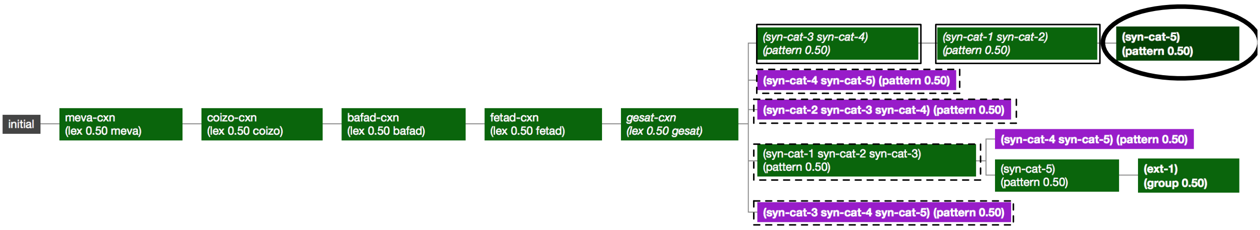


Figure 5.2: The scores of those constructions that started off a wrong branch in the search tree during comprehension (*parsing* together with online *interpretation*) are decreased because those are the constructions that increased the computational cost in language processing. A wrong branch is a branch which was not on the path to the final solution. The solution node is marked with a circle and the nodes generated by constructions that will be strengthen, with a solid line, while the ones generated by constructions that will be weaken, with a dashed line.

Therefore, the proposed alignment is defined and applied only for the hearer. I call this operator the *cost selection* alignment operator.

I use a system of scores to track the changes in priorities by updating the score of constructions using the lateral inhibition learning rule [111]: $\sigma_{c_i} \leftarrow \sigma_{c_i}(1-\gamma)+\gamma$ for increasing, and $\sigma_{c_j} \leftarrow \sigma_{c_j}(1-\gamma/2)$, for decreasing, with $\gamma = 0.2$. σ_{c_i} represents the score of construction c_i and all scores are initially set to 0.5 when the constructions are created. Therefore they have a minimum at 0 and a maximum at 1. See figure 5.3 to visualize the effect of this rule on the scores.

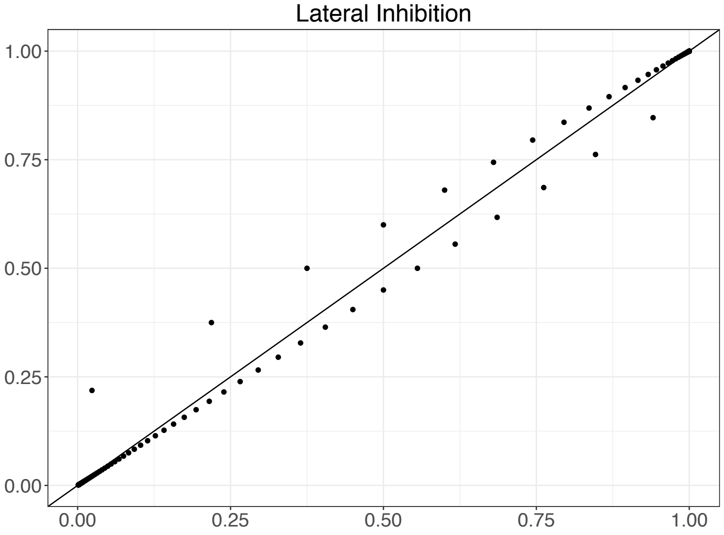


Figure 5.3: The effect of lateral inhibition as it is used in the experiment. x coordinate corresponds to the score before the application of the rule and y coordinate corresponds to the score after the application of the rule. Therefore, points below the $x = y$ line represent score values decreasing and points above the $x = y$ line represent score values increasing.

The scoring system models the success of a grammar variant in being used by the communicative agent, however it cannot be directly interpreted as a population growth for that grammar variant with respect to competitor variants, because on one side scores are only modified when the corresponding constructions are used, and on the other, competitor scores don't sum up to 1. Diffusion mechanisms could be explored in combination to the proposed selective force in order to assimilate the result of applying an alignment strategy to a population growth for a grammar variant, and deriving an stochastic model with known properties that could be interpreted. However, this is not an easy task because heuristics are relevant to language processing in the kind of grammars studied here, where a chain of constructions is needed to get to a solution every time that grammar is applied, instead of a single construction.

5.1.3 Bias in Language Processing

Constructions for application are ordered by length and scores, and flexible grammar application of constructions with higher score and longer length has priority over grammar application of shorter constructions or with lower score values.

Finally, when all constructions are tried as both grammar application and flexible grammar application and still no solution is found, the contextual inference (CI) operators are used. Figure 5.4 illustrates the priority of constructions in grammar application and learning operators in flexible grammar application.

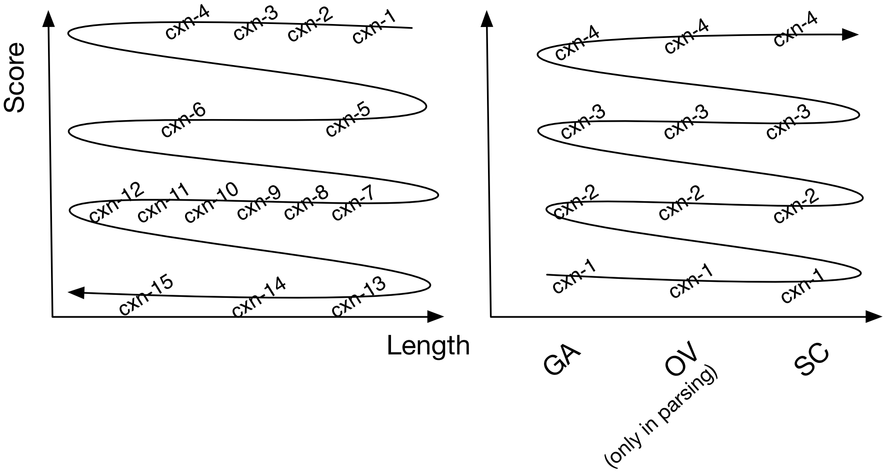


Figure 5.4: Phrasal construction priorities for grammar application depends on the length and the score of the constructions, and the flexible application of constructions by means of the learning operators ordering variation (OV) and syntactic coercion (SC) comes first than the grammar application (GA) of constructions with lower scores or shorter length.

5.2 Baseline Experiments

The focus of the main experiment here is the effect of the alignment operator proposed. This operator implements selection on the variation generated by the application of the invention operators. Adoption operators were already tested in chapter 4.

Two baseline experiments are proposed: one with the alignment operator disabled and a second one considering an alignment operator which computes competing constructions as permutations of the construction form.

5.2.1 Experiment 1: No Alignment Operator

In this experiment, agents are initialized with adoption and invention operators, but not with the alignment operator.

Ontology Size

Semantic Types A	Properties: P
a-1	p-1-1 p-1-2
a-2	p-2-1 p-2-2
a-3	p-3-1 p-3-2

There are no incompatibilities between attributes.

Utterance length

words = 3

Population Size

5 agents.

Notice that the case with only 2 agents would not generate grammar variants and instead it would be a very similar case than the ones studied in 4.3. The only difference is that in 4.3 one of the agents is already endowed with a fully-aligned grammar and in this case they both would be acquiring the same grammar alongside.

Situation Size

Two times the number of words, i.e. 6.

Generation of Situations

The generator of situations is the same that was used in chapter 4, although in this case the utterance length is always 3 and the situation size is 6 (two times the utterance length). Additionally, agents acquire phrasal constructions to express the meaning of objects with 3 attributes before than the ones to express the meaning of objects with less attributes.

This design choice is justified by the fact that when the operators provided are active, mixing the learning of 2 and 3-word phrases gives no option to converge to the optimal grammar found in chapter 4. The introduction of new categories happens only when no constructions can apply, further experiment configurations should be studied to improve the abstraction operation of creating a new category.

Setup in which phrasal constructions of length 3 are learned before. Only from interaction 600, agents are allowed to learn phrasal constructions of length 2.

- Interaction 1: three attribute values (properties) of three distinct attributes (semantic types) can be used to generate the situation. I represent them as p-1-1, p-2-1 and p-3-1, where the first number is an index for the attribute and the second, for the value.
- Interaction 51: introduction of the attribute value p-1-2 to be selected to take part in a situation.

- Interaction 101: introduction of the attribute value p-2-2 to be selected to take part in a situation.

- Interaction 151: introduction of the attribute value p-3-2 to be selected to take part in a situation.

5.2.2 Experiment 2: Alignment using Form Competitors

The second baseline experiment assumes an alignment operator which identifies meaning competitors as form permutation competitors, i.e. constructions using permutations of the same sequence of lexical categories. Moreover, agents are also endowed with the adoption and invention operators described.

The configuration of this experiment is the same than the previous one.

5.2.3 Results

In the first baseline experiment, invention and adoption operators are active, but the alignment operator is not.

The results measure the same 6 measures as in section 4.3.2:

1. The **communicative success** (CS) of an interaction is 0 when the game is a failure and 1 when it is a success.

2. **Alignment** measures the consensus between agents. In every interaction it is checked which utterance would have the hearer used to express the same meaning. If both utterances are different, alignment gets the value 0, and if they are equal, it gets the value 1.

3. **Phrase alignment** is similar to alignment but instead of comparing whole sentences I compare phrases individually.

4. The **number of phrasal constructions** (Number of Phrasal Cxns) measures the number of phrase structure rules in a population.

5. **Expressive power** measures the ratio of variable equalities that are due to phrasal constructions.

6. The **computational cost** is derived from the search space to find the solution of a grammar application. It is defined as the ratio between the number of processing steps and the optimal number of processing steps, minus 1. Where the optimal number of steps equals the number of expected phrases.

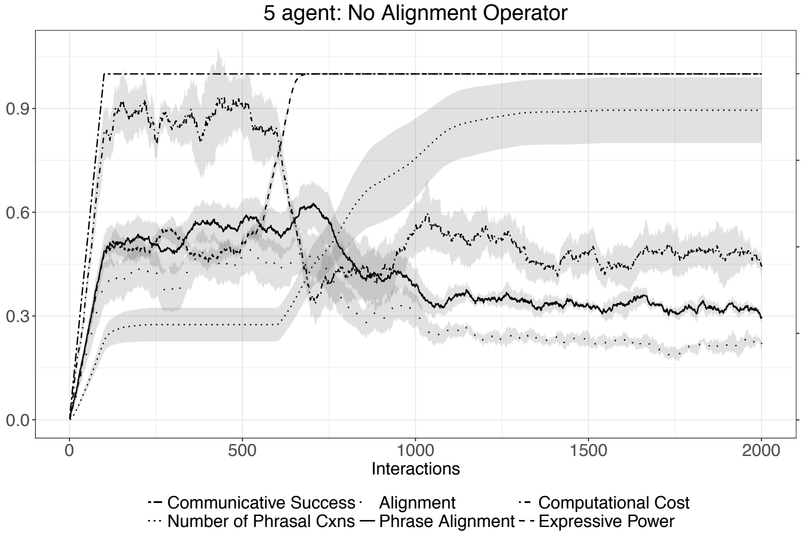


Figure 5.5: When no alignment is activated, but invention and adoption are, phrase alignment is worse than in the case of no invention nor adoption operators. The latter is equivalent to the baseline experiment in section 4.2.

Figure 5.5 shows the results of the first baseline experiment where the following can be observed.

Communicative Success (CS) and Expressive Power (EP)

CS immediately gets to its maximum and EP is around 0.5 while it is not allowed to acquire 2-word phrases (which are present in half of the situations half) and as well it gets immediately to 1 after interaction 600 where it is allowed that agents acquire the remaining phrases.

Phrasal Alignment (PA) and Alignment (A)

PA and A get stabilized to low values because several variants of the same phrase are considered and in use from interaction 600 onwards. For some runs they acquire all the possible variants exhibiting the same behaviour as the baseline experiment in section 4.2.

Computational Cost (CC)

During the first 600 interactions CC is close to 0.9, and it ends being close to 0.5.

Number of phrasal constructions

Finally, the number of phrasal constructions grows up to a small interval centered in 35, where the variability comes from agents acquiring 6 to 8 constructions mostly.

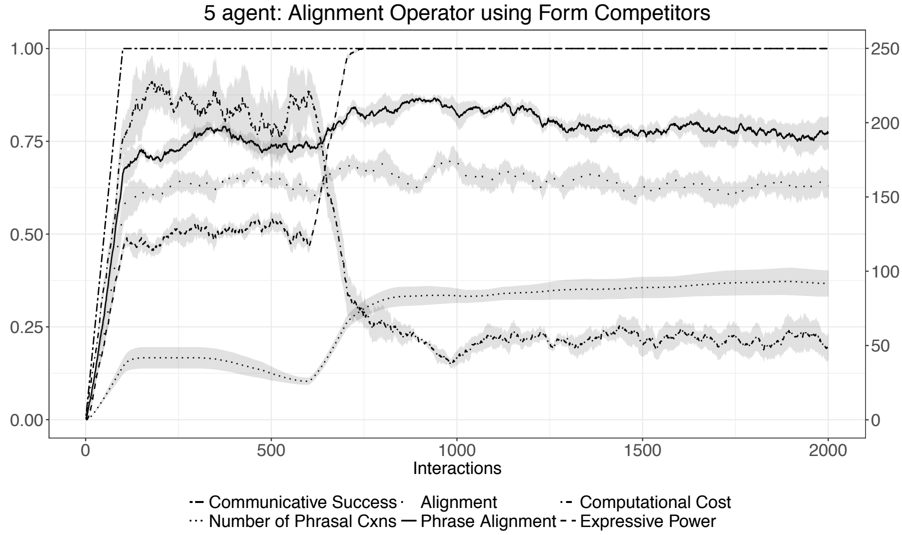


Figure 5.6: Again, when alignment is used as a form permutation competitor for more than 2 agents, they don't reach full phrase alignment.

Results from the second baseline experiment are shown in figure 5.6 and discussed below:

Communicative Success (CS) and Expressive Power (EP)

Again CS immediately gets to its maximum and EP is around 0.5 while it is not allowed to acquire 2-word phrases, and as well it gets immediately to 1 after interaction 600 where it is allowed that agents acquire the remaining phrases.

Phrasal Alignment (PA) and Alignment (A)

Alignment is slightly better that in the case of no operators, although there are competing constructions that are still kept in use until the end of the experiment.

Computational Cost

Computational Cost is slightly worse than before because although PA is better, the combination of keeping competing constructions and using an alignment operator for form permutation competitors facilitates the wrong usage of constructions. That is because not all the agents get aligned.

Number of phrasal constructions

Contrary to the improvement in phrase alignment the number of constructions that are acquired exceeds the number of constructions in the case of no operators, given that all agent at the acquire many more phrasal constructions than the optimal number for SBP.

If agents were applying a word-based pattern (WBP) strategy instead, they would reach communicative success and phrase alignment using the form permutation competitors. The reason is that the experiment would have the same dynamics as independent naming games.

The dynamics of the phrasal construction scores is another view that can be taken from the experiments. In figure 5.7 these dynamics are represented for the phrasal constructions of 3 agents in the experiment. While two of the agents don't exhibit apparent competition between constructions and scores only go up or down, the other agent gets a pair of competing constructions.

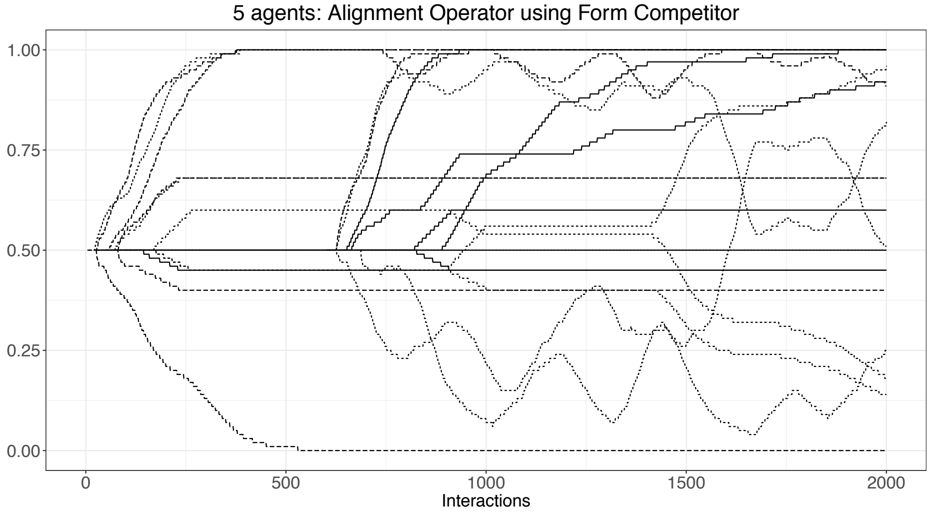


Figure 5.7: Dynamics of the phrasal construction scores for three agents in the experiment. Each agent is represented using a different line type. One agent keeps using a couple of pairs of competing constructions whose scores oscillate so that it depends on the period which is the construction that is prioritized.

Finally, the resulting grammars of a run of the first baseline experiment are shown in figure 5.8. In this case, competitor constructions are form permutation competitors, hence they are explicitly given by those constructions whose connections to lexical categories are exactly the same ones.

5.3 Formation Experiment

In this section I present and discuss the results for the experiments testing the performance of the model based on the operators proposed in allowing a population of artificial agent to self-orgranise an SBP system, where lexical categories and phrasal constructions are the means to align grammars.

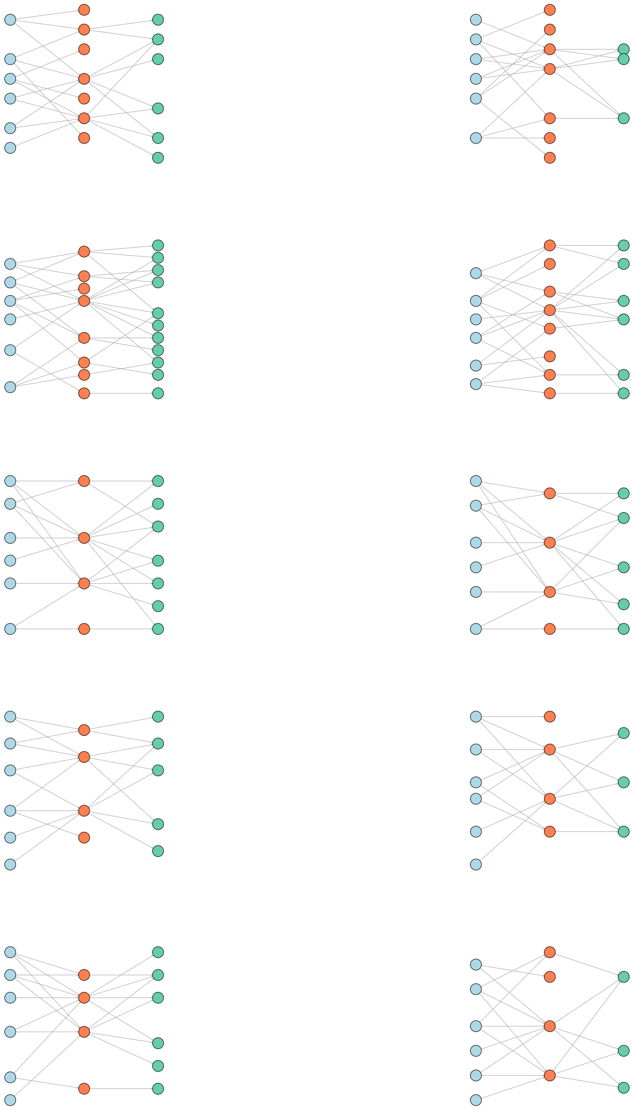


Figure 5.8: Representation of the agent grammars for a run. Blue: lexical constructions, red: lexical categories, green: phrasal constructions. Left: phrasal constructions with a single constituent are included. Right: without including phrasal constructions with a single constituent.

5.3.1 Main Experiment: Self-Organisation by SBP

Formation experiments test whether a population of agents is capable to construct its own language system from the given lexical system, by applying learning operators of a language strategy and agreeing on one of the possible language systems.

Ontology Size

Again 3 attributes and 2 values each.
There are no incompatibilities among attributes.

Utterance length

words = 3

Population Size

2 and 5 agents.

Situation Size

Two times the number of words, i.e. 6.

Generation of Situations

The generator of situations is the same that was used for the baseline experiments. And the introduction of attribute values slightly varies depending on the population size.

- Interaction 1: the three attribute values p-1-1, p-2-1 and p-3-1 are in use.
- Interaction 51 for 2 agents and 151 for 5 agents: introduction of the attribute value p-1-2 to be selected to take part in a situation.

- Interaction 101 for 2 agents and 251 for 5 agents: introduction of the attribute value p-2-2 to be selected to take part in a situation.
- Interaction 151 for 2 agents and 351 for 5 agents: introduction of the attribute value p-3-2 to be selected to take part in a situation.

5.3.2 Results

In every experimental run agents self-organise a particular language (i.e. particular word orders) implemented in their grammars by the means of phrase structure grammars. And the language strategy that allows agents acquiring a phrase structure grammar, together with its learning operators, is twofold selected by their power of reducing the cognitive effort in language comprehension.

2 agents

Results are from 5 independent experimental runs of 2000 interactions.

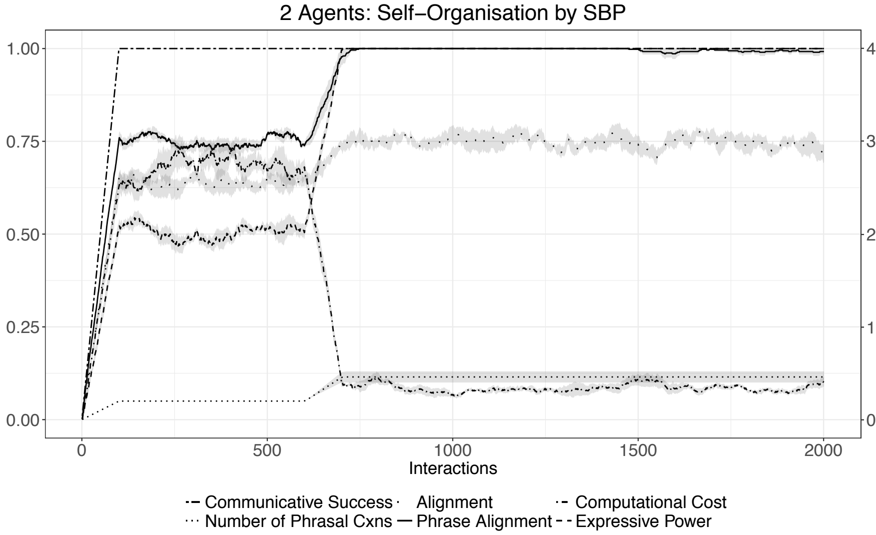


Figure 5.9: Agents reach communicative success and the size of the grammars are optimal. Alignment and phrase alignment don't converge. The expressive power reach its maximum. Phrase alignment doesn't decrease, and so syntactic ambiguity doesn't increase after no more construction are created.

In the first series of experiments two agents endowed with all the operators discussed were considered. Results are shown in figure 5.9, and the interpretation of the results follows:

Communicative Success (CS) and Expressive Power (EP)

Similarly than in the baseline experiments CS immediately gets to its maximum and EP is around 0.5 while it is not allowed to acquire 2-word phrases (half of the situations) and as well it gets immediately to 1 after interaction 600 where it is allowed that agents acquire the remaining phrases.

Phrasal Alignment (PA) and Alignment (A)

Before 2-word phrases are acquired, PA is close to 0.75 because for half of the situations is 1 and for half of the situations is 0.5. After 2-word phrases are acquired it reaches its maximum. Similarly, A is close to 0.625 before 2-word phrase are acquired because it is also effecting by the random order between constituents in the cases where 2 object meanings are express, and finally stabilizes around 0.75 (half of the cases is 1, and half of the cases of the remaining half is 1).

Computational Cost (CC)

CC is closed to 0.7 before interaction 600 but after 2-word phrasal constructions are acquired it goes down and stays close to 0.

Number of phrasal constructions

Up to interaction 600 each of the two agents acquire one construction for 3-word phrases and after interaction 600 either one or two more constructions are acquired for 2-word phrases as the examples seen in section 4.3.2.

Figure 5.10 shows the evolution of the scores of phrasal constructions for an experimental run. In this case, each agent has acquired a single construction for 2-word phrases. Notice that given that the search space of language processing in interpretation is always explored efficiently, the score of constructions never decreases.

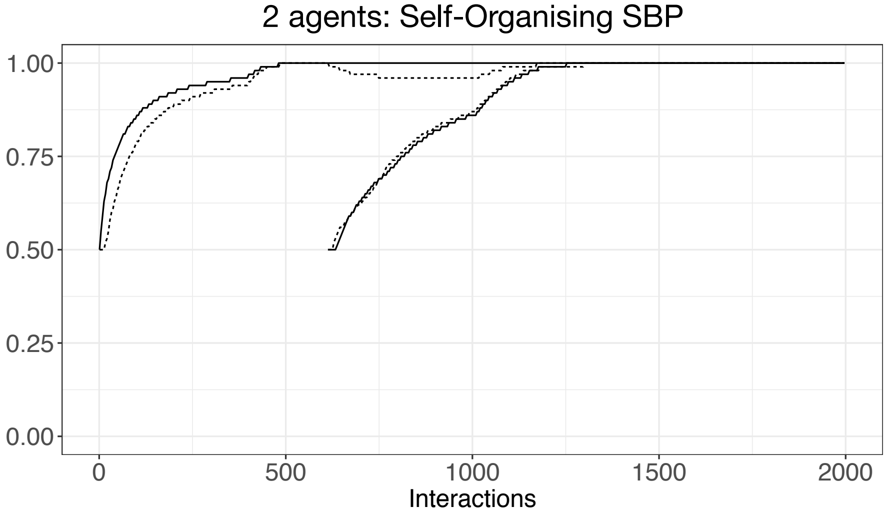


Figure 5.10: The evolution of individual scores for the 2 agents is shown. Each type of line represents the constructions of an agent. Each agent acquires one 3-word phrasal construction and one 2-word phrasal construction and no variants are generated.

Resulting Grammars

Figures 5.11 show the evolution of the two agent grammars along one experimental run.

Figure 5.11: Interaction 1, one 3-word phrasal construction is acquired.

Figure 5.12: Interaction 53. Syntactic coercion applies to one lexical category.

Figure 5.13: Interaction 104. Syntactic coercion applies to another lexical category.

Figure 5.14: Interaction 600. A 2-word phrasal construction is acquired.

Figure 5.15: Interaction 620. Syntactic coercion applies and expands the usage of the 2-word phrasal construction.

Figure 5.16: Interaction 802. Again syntactic coercion applies and expands the usage of the 2-word phrasal construction leading to the final grammar.

5 Agents

In the next series of experiments five agents endowed with all the operators discussed were considered. In this case, the experimental runs consisted of 5000 interactions. And the interpretation of the results is very similar than in the case where only two agents were involved. Results are shown until interaction 2000 in figure 5.17 to facilitate the comparison with the previous experiment, and because convergence is reached much before and no relevant information is lost.

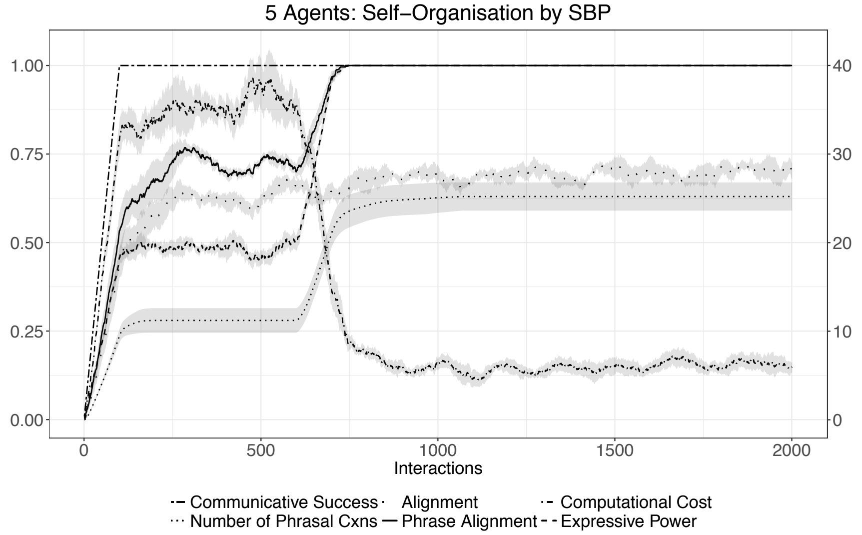


Figure 5.17: Results are shown until interaction 2000 so that they can be compared with the runs for 2 agents. Agents reach communicative success and the size of the grammars are not optimal. Alignment and phrase alignment don't converge. The expressive power reaches its maximum. Syntactic ambiguity (phrase alignment doesn't increase) doesn't decrease after no more constructions are created.

Communicative Success (CS) and Expressive Power (EP)

CS immediately gets to its maximum and EP is around 0.5 while it is not allowed to acquire 2-word phrases (half of the situations) and as well it gets immediately to 1 after interaction 600 where it is allowed that agents acquire the remaining phrases.

Phrasal Alignment (PA) and Alignment (A)

PA alignment reaches its maximum and A stays slightly below 0.75.

Computational Cost (CC)

CC is close to 0.9 before the interaction 600 and close to 0.15 when grammar is stabilized.

Number of phrasal constructions

Agents use more constructions than in the previous case and grammar variants are generated.

However, the effect of the competition among constructions is clearer when looking into the evolution of individual scores. Figure 5.18 shows the evolution of the scores.

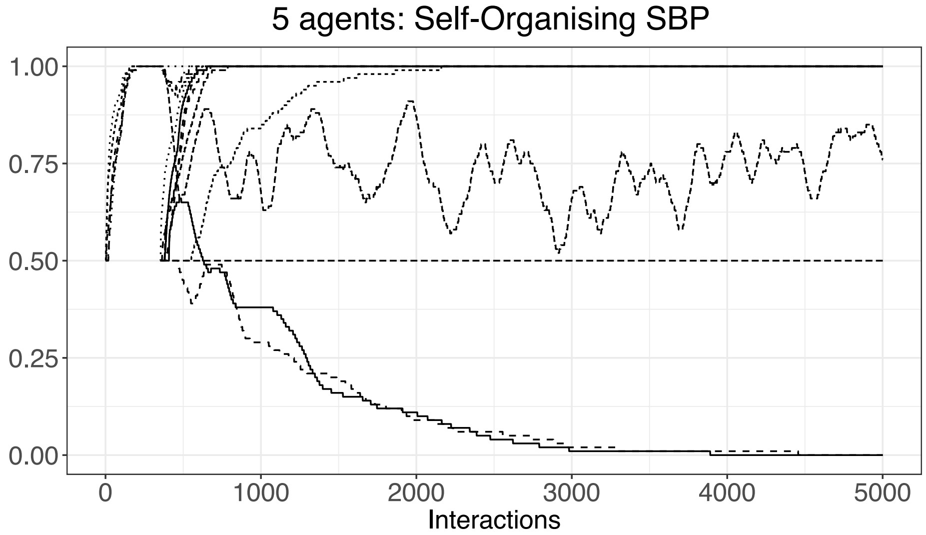


Figure 5.18: Again each line type represents an agent. The cost selection alignment acts in such a way that for those constructions that are not in use, the score value is constant and different than 1, and for the rest, is either 1 or oscillating. Construction scores oscillate when there always comes interactions where the application of the corresponding construction contributes to increasing the computational cost of processing an utterance.

Resulting Grammars

In this case, I also compare resulting grammar for different configurations of the generation of situations, particularly on how attribute values are introduced. Figure 5.19 compares pairs of agent at the end of an experimental run for three different configurations on the introduction of properties to be selected as part of the situation:

- First row: the same setup used in the experiments above.
- Second row: all the attribute values are considered from the begin-

ning of the experiment and agents learn 3-word phrasal constructions from the beginning and 2-word phrasal constructions after interaction 600.

- Third row: 2-word phrasal constructions can be acquired from the beginning and 3-word phrasal constructions after interaction 600. At the beginning, attribute values p-1-1, p-1-2, p-2-1 and p-2-2 are considered and afterwards attribute values p-3-1 and p-3-2 are introduced.

The size of the nodes representing phrasal constructions represents their score and this way the variation and selection on the phrasal constructions is clearly visualized. As expected, the setup used in the experiment is the one that generates less variation, given that it was selected to facilitate the learning path towards the optimal grammar found in the acquisition experiment in chapter 4.

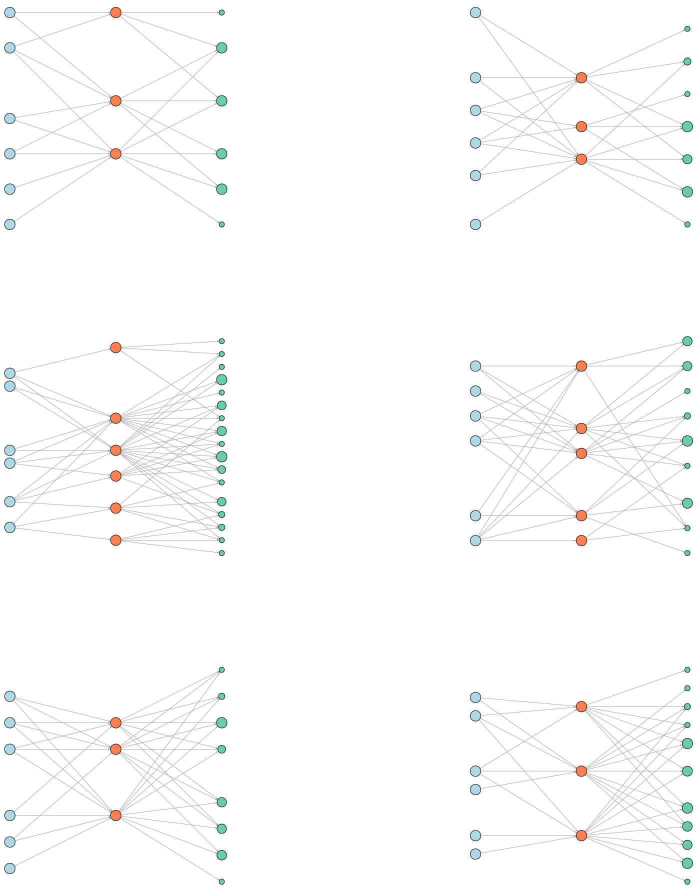


Figure 5.19: Comparison of the final grammars of three paris of agents from different experimental configurations. In this case, the size of the green nodes is proportional to the score of the corresponding phrasal construction.

Notice that the distribution of lexical categories across lexical constructions may vary between agents, and so may be for the way phrasal constructions combines the lexical categories. In figure 5.20 two agent grammars are compared

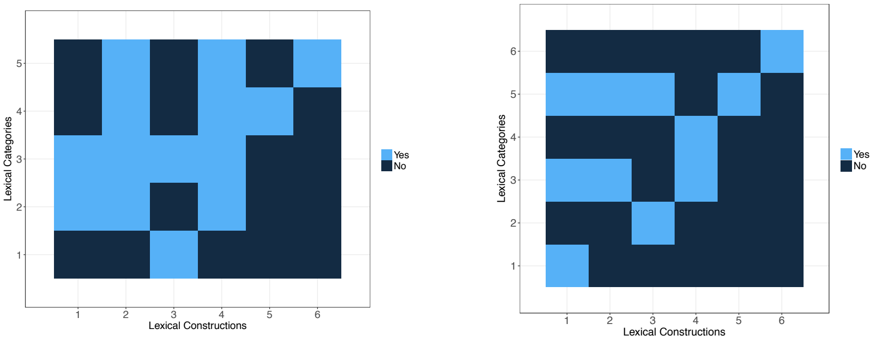


Figure 5.20: Comparison between two grammars from different agents of the same experimental run. Notice that they also differ in the number of lexical categories that are created.

The resulting FCG constructions are similar than the ones described for the acquisition experiment in section 4.3.2.

Learning Efficiency

Below the learning efficiency for phrasal constructions and for lexical categories is studied, i.e. the ratio of constructions (or lexical categories) acquired that are actually in use in the resulting grammar. Figure 5.21 plots the learning efficiency for phrasal constructions comparing the experiments involving 2 and 5 agents.

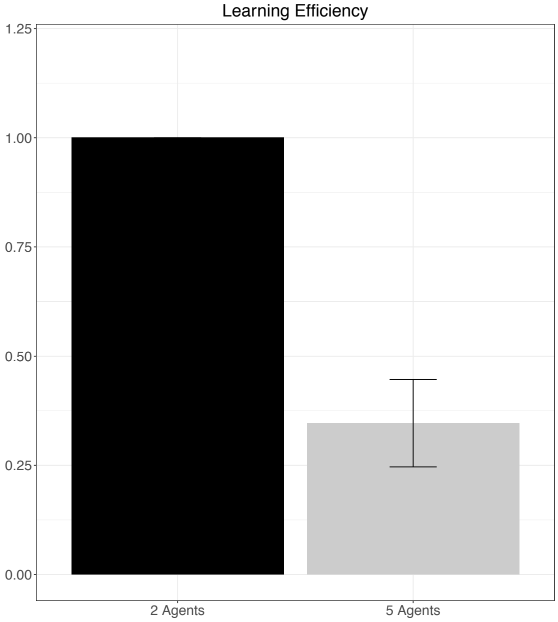


Figure 5.21: Learning efficiency of phrasal constructions. The ratio between the total number of phrasal constructions learned and the ones used when grammar agents converge.

On the other side, the learning efficiency of lexical categories is maximum, i.e. 1, because of the configuration used in the formation experiment as it comes to introducing new object properties in the situations.

5.4 Discussion

This chapter argues that syntax in the form of phrase structure is motivated by the reduction of the computational cost of semantic interpretation. It reports on a case study for the emergence and sharing of first-order phrase structure in a population of agents playing language games. First-order phrase structure combines words into phrases but do not yet generalize to hierarchical or recursive phrases. This is covered in the coming chapter.

The computational model proposed shows that a series of learning operators for invention, adoption and alignment lead to the emergence of phrase structure grammar in a population of artificial agents.

Both invention and adoption operators are based on the same fundamental operations. On the other side, the alignment operator requires a definition for competing constructions and rules to modify construction priorities in application.

In order to identify competing constructions I subscribe to concrete selectionist criteria that punish the increase of computational cost. Hence, competitors are determined online in language processing, instead of being offline determined.

In contrast, in the case of the Naming Game [112], competitors are explicitly given through strategies. That is because the strategies to play naming games are already meant to decrease synonymy and polysemy and the set of competing constructions can be offline explicitly computed, for example as the list of synonyms.

I test the hypothesized alignment operator by comparing an experiment without the alignment operator enabled and an experiment with an alignment operator explicitly determining competing constructions as meaning constructions given only by form construction permutations. While none of the latter leads to the emergence of phrase structure, the alignment favoring the reduction of the computational cost does. So, these selective pressures on the computational cost help phrase structure to arise. The pressures are part of the successful language strategy by the means of its alignment operator. This result supports the hypothesis that phrase structure is to some extend motivated by the need to decrease semantic ambiguity and avoid combinatorial explosion in parsing, as the main source of computational cost.

Recall that more details on the model and the operators discussed can be found in an interactive web demonstration in the site
www.biologiaevolutiva.org/lsteels/tesi-emilia/ps

In the coming chapter, the same ideas are applied to another agent-based model to study the case of higher-order phrase structure as a means to expressing ontologies including relations.

Chapter 6

EMERGENCE OF RECURSIVE PHRASE STRUCTURE

Luc Steels and Emília Garcia Casademont.
In Preparation [99]
Emília Garcia Casademont
Published [48]

The second hypothesis of the thesis states that the same language strategy which solved the problem of self-organising a first-order phrase structure grammar can solve the problem for recursive phrase structure grammars as well.

Human languages exploit recursive structure as a part of their grammar, as it was illustrated in the first chapter of the thesis. Particularly, they do it by the means of recursive phrase structure, which leads to unbounded syntactic compositionality that in turn empowers language to express unbounded semantic compositionality.

Even though there has been a debate about whether all languages use recursion to the same extent [36], or whether they all really use it [25], there is a widespread consensus that recursive phrase structure is a core property of many human languages [21]. Other animal communication systems (particularly bird song) may feature recursive syntax as well, but, as far as we know, syntactic structure does not express compositional meaning [10].

In this chapter I introduce a model that shows how a population of language agents endowed with the same limited set of grammar building operators as before is collectively able to create a shared complex phrase structure grammar exhibiting recursion, given a purely lexical system.

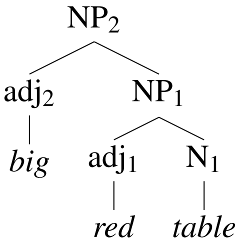
I will not address the question whether the mechanisms required for processing and learning recursive grammar require neuronal structures unique to language [41] or whether handling recursion is unique to the human lineage [53, 59], nor to what extend the cognitive abilities modelled by these operators are innate or culturally acquired.

The structure of this chapter is as follows. First, a more detailed description of recursion is provided; second, the application of the learning operators to the syntax game used in this chapter is discussed; third, the baseline experiments against which the main experiment is compared are presented and their results are studied; and finally, the last two sections are dedicated to investigate the performance of the learning operators by presenting and discussing the results of the main formation experiment.

Again there is an online interactive web demonstration on the site www.biologiaevolutiva.org/lsteels/tesi-emilia/rps to explore the details of the model.

6.1 Recursion

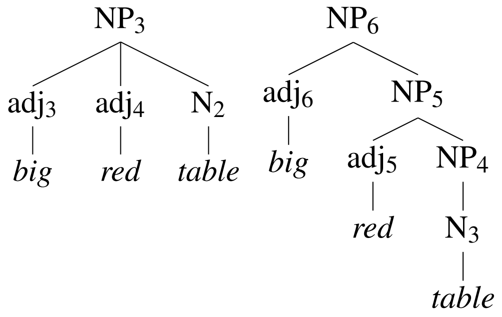
I study recursion in the form of recursive phrase structure. Recursion typically refers to *syntactic recursion*, i.e., the presence of syntactic constituents containing elements of the same syntactic category as the whole constituent. See figure 6.1, where the subindices just indicate different occurrences of the syntactic categories.



(6.1)

Figure 6.1: Example of syntactic recursion: $NP \rightarrow adj\ NP$

However, the linguistic analysis (the tree) depends obviously on the formalism and rules used to generate the linguistic tree. The same phrase could be analysed otherwise in the following two ways, as well as in various other ways:



(6.2)

Figure 6.2: Linguistic theories and formalisms determine the analysis of syntactic trees. In this case, two different syntactic trees come up depending on whether the rule *NP* $\rightarrow$ *N* or *NP* $\rightarrow$ *adj adj N* is used.

While the left tree in figure 6.2 doesn't exhibit recursion, the right tree exhibits it two times, both NP_5 and NP_6 are syntactically structural recursive.

Considering a sequence of rules that generates a linguistic tree, I define *processing recursion* as the situation in which a rule applies to an input containing an element which is on its own the result of the application of this same rule. Among the previous examples (figures 6.1 and 6.2), only the right tree in figure 6.2 exhibits processing recursion: the rule *NP* $\rightarrow$ *adj NP* applies to NP_5, which is on its own the result of applying *NP* $\rightarrow$ *adj NP* to adj_5 and NP_4.

Besides *syntactic recursion* for linguistic trees and *processing recursion* for sequences of rules generating linguistic trees, I also define *semantic recursion* for meanings.
Remember that in the models of the thesis, meanings are represented using predicate calculus. In this chapter, I consider the second ontology configuration, in which there are relational predicates introducing more than one predication. I use this configuration because it motivates the use of higher-order phrase structure (see section 2.2.3).

Each relational predicate introduces a variable for the relation itself, e.g., (cogn-action see ?r-2) represents a predicate ’see’, which is of semantic type ’cogn-action’, and introduces ’?r-2’ as the variable for the entity itself. Moreover, the arguments of a predicate relation have their own predications, each of which introduces another variable for the corresponding argument. E.g., (see-arg-1 ?r-2 ?o-3) and (see-arg-2 ?r-2 ?o-2) are the predications for the first and second arguments of the entity represented by ?r-2, while ?o-3 and ?o-2 are the respective variables for the arguments. Finally, variable bindings are used to show various roles of the same entity, e.g., in figure 6.3, ?o-2 represents both the variable introduced by a predicate ’ball’ and the variable introduced by the second argument of a predicate ’see’, which means that the same entity is performing these two roles.

A meaning exhibits *semantic recursion* when the semantic type of one of the arguments of a predicate is the same as the semantic type of the predicate itself. In figure 6.3, ’cogn-action’ is the semantic type of ’?r-3’ and also the semantic type of ’?r-1’, which is the second argument of ’?r-1’. The same occurs between ’?r-1’ and ’?r-2’, therefore this meaning exhibits semantic recursion two times.

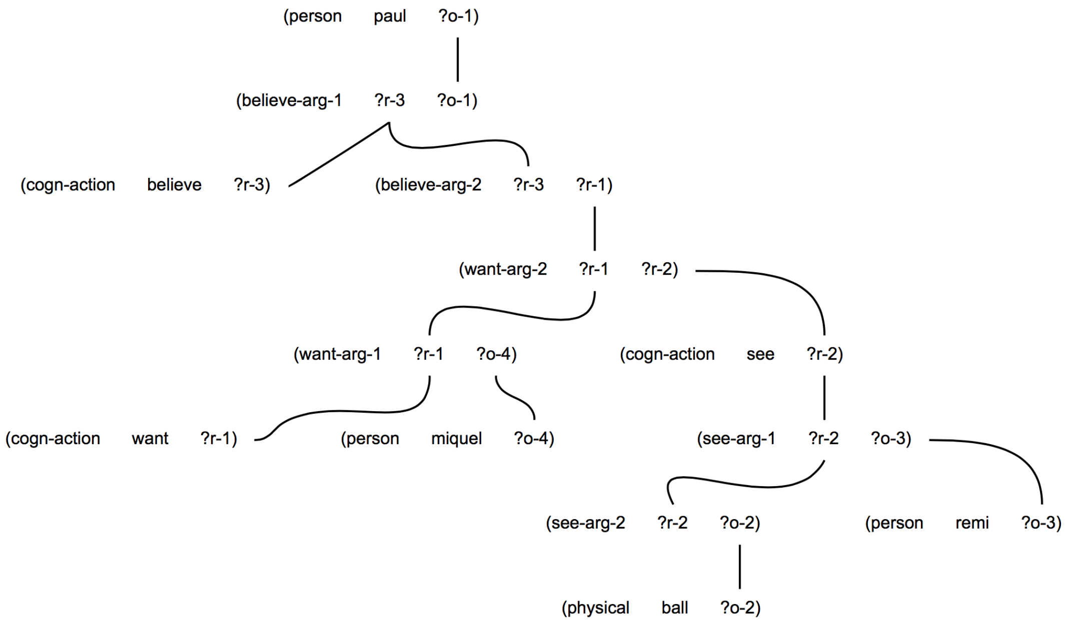


Figure 6.3: Meaning representation of the meaning of the utterance: *Paul believes Miquel wants Remi sees ball*. This meaning exhibits semantic recursion given that the predicate (cogn-action believe ?r-3) has as an argument the predicate (cogn-action want ?r-1) and both of them have the same semantic type.

Therefore, I distinguish between three kinds of recursion: *syntactic recursion* (syn), *processing recursion* (pr) and *semantic recursion* (sem).

The table below represents how these recursions can be combined. $a_{ij} = +$ implies that Rec_i must occur together with Rec_j and $a_{ij} = +/-$ implies that Rec_i can occur both together with Rec_j and without Rec_j. Notice particularly that semantic recursion always implies syntactic recursion, but syntactic recursion not always implies semantic recursion.

j i	syn	pr	sem
syn	■	+/-	+/-
pr	+	■	+/-
sem	+	+ (>1) / -	■

Table 6.1: Dependencies between types of recursion. While semantic recursion implies syntactic recursion, the opposite is not always true.

6.2 Learning Operators for the Emergence of Recursive Phrase Structure

In this section I first describe the differences in the syntax game model for the emergence of higher-order phrase structure considered and then I describe the application of the same learning operators used in chapters 4 and 5 in this model.

6.2.1 Syntax Game for Higher-Order Phrase Structure

The differences in the language game with respect to the case of first-order phrase structure are:

1. The meanings that need to be expressed include relations in order to force the usage of higher-order phrase structure. Therefore, more complex meanings and structures are encountered. In this case, the utterance meanings are always represented as connected networks, and the situations as sets of such connected networks.

2. The language game script may have an extra step in this case, given that when no grammar is yet conventionalized the chance for a successful language game interaction is lower than for the previous game. When a language game interaction fails I introduce an extra step where the hearer is forced to find a linguistic structure which

represents the meaning and utterance of the interaction and he uses this one to acquire new grammar.

Recall from chapter 2 that the syntax game setup used to motivate the usage of higher-order phrase structure was a discrimination language game and that the feature *referent* was considered for phrasal units. These conditions make communicative failures to occur more frequently than in the description games for first-order phrase structure, where the online interpretation in parsing allowed to discard wrong hypotheses immediately. The reason is that the topic of the utterance is a single entity (discrimination game) and the referential choices on every construction application effects the final result.

In order for the agents not to lose the opportunity to align their grammars in any interactions, when communication fails, the hearer agent tries to emulate what the speaker produced in a process called reentrance [90]. The reentrance process is the process in which the hearer produces himself the same utterance based on the meaning deduced from the comprehension process. Finally, in case of communicative failure, the solution of reentrance is the one used to learn or extend constructions, as well as to find the constructions that should be punished as competitors for those that started off a wrong branch while producing.

3. There are some limitations in the identification of the topic of the game. Conceptualization is restricted to the topics that are identifiable by means of phrase structure, this means that entities which are argument of a relation and are not the referent of the resulting phrasal unit cannot be the topic of an interaction. Considering these topics would require other syntactic mechanisms, e.g. anaphoras or morphology.

   Notice that hierarchy in language is introduced by unary and binary relations. Each time a relation is analysed, one of its arguments or the relation itself is chosen as the referent of the whole phrase.

Topic selection is limited in the cases where the expressive power of language is reduced only to phrase structure. For example, in figure 6.4, only *?o-1*, *?r-3* and *?r-1* could be the topic of the meaning network, while expressing the other entities requires other linguistic features, e.g. anaphoras.

For instance, a language structure such that *?o-4* is the topic cannot be built as a tree, because the relation introduced by *?want* is the topmost relation and it is not the topic, and on the other side is the argument of another relation, *?r-3*.

4. And finally, the lexicon and the phrasal constructions that the learning operators explore are the ones that support higher-order phrase structure (see section 2.3). In this case, the lexical system contains two kinds of words, words introducing one variable, and words introducing 2 or 3 variables. And a phrasal construction is recursive when the phrasal category that it adds to the new unit equals the phrasal category of one of its constituents.

Moreover, phrase structure grammar as modelled in this chapter doesn't assure that full alignment derives from phrasal alignment. Phrasal alignment is defined as the consistent order across agents on particular phrasal encodings, but the order of application may change the final word order of a sentence as it is shown in the examples below (recall that the subindices refer to the argument bindings and the + sign to the referential choice):

- $(+((+\mathit{table}_2)\ (\mathit{paul}_1)\ (\mathit{wants}))_1)\ (\mathit{sits})\ (\mathit{miquel})$ and $(+((+\mathit{table}_1)\ (\mathit{sits})\ (\mathit{miquel}))_2)\ (\mathit{paul}_1)\ (\mathit{wants})$
  The order of application effects the final utterance.

- $+((+\mathit{table}_1)\ (\mathit{big}))_1)\ (\mathit{red}))_1)$ and $+(+(\mathit{table})_1\ (\mathit{red})_1)\ (\mathit{big})$
  The order of application effects the final utterance.

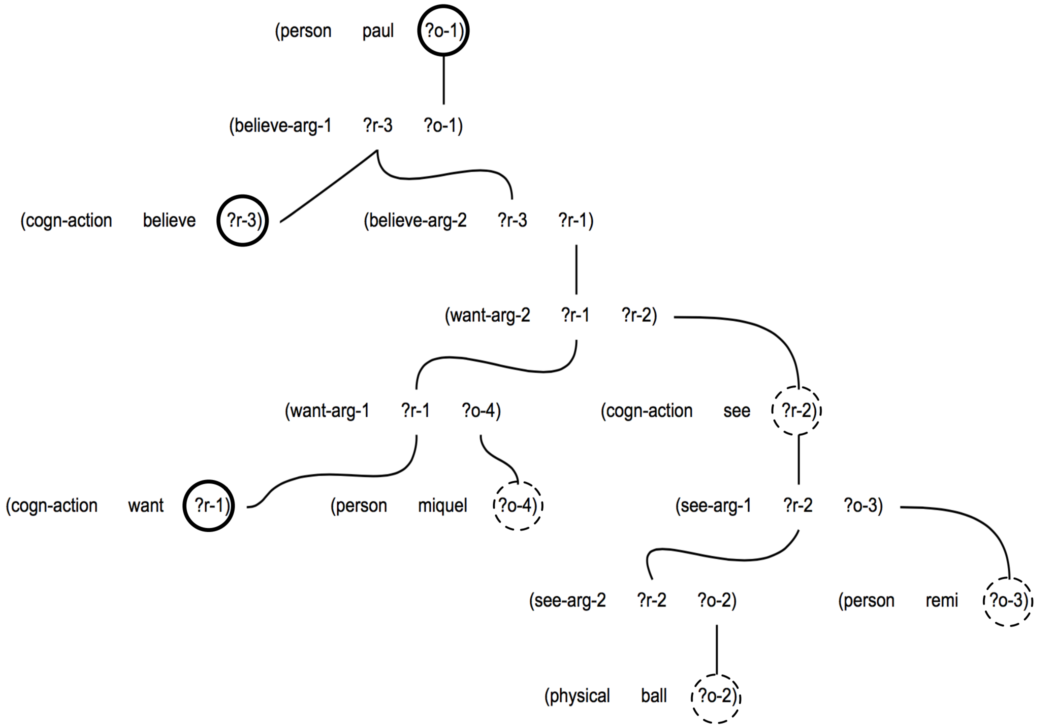


Figure 6.4: Topic selection is limited in the cases where the expressive power of language is reduced only to phrase structure. Solid circle lines highlight the entities that can be the topic of this network by means of phrase structure grammars, and dashed circle lines highlight those entities that can't be the topic of this network.

- *(red) +(+table*$_1$ *(big))*$_1$ and *(red) +(+(table)*$_1$*) (big))*$_1$
  The order of application doesn't effect the final utterance.

While for some cases phrasal alignment suffices to get full alignment, that is not the general case. The reason for this is that the proposed phrase structure grammars apply only locally and don't have global properties. Full alignment can only be reached using global measures, for example, by refining dependencies and minimizing dependency crossing, as in [45]. Similarly than in the previous model, I introduce a bias in prioritising phrasal constructions involving more constituents (3-constituent phrases over 2-constituent phrases, but this phenomenon still occurs when two phrasal construction involving the same number of constituent lead to two different utterances depending on the order of application.
Variation and selection are again implemented by means of learning operators. Variation is introduced by the application of invention operators. Invention and adoption operators are essentially the same ones than in the case of first-order phrase structure, with a natural extension in the learning due to syntactic coercion, which this time can possibly create a new pattern as well. Alignment strategies are as well the same than in the case of first-order phrase structure.

6.2.2 Contextual Inference Operator

When no partially matching constructions could be found, an operator for contextual inference can apply again. However in this case, it is not enough to apply the operator, that indeed leads to a correct meaning in terms of variable bindings, because the referent of the phrasal unit that it creates is chosen randomly, and therefore not always the right one is selected straight away, or ever in that interaction.
Again the consolidation for new phrasal constructions works the same, when no lexical category is in there a new one is created and added to the corresponding lexical constructions as well; and when there is more than one, a category is selected randomly. In this case, this may occur only once per phrasal construction, given that the rest of constituents are

phrases. For phrases it occurs either that they have one category or that they have no categories at all, if they don't have any category, the new construction cannot be created, and if they do, that syntactic category is taken as a matching category for the new construction.

The only difference observable in the performance is that it is possible to build new phrasal constructions only when the constituents that are phrases (and not words) have already been built by grammatical constructions, and hence, they have a syntactic category that the new pattern can use. Figure 6.5 shows an example per each of the cases in which the operator can apply.

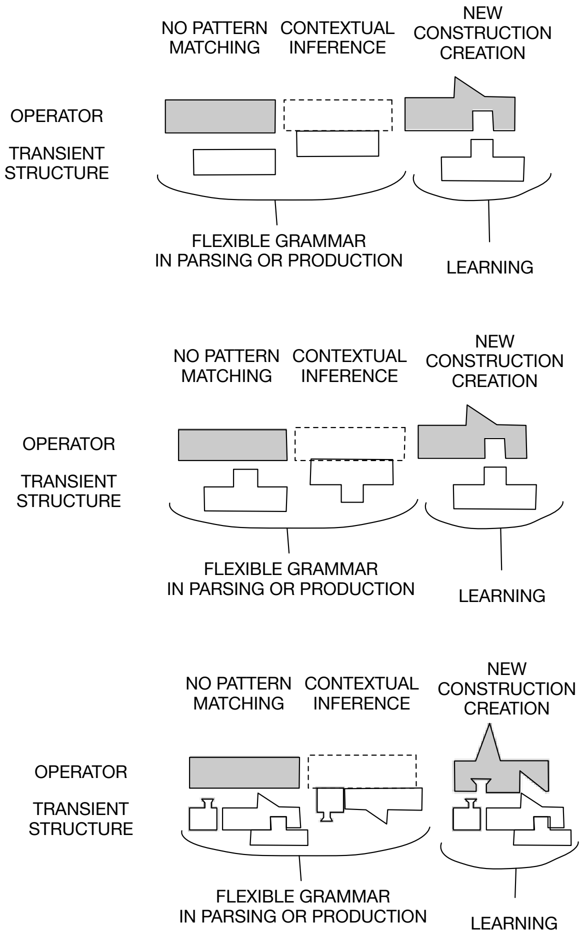


Figure 6.5: Top: The operator applies to build a phrasal construction for a single noun-phrase and it also adds a lexical category to the corresponding lexical construction. Middle: The operator applies to build a phrasal construction for a single noun-phrase but in this cases it uses the lexical category that the corresponding lexical construction introduced. Bottom: The operator applies to build a phrasal construction and it reuses the phrasal categories added by the corresponding phrases below whereas it creates a new phrasal category for its own.

Notice that in this case the new phrasal constructions encodes as well what is the local referent of a phrase. At a first sight, this might look as an extra functionality but in fact it is a derivation of the functionality to choose a topic in the meaning description, so it is a functionality of a discrimination game, as opposed to the description game used in chapters 4 and 5. The feature *referent* emerges as a result of the preference for minimal phrases and the effect of the communicative action.

6.2.3 Syntactic Coercion Operator

The syntactic coercion of lexical constructions works in the same way as in the previous experiments for first-order phrase structure. However, in this case the operator has to solve also the situation in which a phrasal construction partially matches to a sequence of constituents and the constituent that is not fully matching is a phrasal unit, instead of a lexical unit. In this case, when that application leads to the solution a new phrasal construction for the syntactic category of that phrasal unit is created. Figures 6.6 and 6.7 illustrate examples of both cases respectively.

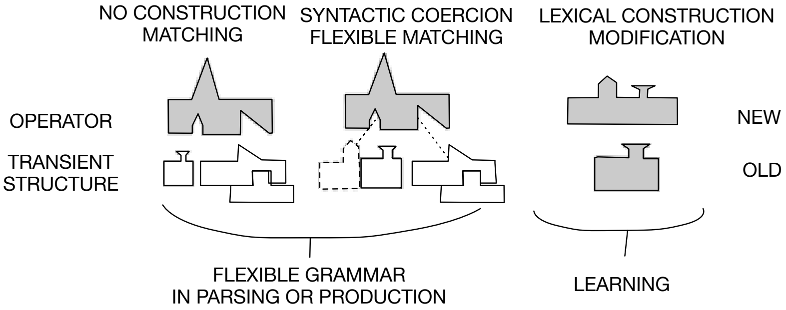


Figure 6.6: Syntactic coercion as in the case of first-order phrase structure.

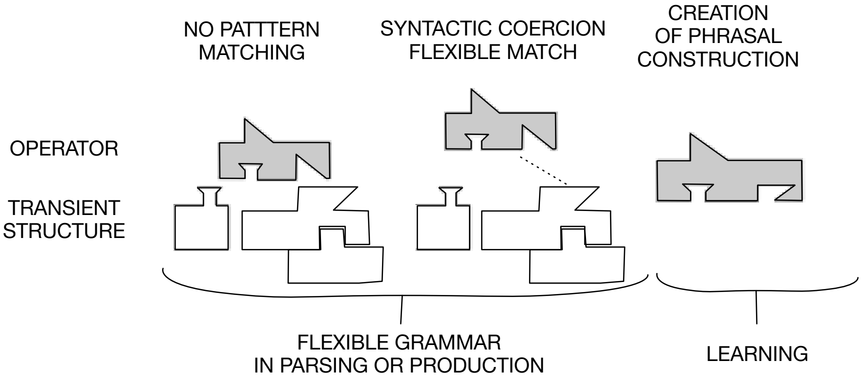


Figure 6.7: Syntactic coercion also applies for phrasal constructions this time, and therefore it may create a new pattern.

This strategy is essentially the same as the one used to make lexical constructions more flexible, but now the flexibility has to be stored in the form of phrasal constructions given that phrasal constructions matched and assign single categories to units. The efficiency of learning in this case is highly measured by the pattern efficiency.
Therefore, lexical and phrasal categories are never mixed.

In syntactic coercion for phrasal categories the learning could be done in two ways: either a variant of the construction which built the unit on the top is created or a variant of the construction which built the unit below is created. In this chapter I consider the first case for all cases except when the bottom unit is a single phrase.

6.2.4 Ordering Variation Operator

Again, this operator is similar than in the previous chapter, although now it includes variants assigning the referent to other constituents as well, and it is used in production.

6.2.5 Alignment Operator

In this setup, the choice of phrasal referents increases the search space of language interpretation and therefore the computational cost of language processing tends to be greater, specially when there is no grammar yet. This observation is another justification for considering an alignment operator as a selective force favoring minimal computational cost in language processing, because in this case trying to define offline competitor grammatical constructions is even less natural than in the case of first-order phrase structure.
Notice that in chapter 5 I introduced the notion of form competitors and meaning competitors, but only the latter were relevant. In the present case, form competitors are also present and therefore I have to take into account as well the internal meaning structure added by the constructions, and finding meaning competitors would require even a more complicated procedural than before if it was done offline the search process. For example, *(syn-cat-1*$_1$ *syn-cat-2*$_2$ *lex-cat-1)* and *(syn-cat-1*$_2$ *syn-cat-2*$_1$ *lex-cat-1)* are form competitors regardless of the referential choice, and *(+syn-cat-1*$_1$ *lex-cat-1 syn-cat-2*$_2$*)* and *(+syn-cat-1*$_2$ *syn-cat-2*$_1$ *lex-cat-1)* are meaning competitors but the referential choice matters.

Therefore, I use again the cost selection alignment operator that before proved to be successful.

6.3 Baseline Experiment

Given that fundamentally the same learning operators tested in this chapter have been tested in chapter 4 for acquisition experiments, I don't treat again acquisition experiments here and move immediately to formation experiments.

In this section I overview the characteristics and conditions of the baseline experiment and discuss its results.

6.3.1 Experiment: No Invention/Adoption Operators

The main interest of this chapter lies on the application of the operators presented in chapters 4 and 5 to higher-order phrase structure. I consider one baseline experiment where both the invention and adoption operators are inhibited.

Population size

2 agents. In this setup it makes no differences how many agents interact, as their grammars don't change over time.

Utterance length

3, 4, 5 and 6 words.

Ontology size

Semantic Types A	Properties P
a-1	p-1-1 p-1-2
a-2	p-2-1 p-2-2
a-3	p-3-1 p-3-2
Unary relations Q	Values Q'
b-1	q-1-1 q-1-2
Binary relations R	Values R'
r-1	r-1-1

The first argument of the unary relations b-1 is compatible with the semantic type (attribute) a-1, while the first argument of the binary relation is compatible with r-1, a-1 and b-1, and the second argument is compatible with a-2 and a-3.

Situation size

Two connected networks each of them as the same size as the utterance length.

Generation of Situations

Situations consist of two connected meaning networks as the same size of the utterance length parameter. Therefore, per each utterance length, only few configurations are possible. I describe each of them below accounting for the number of single phrases, the number of unary relations, the number of binary relations, and the network hierarchy (or utterance hierarchy), which I define as the number of hierarchical levels in the corresponding linguistic structure.

- 3 words: 1 binary relation and 2 single noun-phrases. Network hierarchy = 1
- 4 words: 1 binary relation, 1 unary relation and 2 single noun-phrases. Network hierarchy = 2
- 5 words: 1 binary relation, 2 unary relations and 2 single noun-phrases, (with probability 0.75, derived from the situations generator) or 2 binary relations, no unary relations, and 3 single noun phrases (with probability 0.25, derived from the situations generator). Network hierarchy = 2 and 2, respectively. E.g. (”joile” (”naupo” ”gaitxo”) (”nonyas” ”nyeina”)) and ((”dotxap” ”xalo” ”moge”) ”nopa” ”vautxe”).
- 6 words: 2 binary relations, 1 unary relations and 3 single noun-phrases. Network hierarchy = 3.

Moreover, the values of attributes (i.e. properties of semantic types) and values of relations are introduced sequentially as:

- Interaction 1: the following properties and relation values are in use p-1-1, p-2-1, q-1-1, q-1-2 and r-1-1
- Interaction 25: introduction of the attribute value p-1-2 to be selected to take part in a situation.
- Interaction 51: introduction of the attribute value p-2-2 to be selected to take part in a situation.
- Interaction 101: introduction of the attribute value p-3-1 to be selected to take part in a situation.
- Interaction 151: introduction of the attribute value p-3-2 to be selected to take part in a situation.

6.3.2 Results

In the baseline experimental setup the number of words in the utterance increases by 1 every 1000 interactions. In this case, starting from 3 (minimal utterance including one binary relation) and ending at 6. Figure 6.8 shows the average results of 5 independent runs with a gray ribbon representing the sample standard deviation.

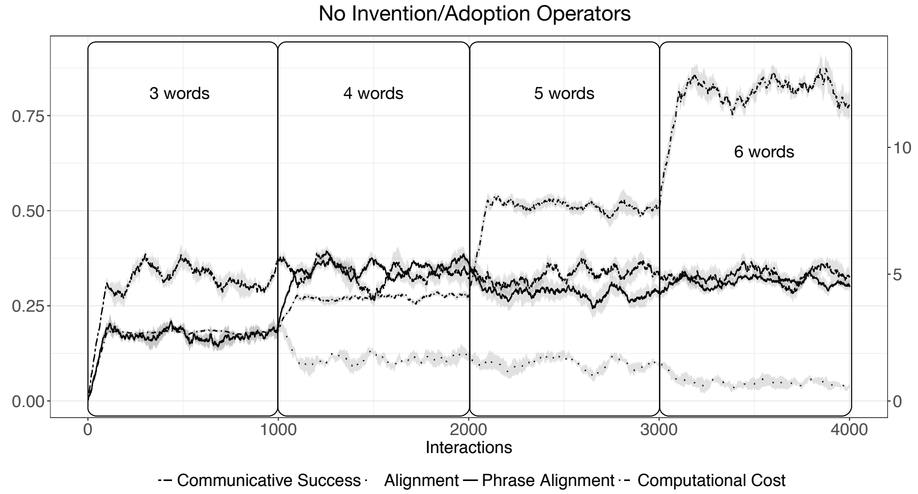


Figure 6.8: The computational cost (right y-axis) increases as the utterance length increases, and communicative success remains roughly the same. Phrase alignment in this case includes phrases of length 2 and 3 for all the utterance lengths.

The interpretation of the results follows:

Communicative Success (CS)

CS is around 1/3, for all words, as discussed earlier in most of the cases communication failure comes from wrong referent identification of the top relation, and not from the linguistic structure itself.
CS slightly increases with the number of words due the increase in frequency of cases where the top referent choice is given by a unary relation instead of a binary relation where the chances to make the right choice are 1/2 instead of 1/3. However, this cannot be appreciated in the graph.

Phrasal Alignment (PA) and Alignment (A)

PA keeps stable and it is slightly higher for the periods involving $\{4, 6\}$-word utterances due to the higher contribution of 2-word phrases to the measure in those cases. On the other side, A decreases as long as the utterance length increases due to the increase of possibilities to use different orders in more phrases.

Computational Cost (CC)

Concerning CC notice that in the model for first-order phrase structure the only contribution to CC was due to the wrong identification of phrases in the utterance. Instead, in this case the hearer has to make more choices in language comprehension that may increase the computational cost. Wrong choices inconsistent with the situation still come from wrong coreferential assignment, in this case, argument bindings. But additionally, wrong referential choices may also increase the CC due to the fact that there is no blocking mechanism for these branches in the search space and a depth-first algorithm is used to explore it. The branches that started off after a wrong referential choice are blocked only when it is found that they cannot lead to a solution because not all predicates can be connected. Consequently, figure 6.8 shows an increasing computational cost from 3 to 6 words.

6.4 Formation Experiment

In the next series of experiments I explore the performance of the hypothesized operators to allow a population of agents to self-organise a language system exhibiting higher-order phrase structure and eventually exploiting recursion on it.

6.4.1 Main Experiment: Self-Organisation by SBP

In this experiment I test whether agents are capable to self-organise a language system requiring hierarchical phrases and whether recursion is exploited or not by the system.

Population size

2 agents and 5 agents.

Utterance length

3, 4, 5, 6, 7, and 8 words, in a sequence of 500 interactions each. A setup with an increasing number of words is considered in order to visualize better the effect of recursive constructions.

Ontology size

Semantic Types A	Properties P
a-1	p-1-1 p-1-2
a-2	p-2-1 p-2-2
a-3	p-3-1 p-3-2

Unary relations Q	Values Q'
b-1	q-1-1 q-1-2

Binary relations R	Values R'
r-1	r-1-1

Similarly than in the baseline experiment, the first argument of the unary relations b-1 is compatible with the semantic type (attribute) a-1, while the first argument of the binary relation is compatible to r-1, a-1 and b-1, and the second argument is compatible to a-2 and a-3.

Situation size

Two connected networks each of them as the same size as the utterance length.

Generation of situations

The generation of situation also follows the same schema than the baseline experiment, namely, first it randomly selects a possible network configuration for the corresponding number of words, after that it randomly selects semantic types and relations per each of the position compatible with each other, which are finally instantiated by a randomly selected value. And the same process is repeated two times given that the situation size includes two network of the size of the utterance.
In this experimental setup agents use also utterances of 7 and 8 words whose possible meaning configurations are listed below given that they did not apply in the baseline experiment and therefore were not mentioned.

- 7 words: 2 binary relations, 2 unary relations and 3 single noun-phrases, or 3 binary relations, no unary relations, and 4 single noun phrases. Network hierarchy = 3 and 3, respectively. E.g. (”sailo” (”guamequa” ”togogua”) (”nyenegua” (”mohe” ”togogua”) ”bemare”)) and (”henyavo” ((”geto” ”nyenegua” ”seubo”) ”bape” ”geto”) ”gefex”)
- 8 words: 2 binary relations, 3 unary relations and 3 single noun-phrases, or 3 binary relations, 1 unary relations, and 4 single noun phrases. Network hierarchy = 3 and 4, respectively. E.g. (”lozo” (”nyoude” ”reuce”) (”feube” (”roxoba” ”tomax”) (”none” ”zebe”))) and (”fonyaf” ”nyejope” (”lozo” (”tareny” ”volaje” (”heibo” ”tomax”)) ”panelo”))

E.g. 7 words ⇒ 3 binary relations, 0 unary relations and 4 single noun-phrases ⇒ r-1, r-1, r-1, r-1, a-1, a-2, r-1, b-1, a-1, a-2 ⇒ *((R-1 R-1-1*

O-18214) (R-1-1-ARG-1 O-18214 O-18215) (R-1-1-ARG-2 O-18214 O-18220) (R-1 R-1-1 O-18215) (R-1-1-ARG-1 O-18215 O-18216) (R-1-1-ARG-2 O-18215 O-18219) (R-1 R-1-1 O-18216) (R-1-1-ARG-1 O-18216 O-18217) (R-1-1-ARG-2 O-18216 O-18218) (A-1 P-1-1 O-18217) (A-2 P-2-1 O-18218) (A-2 P-2-1 O-18219) (A-2 P-2-1 O-18220)) which is represented in figure 6.9.

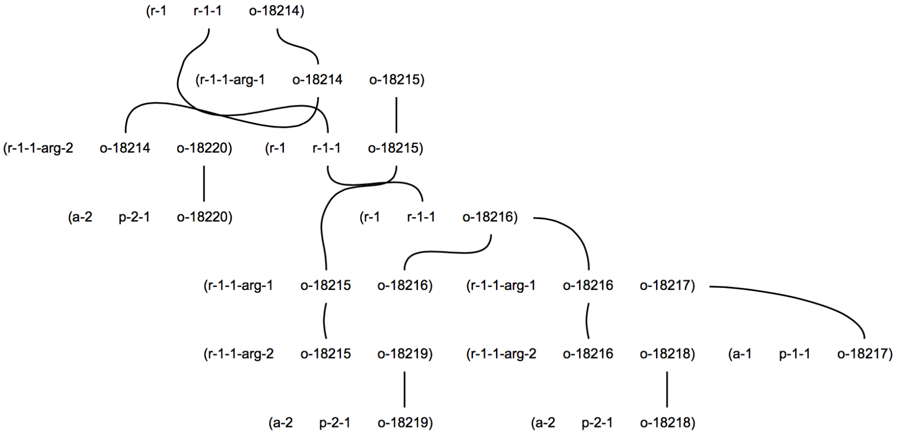


Figure 6.9: A network of a situation which includes 7 predications, so whose meaning requires a 7-word utterance to be expressed.

Figure 6.10 shows which which number of binary relations are required for every number of words.

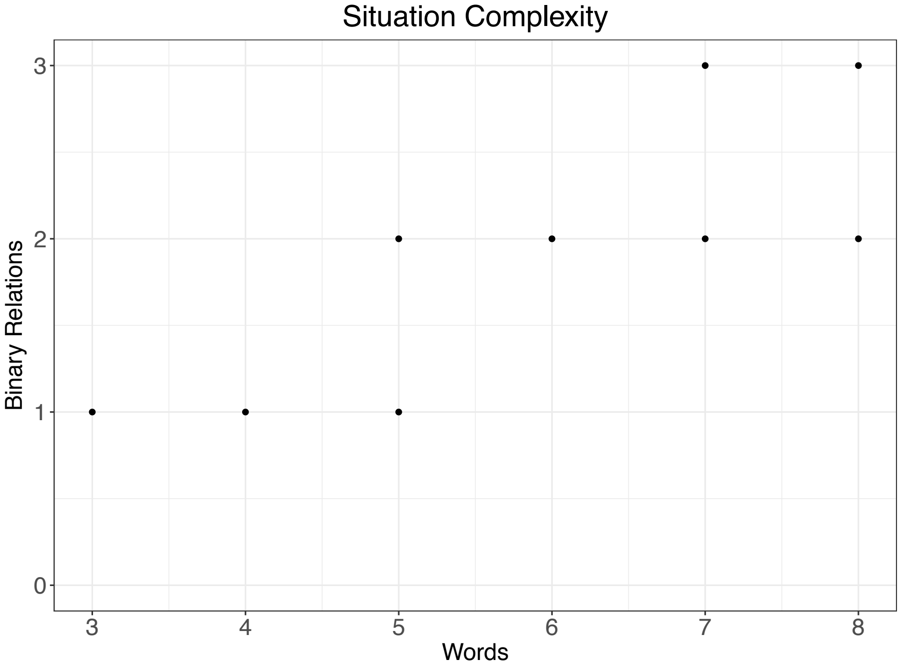


Figure 6.10: Utterance length increases incrementally in this setup from 3 words to 8 words, adding one word every 500 interactions. The graph shows the number of binary relations possible at each case. From 5 words on it is possible to use recursive phrasal constructions because unary or binary relations are required.

6.4.2 Results

In this section I present and discuss the results of the computational experiments to test the model for the emergence of recursive phrase structure proposed above. I first present the results for the experiments involving 2 artificial agents, and later, the results for the experiment involving 5 artificial agents.

2 agents

The following results are taken from 5 independent experimental runs of 3000 interactions each. Every 500 interactions the utterance length increases by one word. As it is shown in figure 6.11. The alignment operator using cost selection had a positive impact again in this case, where meaning had a more complex structure.

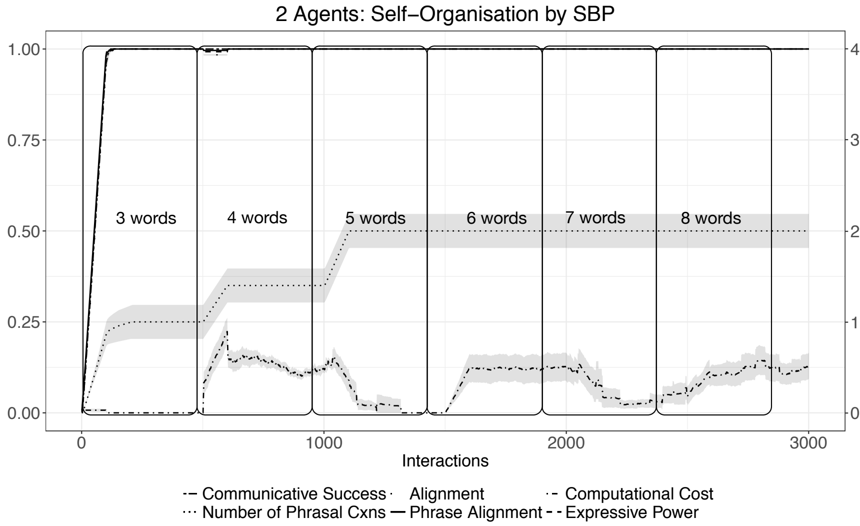


Figure 6.11: Full alignment, phrase alignment, communicative success and expressive power are reached immediately. While the number of phrasal constructions (right y-axis) has some variance, and the computational cost remains slightly higher than 0 for 6 and 8.

Communicative Success (CS) and Expressive Power (EP)

CS and EP get immediately to its maximum.

Phrasal Alignment (PA) and Alignment (A)

PA and A also get immediately to its maximum. Interestingly, A can also get to one because all the phrasal constructions which introduce constituent orders are variants of the same original constructions; one for unary relations from which the two variants to discriminate referential choices derive, and one for binary relations, from which as well the three variants to discriminate referential choices derive. This is not the case of the experimental runs involving 5 agents as it is shown in figure 6.14.

Computational Cost (CC)

CC keeps very low except for the periods of $\{4, 6, 8\}$-word utterances where it gets close to 0.12. This is due to the high presence of meaning networks where both unary and binary relations occur. Given that both semantic types involved in unary relations (*b-1 and a-1*) can be arguments of binary relations, the probability to follow a wrong hypothesis in a branch started off by choosing the uncorrected referent is higher, and so it is the CC.

Number of phrasal constructions

Each agent acquires at least 2 phrases with two constituents and 3 phrases with three constituents. However, if recursion is not exploited early enough, they can learn up to 12 phrases with 3 constituents. In the graph, the single-noun phrases are not counted.

Similarly than looking at the number of phrasal constructions, recursion can be visualized by looking at the individual score evolution, figure 6.12.

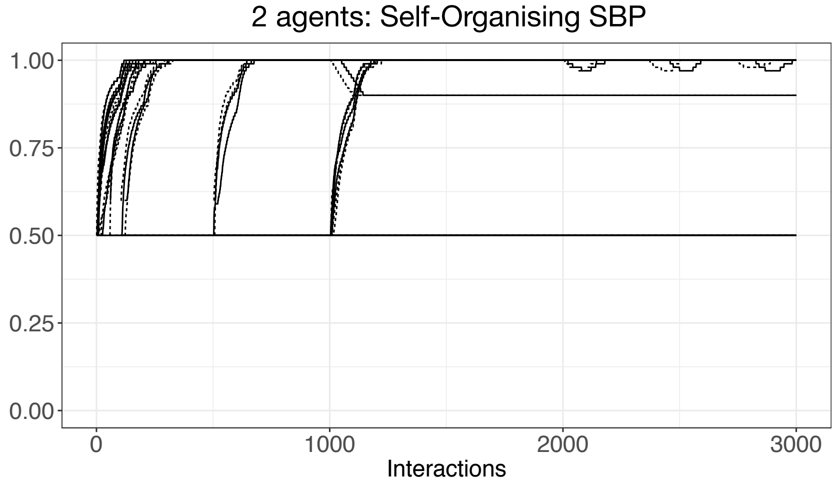


Figure 6.12: Evolution of individual scores. Each line type represents an agent. During the interactions of utterance of 5 to 8 words, agents don't acquire new grammar. This proves that they are capable to self-organise phrase structure grammars that exploits recursion.

Resulting Grammars

Figure shows the evolution of an agent grammar for one experimental run. Grammars are represented as graphs in two ways:

- in the left column nodes represent from left to right: lexical constructions, lexical categories, phrasal constructions and phrasal categories; and similarly than in previous chapters edges represent respectively lexical constructions introducing lexical categories, lexical categories matching on phrasal constructions, and now the addition of outgoing arrows from phrasal categories to phrasal constructions representing categories matching and ingoing arrows to phrasal categories from phrasal constructions representing the phrasal category of the phrase built by the construction.
- In the right column, in order to visualize better the appearance of recursion, nodes represent lexical categories and phrasal categories, and outgoing arrows represent the relation of a category matching a phrasal construction whose resulting phrasal category is the ingoing node.

Notice that green nodes (third position from the left in the first column) represent phrasal constructions, and single noun phrase are uniquely combined with one lexical construction when the operators proposed are in use. It is more relevant in this case to look at the graph on the second column to see that indeed several lexical units further match to phrasal constructions by means of the same phrasal category.
In other words, processing recursion is forced and as a consequence recursion exploited. It is worth saying that no specific assumption about the existence of a syntactic head had to be made in order to observe this outcome.

5 Agents

In the next series of experiments five agents endowed with all the operators discussed were considered. Again, experimental runs of 3000 interactions

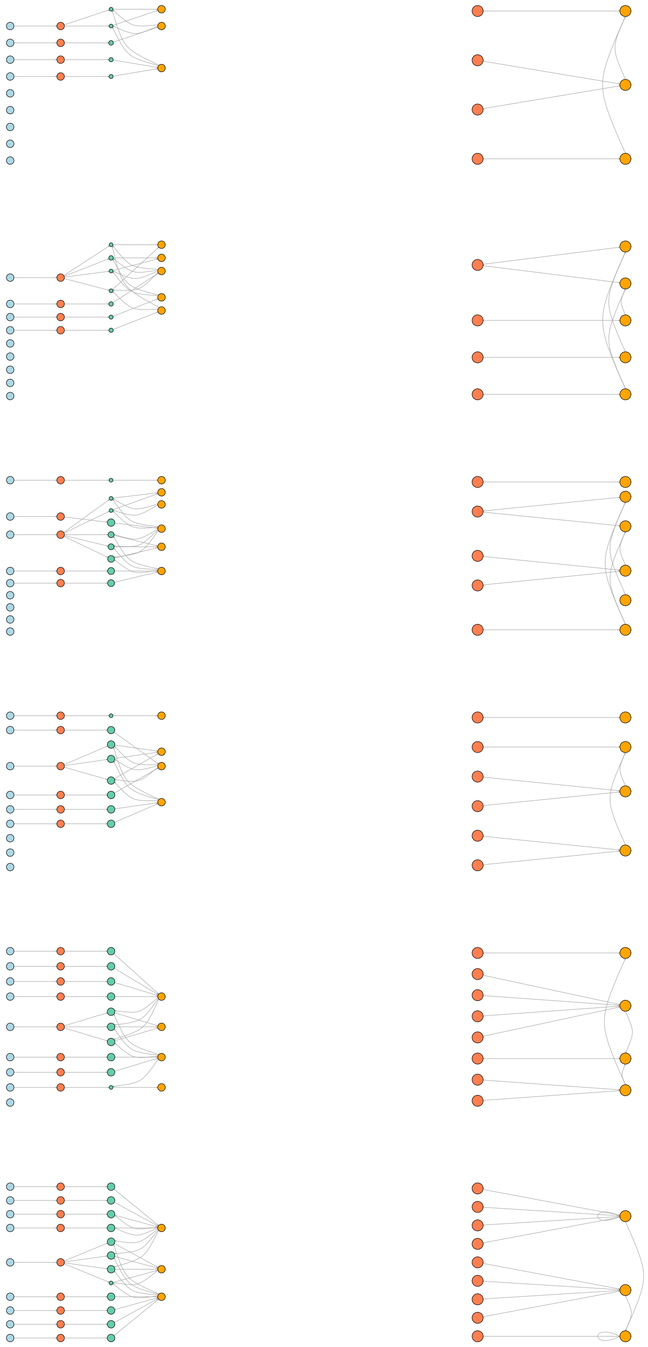


Figure 6.13: Grammar of the same agent in interactions 5, 9, 53, 104, 503, and 1005.

and every 500 interactions the utterance length increases by one word. Figure 6.14 shows how phrase alignment gets to its maximum and agents acquire phrasal constructions only until facing 5-word utterances, and from there on they already have enough grammar to successfully process the coming utterances, proving the emergence of recursive phrase structure.

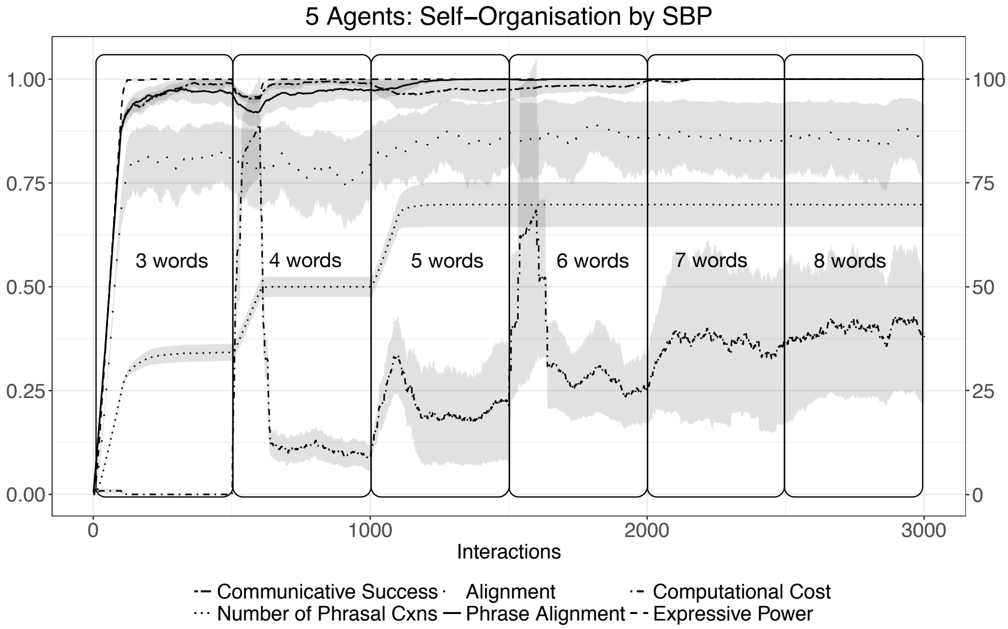


Figure 6.14: Agents reach full communicative success, phrase alignment and expressive power, although it takes longer than in the setup with only two agent. Again the variance in the size of the grammar is big and the computational cost stays low but not 0.

Communicative Success (CS) and Expressive Power (EP)

CS eventually reaches its maximum after unstable moments during learning periods and few failures during utterances of 5 and 6 words. EP reaches its maximum immediately.

Phrasal Alignment (PA) and Alignment (A)

PA reaches its maximum around half of the period where 5-word utterances are considered, and A doesn't reach its maximum due again to the fact that the order of application of constructions might lead to different utterances while having full phrase alignment.

Computational Cost (CC)

The CC cost is much greater than it was for the experiments involving two agents. Particularly is much greater in learning periods, around the first interactions of 4, 5 and 6 word utterances, but also once grammar is stabilized it keeps slightly greater than 0.3. For learning periods the reason is that more competing variants are generated as it is shown in the resulting grammars section. At the end of the competing period the increase of CC with respect to the experiment involving two agents can be explained by two factors:

- agents don't erase the constructions that are not successfully apply (i.e. they don't forget)
- agents use a depth-first search in language processing which let them follow incorrect hypotheses due to wrong referential choices.

However, it is still much more smaller than in the baseline experiment confirming the positive effect of the emergence of phrase structure to reduce the computational cost of language interpretation.

Number of phrasal constructions

Agents acquire many more constructions than in the case were only 2 agents are involved due to the generation of grammar variants originated from different interactions. Again, the evolution of individual scores helps to visualize the evolution of these grammars along the experiment. See figure 6.15.

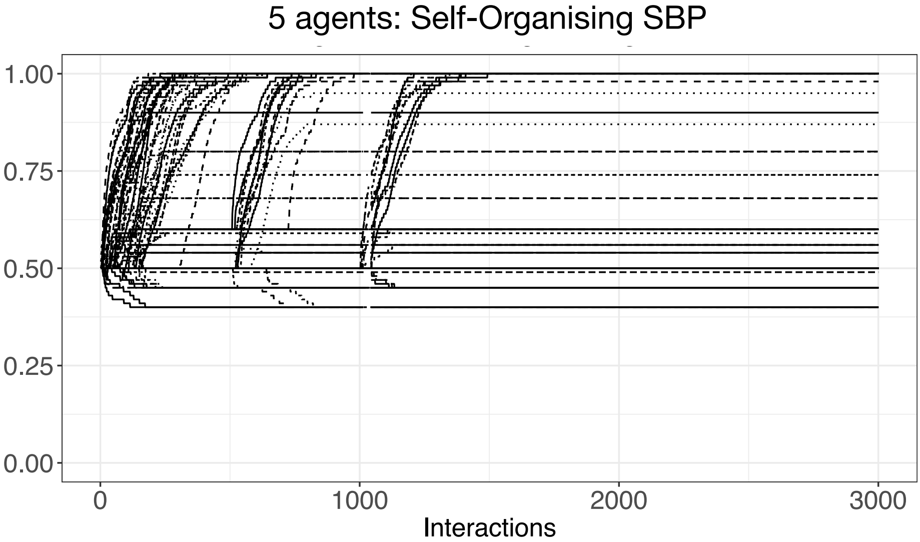


Figure 6.15: Evolution of individual scores. Again it is clear that agent find grammars exploiting recursion given that they don't need to acquire new grammar after having acquired sufficient grammar of 5-word utterances, and being able to successfully process and use utterances of up to 8 words (in the experiments), and ad infinitum (theoretically).

Some of the constructions are punished in this case, although not during long periods. Notice than any phrasal construction whose whose score remains static and not 1 is not being used for the final grammar. The final stabilization of grammar and end of grammar variant competition occurs around interaction 1300, which coincides with the moment in which phrase alignment also gets maximum.

Resulting Grammars

In this section I take a closer look to the resulting grammars. On one side, in figure 6.17 the evolution of an individual agent grammar is represented along the experiment.
On the other side, figure 6.16 helps visualizing the final configuration of phrasal categories added by phrasal constructions matching on syntactic categories (lexical of phrasal) and it shows that indeed categories 7 and 11 are used recursively in at least one phrasal construction.

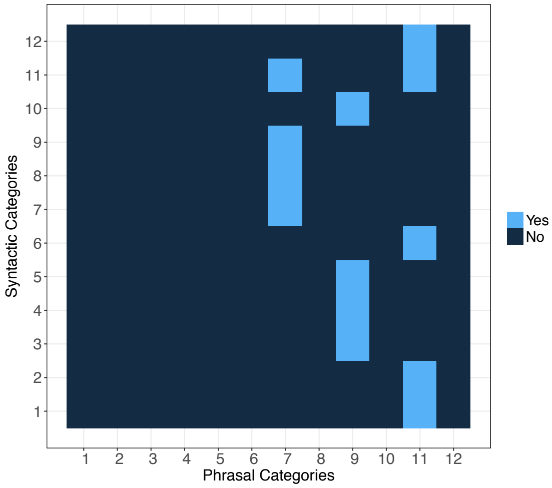


Figure 6.16: Recursion in syntactic categories is represented by the elements in the diagonal. Moreover, only 3 phrasal categories are used in the function of assigning a concrete argument or the relation itself as the referent of the new unit, inducing the idea of a syntactic head, given that syntactic properties (phrasal categories) depend on the constituent that carries the referent of the new phrase.

In this case the discussion on the distribution of lexical categories is less relevant, and what matters is the distribution of phrasal categories. Figures 6.18 and 6.19 represent again the distribution of phrasal categories as the

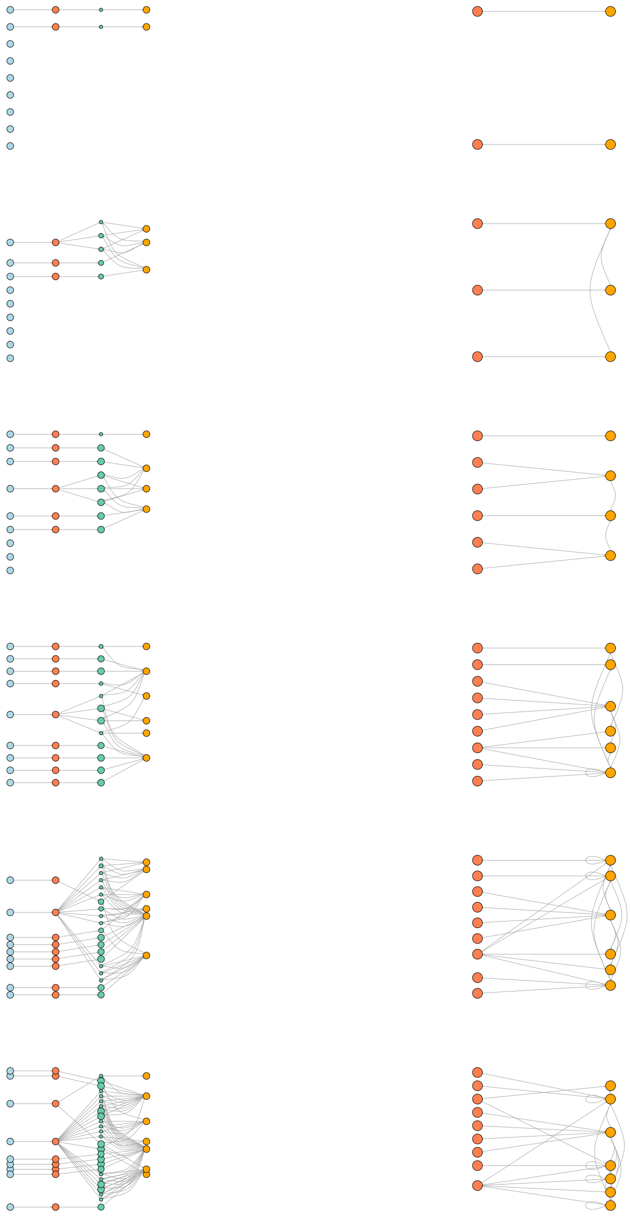


Figure 6.17: Grammar of the same agent in interactions 1,12, 103, 506, 521 and 1504. On the left column it can be appreciated how competition is much more active in this case than when only two agent interact. The size of the nodes representing phrasal constructions are according to their scores. On the right column in interaction 506 the first recursive phrasal construction appears as the loop arrow in on of the nodes representing phrasal categories shows.

categories introduced by a phrasal construction with respect to syntactic categories (lexical or phrasal) as the categories matching the corresponding constructions. The configuration of an agent grammar before and after convergence is shown.

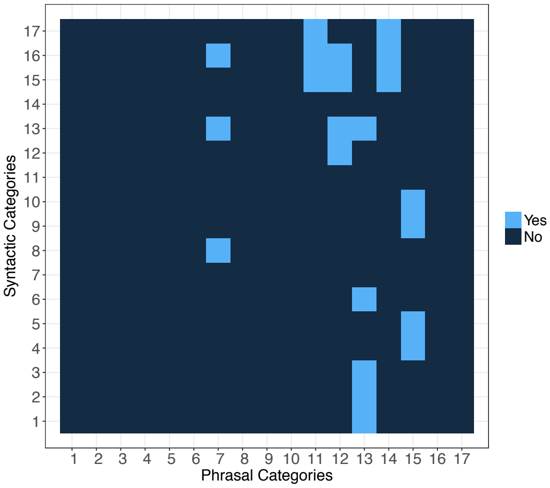


Figure 6.18: Before convergence six phrasal categories are added by phrasal constructions.

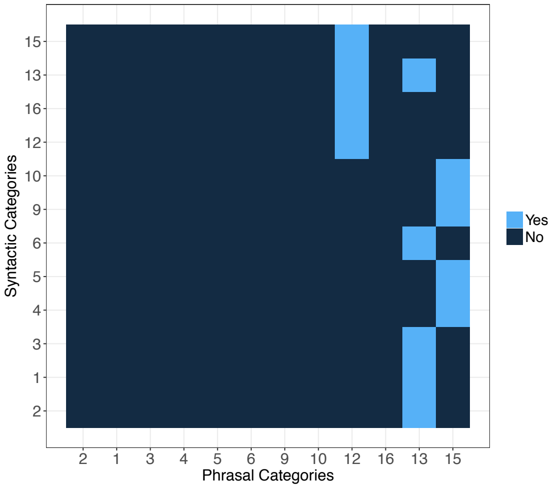


Figure 6.19: After convergence only three phrasal categories are added by phrasal constructions.

In order to illustrate the variation across grammars, figure 6.20 compares the grammars evolved by 4 agents in the same experimental run.

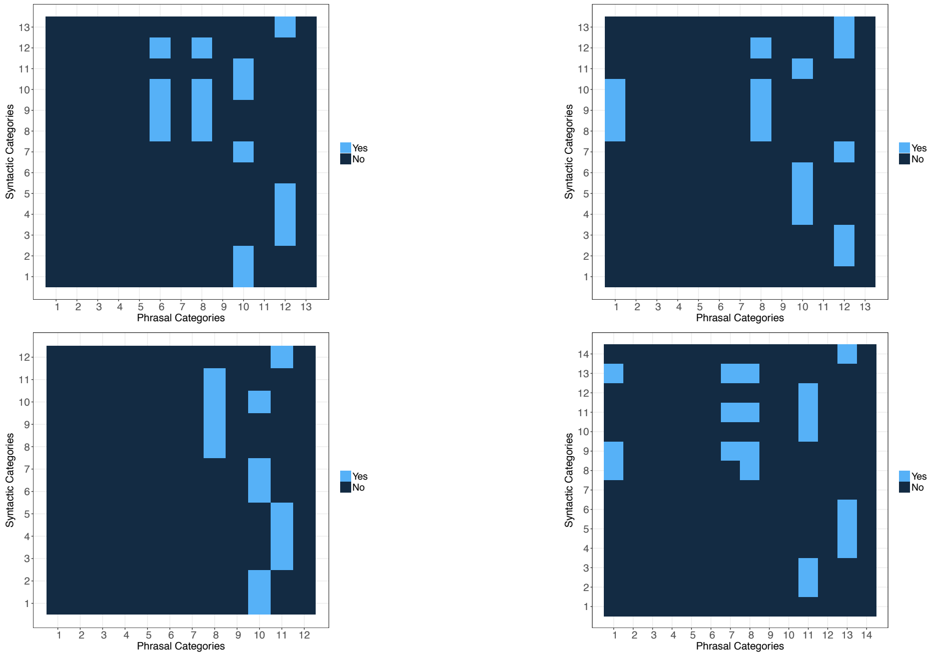


Figure 6.20: There is variation of grammars across agents, both for the number of phrasal constructions and categories, and the distribution of categories across lexical and phrasal constructions.

Learning Efficiency

Learning efficiency is worse than in the case of first-order phrase structure and a better quantification is required in order to understand deeper the role of the mechanisms of coercion proposed in the thesis.

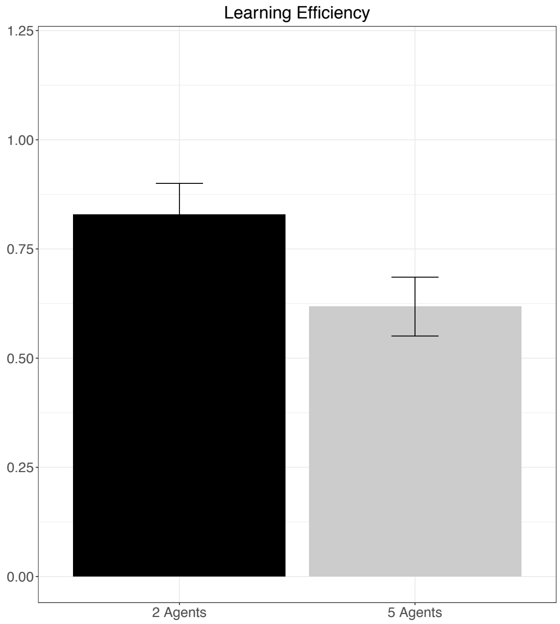


Figure 6.21: Learning efficiency of phrasal constructions. The ratio between the total number of phrasal constructions learned and the ones used when grammar agents converge.

FCG Constructions

Finally, some of the resulting phrasal construction using FCG notation are represented below:

$$\left[\begin{array}{l}\textbf{?np-unit}\\ \hline\hline \text{referent: ?rel}\\ \text{subunits: \{?word-rel, ?group-1\}}\\ \text{syn-cat: syn-cat-16}\\ \text{boundaries: [?word-rel, ?right-group-1]}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{?word-rel}\\ \hline\hline \text{args: [?rel, ?referent]}\\ \text{potential-syn-cat: \{syn-cat-91\}}\\ \hline \text{args: [?rel, ?referent]}\\ \text{potential-syn-cat: \{syn-cat-91\}}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{?group-1}\\ \hline\hline \text{referent: ?referent}\\ \text{syn-cat: syn-cat-16}\\ \text{boundaries: [?left-group-1, ?right-group-1]}\\ \hline \text{referent: ?referent}\\ \text{syn-cat: syn-cat-16}\\ \text{boundaries: [?left-group-1, ?right-group-1]}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{?word-rel}\\ \hline\hline \text{referent: ?rel}\end{array}\right] \leftarrow$$

$$\left[\begin{array}{l}\textbf{?np-unit}\\ \hline\hline \emptyset\\ \hline \text{\# form: \{meets(?word-rel, ?left-group-1)\}}\end{array}\right]$$

(syn-cat-91 syn-cat-16)

$$\left[\begin{array}{l}\textbf{?np-unit}\\ \hline\hline \text{referent: ?arg-1}\\ \text{subunits: \{?group-1, ?word-rel, ?group-2 \}}\\ \text{syn-cat: syn-cat-30}\\ \text{boundaries: [?left-group-1, ?right-group-2]}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{?word-rel}\\ \hline\hline \text{args: [?rel, ?arg-1, ?arg-2]}\\ \text{potential-syn-cat: \{syn-cat-19\}}\\ \hline \text{args: [?rel, ?arg-1, ?arg-2]}\\ \text{potential-syn-cat: syn-cat-19:}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{?group-1}\\ \hline\hline \text{referent: ?arg-1}\\ \text{syn-cat: syn-cat-30}\\ \text{boundaries:}\\ \quad\text{?left-group-1: ?right-group-1}\\ \hline \text{referent: ?arg-1}\\ \text{syn-cat: syn-cat-30}\\ \text{boundaries:}\\ \quad\text{?left-group-1: ?right-group-1}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{?word-rel}\\ \hline\hline \text{referent: ?arg-1}\end{array}\right] \leftarrow$$

$$\left[\begin{array}{l}\textbf{?np-unit}\\ \hline\hline \emptyset\\ \hline \text{\# form: \{meets(?right-group-1, ?word-rel),}\\ \qquad\text{meets(?word-rel, ?left-group-2)\}}\end{array}\right]$$

$$\left[\begin{array}{l}\textbf{?group-2}\\ \hline\hline \text{referent: ?arg-2}\\ \text{syn-cat: syn-cat-14}\\ \text{boundaries:}\\ \quad\text{?left-group-2: ?right-group-2}\\ \hline \text{referent: ?arg-2}\\ \text{syn-cat: syn-cat-14}\\ \text{boundaries:}\\ \quad\text{?left-group-2: ?right-group-2}\end{array}\right]$$

(syn-cat-30 syn-cat-19 syn-cat-14)

?np-unit
referent: ?arg-2
syn-cat: syn-cat-30
subunits: {?word-rel, ?group-1, ?group-2}
boundaries: [?word-rel, ?right-group-2]

?word-rel
args: [?rel, ?arg-1, ?arg-2]
potential-syn-cat: {syn-cat-19}
args: [?rel, ?arg-1, ?arg-2]
potential-syn-cat: {syn-cat-19}

?group-1
referent: ?arg-1
syn-cat: syn-cat-30
boundaries:
?left-group-1: ?right-group-1
referent: ?arg-1
syn-cat: syn-cat-30
boundaries:
?left-group-1: ?right-group-1

?word-rel
referent: ?arg-2
←

?np-unit
∅
form: {meets(?word-rel, ?left-group-1),
meets(?right-group-1, ?left-group-2)}

?group-2
referent: ?arg-2
syn-cat: syn-cat-14
boundaries:
?left-group-2: ?right-group-2
referent: ?arg-2
syn-cat: syn-cat-14
boundaries:
?left-group-2: ?right-group-2

(syn-cat-19 syn-cat-30 syn-cat-14)

?np-unit
referent: ?rel
syn-cat: syn-cat-30
subunits: {?word-rel, ?group-2, ?group-1}
boundaries: [?word-rel, ?right-group-1]

?word-rel
args: [?rel, ?arg-1, ?arg-2]
potential-syn-cat: {syn-cat-19}
args: [?rel, ?arg-1, ?arg-2]
potential-syn-cat: {syn-cat-19}

?group-1
referent: ?arg-1
syn-cat: syn-cat-30
boundaries: [?left-group-1, ?right-group-1]
referent: ?arg-1
syn-cat: syn-cat-30
boundaries: [?left-group-1, ?right-group-1]

?word-rel
referent: ?rel
←

?np-unit
∅
form: {meets(?word-rel, ?left-group-2),
meets(?right-group-2, ?left-group-1)}

?group-2
referent: ?arg-2
syn-cat: syn-cat-14
boundaries: [?left-group-2, ?right-group-2]
referent: ?arg-2
syn-cat: syn-cat-14
boundaries: [?left-group-2, ?right-group-2]

(syn-cat-19 syn-cat-14 syn-cat-30)

6.5 Discussion

Recursion in the form of recursive phrase structure can emerge culturally. The agent-based model studied in this chapter demonstrated how a learning strategy and its set of learning operators allow a population of agents to self-organise a language exhibiting recursive phrase structure.

The model demonstrated that grammar is of crucial importance for increasing the communicative success of more complex utterances and for damping cognitive effort, in particular, the computational cost of semantic interpretation, which is the main tenet of this thesis. It also showed that recursion does not require a big computational leap but that it can be handled using the same computational mechanisms and grammar building operators that are required for non-hierarchical nor recursive phrase structures in chapter 5 and a natural extension of the coercion operator. And selective pressures on the computational cost are again the main driving force for the emergence of the desired grammatical system exhibiting recursion.

Phrasal constructions in this chapter have an additional feature to represent the referent of a phrasal unit. This feature is justified as the result of using discrimination games as opposed to description games. The models for the emergence of recursion that used description games had to assume semantic recursion beforehand, and this is not a prerequisite for the model discussed in the present chapter. Furthermore, the pointing action is a much more realistic mechanisms for discrimination games than for description games, therefore, it is a very good finding the demonstration that discrimination games give better results.

Most linguistic theories assume predefined syntactic categories, including phrasal categories, and they rely on the concept of syntactic head in order to decide on the phrasal category of a phrase according to its constituents. For example, they may assign the same phrasal category to the whole phrase than to the syntactic head, e.g. $NP \rightarrow adj\ NP$. The grammars resulting from the model suggest that other approaches to the study of the concept of syntactic head should be considered, as it gives a communicative function to it, namely the function to zoom in and zoom out on the meaning of a sentence in order to facilitate the computation of its meaning.

Moreover, it suggests that the definition of the syntactic head as the constituent from which the whole phrase inherits the syntactic properties is not necessarily equal to the definition of the syntactic head as the constituent reporting the meaning to which the whole phrase refers. However, as I am not considering grammars that add new meaning predicates but only grammars that bind variables on given predicates this statement requires further work to be assessed.
Finally, remember that an interactive web demonstration about the model is available through www.biologiaevolutiva.org/lsteels/tesi-emilia/rps

Part III

Conclusions

Chapter 7

CONCLUSIONS

This chapter discusses the relevance of the results of the thesis in the context of the research hypotheses.
In the thesis I support two hypotheses:

1. the first one is the hypothesis that phrase structure is motivated by computational needs
2. the second and main hypothesis subscribes that the same mechanisms that guide a population of language users to self-organise a first-order phrase structure grammar are also able to lead to a *recursive* phrase structure grammar when a higher-order phrase structure is required.

Concerning the first hypothesis I provided models for validating the function of phrase structure to reduce the computational cost of semantic interpretation. I did it first for the case of first-order phrase structure, though afterwards it was also checked for the case of higher-order phrase structure. My main contribution is to show that selective pressures on the computational cost for interpretation help a population of artificial agents playing language games to agree on a grammar exhibiting phrase structure. Moreover, I justified the choice of specific selective pressures by measuring the computational complexity of interpreting an utterance for a set of language strategies leading to language systems progressively similar to phrase structure, and showing how the complexity reduces as strategies get more similar to phrase structure. In other words, I gave evidence for the hypothesis that phrase structure is there to deal with complexity in parsing and interpretation.

In order to approach the study of the second hypothesis of the thesis, I built further on the agent-based model studied for the first hypothesis by extending the meanings to meanings including relations. A language system for these meanings requires higher-order phrase structure, and not simply first-order phrase structure. However, it is not assured that the language system agents negotiate exploits recursion even if they are forced to use higher-order phrase structure. One of the reasons is that they are not given any sort of syntactic categories in advance, nor lexical nor phrasal. Hence, I explored the extended model in order to study how agents were able to negotiate a grammar exploiting recursion. By exploring this model, I found out two major insights. The first one is that the fundamental learning operators assumed in the case of first-order phrase structure are sufficient for the emergence of recursive phrase structure; and the second one is that
a new conception of the syntactic head and its origins in grammars is derived from the model studied.

7.1 Summary of Results

In this section, the results of each chapter are presented again, and they are followed by a discussion on how their combination contributes to the study of the functions of phrase structure. Finally, some implementation issues and additional comments on the methodology are discussed.

7.1.1 Computational Adaptation of Language Strategies for Phrase Structure

The concept of cognitive effort is central to the understanding of the human condition and its evolution [116, 38], however there isn't yet an agreement on how to define it or how to measure it, despite attempts to do it for both experimental and computational work. The first results of the thesis had to do with defining which kinds of measures are informative when it comes to talk about cognitive effort in language processing, and how I realise them for artificial agents playing syntax games.

Concretely in chapter 3, I defined measures for the computational complexity of language strategies, measures for the computational cost of language processing, and measures for the size of an agent's grammar. I analysed and compared language strategies leading to systems progressively more similar to systems exhibiting phrase structure, and I compared the two cases considered in the thesis: first-order phrase structure and higher-order phrase structure as means to express the co-referentiality of predicate arguments. The outcome of this comparison showed that there is evidence to hypothesize that phrase structure is to some extend the result of a computational adaptation in language, given that language systems more similar to phrase structure grammars decrease the cognitive effort defined for language processing.

The main measure studied was the computational complexity, which is based on ideal worst cases which never occur in a communicative action. However, this frame is a reasonable starting ground because the properties of language systems studied are syntactic properties, namely the ordering of constituent. And these syntactic properties are not motivated to express particular semantics or pragmatics, instead their semantic function is modelled in a simplistic way by considering general co-referentiality of predicates.

7.1.2 Acquisition and Maintenance of Phrase Structure

The second group of results of the thesis came from modelling the acquisition of a language system exhibiting phrase structure. This is a first step towards building a computational model of the emergence of phrase structure, given that it is simpler than a model for the emergence. Moreover, acquisition is a requirement for the maintenance of a language system across generations.

I hypothesize and implement a computational model for the acquisition of phrase structure which consists of two artificial agents, a tutor and a learner, engaging in syntax game interactions. The tutor was initialized with a phrase structure grammar, while the learner was endowed with a learning strategy and its series of operators to acquire a grammar and to aim to become fully aligned with the tutor language. Moreover, the learner could uniquely get the role of the hearer and therefore he did not introduce variation on the language.

The language strategy that was tested responds to two principles: local processing and reduction of computational resources. The first one was implemented in the form of a contextual inference operator (CI), acting as well as a bias towards adjacent constituents, i.e. towards creating phrasal constructions. While the second one, in the form of a syntactic coercion operator (SC) and an ordering variation operator (OV), which was motivated to minimize the number of phrasal constructions and the number of lexical categories.

I considered only adoption operators because the focus of the model was to study the learner acquisition process.

While adoption operators are only applied when the agent has the role of the hearer, invention operators are only applied when the agent has the role of the speaker. In the next chapter I extended the model to study the emergence of phrase structure, and invention operators were studied in combination of alignment operators in a self-organised process. Hence, I didn't study invention operators in isolation in any model.

7.1.3 Cultural Emergence of Phrase Structure

Once validated the model for the acquisition of phrase structure, I extended it to show how a population of artificial agents was capable to self-organise a language system exhibiting first-order phrase structure playing syntax game interactions, i.e. showing how phrase structure could emerge culturally by the cultural processes of language games. Moreover, the model demonstrated that the reduction of the computational cost in language processing could be a functional adaptation of phrase structure.

I did that by hypothesizing an extended model with additional invention and alignment operators as mechanisms generating variation and applying selective forces on it, respectively. Moreover, I implemented and tested the model to show how a population of agents endowed with the set of learning operators tested for the acquisition (i.e. CI, OV and SC) plus the additional invention and alignment operators was collectively able to self-organise (creating and aligning) a language system exhibiting phrase structure.

Moreover, the alignment operator of the model was locally meant to avoid combinatorial search in parsing and semantic ambiguity in interpretation, while always keeping communicative accuracy. Hence, building further on the analytical arguments of chapter 3, the model empowered the hypothesis that those could be a motivation and driving force for the formation of phrase structure.

The results showed that the self-organisation of phrase structure required selection. The reason of this is that the kinds of grammars used to negotiate don't induce a useful notion of form competitor. Grammar variants acting as competitors of other grammar variants cannot be computed straight away from the representation because language processing is the result of the dependent application of a chain of constructions.

Therefore, lateral inhibition is not the fundamental mechanism for self-organisation as it is for the case of simpler grammars [33]. Lateral inhibition as it is used in the artificial language evolution literature leads to grammar agents organising themselves without forces external to the system, e.g. by identifying explicit competitors from grammars representations and strengthening or weakening them accordingly after every interaction. E.g. in the case of Naming Games, competitors usually are part of the same list of hypotheses; or in the case of word-based pattern systems (WBP), competitors are permutations of the same words.

Concretely, the hypothesized selective pressures on the reduction of the computational cost solved the problem of self-organising a phrase structure grammar whereas alignment using form competitors did not. Therefore, the model demonstrated a functional adaptation for phrase structure in the case of artificial agents playing syntax games.

Moreover, agents are capable to self-organise an SBP system that needs less number of constructions than a WBP or a SemBP with respect to attributes and attribute values. However, the scaling of the number of constructions is the same for both SBP and SemBP unless a constrained ontology is considered.

7.1.4 Cultural Emergence of Recursive Phrase Structure

Many human languages exploit recursive structure as a part of grammar. In order to study how recursion may originate and spread in language I generalized the model for the emergence of first-order phrase structure by extending the conceptualizations skills of the agents and allowing them to combine relational predicates, which have more than one variable.

Besides the operators in chapter 5, I included a natural extension of the syntactic coercion (SC) where coercion was allowed not only from a phrasal construction to a lexical construction but also between phrasal constructions. This requires a reformulation of the learning step where a new phrasal construction had to be created, which given that the core of the grammar variants competition is modelled as competition among phrasal constructions made totally sense.

Recursion in semantics was not required for the emergence of recursive phrasal constructions, i.e. constructions introducing as a phrasal category one of the categories matching. And the emerged grammars were capable to capture the structure of the given ontology because agents reached full phrasal alignment of grammars.

In the two previous models (chapters 4 and 5), the ontology that grammars express includes only predicates that had one variable, and utterances were analysed as unconnected phrases. Therefore, grammar had no control over the relative order of phrases and alignment could not be assured. On the contrary, the model in chapter 6 used an ontology which includes relations, i.e predicates that have more than one argument, and utterances express connected meanings and so they were analysed as a unique tree.

In the future, a connection between the two setups in chapter 5 and chapter 6 should be studied using the borders of the phrases as a source of alignment in order to model the gap between the two models.

7.1.5 Functions of Phrase Structure

In order to study the functions of phrase structures, I will consider the phenomenon of phrases and recursive phrases as behaviours, and use the levels of analysis for explanations of animal behaviour introduced by the Tinbergen's four questions [106].

Namely, Tinbergen describes complementary explanations for behavior by combining the *why* and *how* question, and the *dynamic* and *static* view.

	Dynamic View	Static View
How Question	Ontogeny (development)	Mechanism (causation)
Why Question	Phylogeny (evolution)	Function (adaptation)

Table 7.1: The table describes how the Tinbergen's 4 questions are defined.

There is no trivial match from Tinbergen's questions to phrase structure as a feature of the linguistic phenomena in humans. However, it is a convenient conceptual framework to situate the study of the functions of phrase structure. In the approach followed the use of language is viewed

as a social behavioural feature of humans, and phrase structure is viewed as a specific behaviour within such social behavior.
Considering then phrase structure as a specific behaviour, one way to account for Tinbergen's questions is:

a) Mechanism (causation) is given by the computational language processing model, which for the case of the thesis has been the FCG formalism. Specific assumptions from the models are discussed in chapters 4, 5 and 6.

b) Function (adaptation) includes the main research question of the thesis, which asks whether there is any functional adaptation for phrase structure related to the computational power for interpreting language. In which indeed a positive answer is derived.

c) Ontogeny (development), which in this case has to do with two aspects: learning strategies that individual language users have, and cultural processes in the form of communicative interactions in which they engage. This is the part which is formalised as computational models and tested using computer simulations.

d) Phylogeny (evolution) completes the study by exploring the evolutionary pathways that lead to the learning strategies used in c). This part is not approached in the thesis.

Notice that the case of the Naming Game paradigm for the cultural emergence of lexical systems could also be framed using the Tinbergen's 4 questions. However, in that case the adaptations for lexical systems are given by the Naming Game in the form of communicative pressures solely, without including computational pressures which are at the core of the work of the thesis.
Therfore, from the Tinbergen's four questions point of view, the thesis focus then on the *ontogenetic* view and on the *functional adaptation* of phrase structure, and concludes that the reduction of the computational cost of language interpretation is a function of phrase structure.

Finally, it is not a trivial question whether the results of the thesis shed any light on human language processing or deepen the understanding on human language phenomena at all. Such interpretation deeply depends of philosophical approaches, e.g. realism vs idealism, to discuss on to what extend symbolic processing systems are accurate models for language.
This thesis distances itself from such controversy narrowing the research question and merely following the classical AI methodology to study complex phenomena, which consists of reconstructing a system generating the targeted phenomena while minimizing the assumptions. So, generally I explore how the assumptions used so far for the emergence of lexical systems can apply to the case of grammatical systems. Discussions on the relevance of the underlying representations are extensive for existing models on the formation of lexical systems.

Finally, in the next two sections I overview a list of additional properties related to the model and its design and implementation.

7.1.6 Implementation Issues and Other Properties

The models don't specify any concrete interpretation for the time-scale of the communicative interactions in the model, further research on psychology and social sciences may help to enrich the model in that aspect.
Although I study the linguistic feature of word order and phrase structure as the corresponding grammar, the same model assumptions could be used to study lower or higher descriptive aspects of language as long as they use the relative order of sequential elements to express some systematic meaning relations, which should be supported by the corresponding ontology. E.g. the same model assumptions could be applied to study the origins of morphemes.

Additionally, the combination of how learning operators and grammatical constructions apply in the models can be viewed as a cognitive architecture that is able to support insight problem solving, operationalized and tested

for the specific case of language learning.
The implementation of the learning operators for the language strategies proposed fits naturally the architecture of insight problem solving for language processing as a way to restructure and expand the representation of grammar [77]. Routine-processing is modelled as language processing and the meta-level as language learning. Particularly, routing-processing as the application of grammatical constructions, and language learning as the application of learning operators in flexible grammar.
Moreover, the fact that the model supports insight problem solving provides a direct bridge to assess for its neuronal plausibility, given that there exists neuronal plausible model for insight problem solving and their fundamental operators performance can be compared. Such neuronal plausible models are also very interesting with respect to the integration of learning and evolution in the same shot, which is also convenient for the future steps of this research.

7.1.7 Methodological Aspects

The research developed in the thesis, and particularly, the models hypothesized, make use of both analytical and computational tools. Analytical tools are used to formalise and study certain aspects of the models, such as validate the choice of certain language strategies by analysing their computational complexity (see chapter 3), or predict the behaviour of the model concerning specifics aspects (see chapters 4, 5 and 6). On the other side, computational tools, and particularly agent-based simulations are used to computationally explore or validate as well certain aspects of the models by performing systematic computational experiments (see chapters 4, 5 and 6).

It depends on the sophistication of the assumptions whether a model can be studied analytically, i.e. as an equation-based model, or using computer simulations. In the thesis, both cases occur, although simulations occupy a bigger portion of the results achieved.
Agent-based simulations (ABS) are useful to cover situations that might

not be accessible using analytical models, however, it is very complicated for ABS (and it might be impossible) to get an exhaustive model for the given assumptions, which on the other side, can be achieved for an analytical model and the corresponding analysis. A complementary cost of using some kinds of ABS is that their implementation may be highly time consuming, and so may be the computer resources needed to run them. In the case of the models used in this thesis, it took up to more than one year of work to have them operational, and the computer resources needed to run them are not too high for a single experiment but may increase a lot depending on the aspect of the model performance that is analysed.

In the remaining sections of this chapter, I compare the results of the thesis to related studies, I give hints to future investigations, and I close with some final remarks.

7.2 Comparison to other models

The models proposed in the thesis generally inform on why grammar is needed and puts forward a concrete hypothesis on the minimal cognitive requirements to see the emergence of phrase structure and recursive phrase structure. Although the models rely heavily on complex symbolic machinery that have a compositional strategy implicitly coded, they give important insights on the missing link between the emergence of a language-ready brain and the emergence of complex human language [2].

This section briefly discusses how the results of the thesis compare to previous related results and what is the state-of-the-art of the questions approached in this thesis.

The core contribution of the thesis is a model for the emergence of phrases and recursion in the form of phrase structure and recursive phrase structure. Hence, I compare some of the contemporary models for either the cultural emergence of other aspects of grammar, or for other explanations on the emergence of phrases or recursion. Moreover, I compare as well the language processing model used in the thesis against other approaches to model language processing.

7.2.1 Models of the Cultural Emergence of Language

Concerning models for the cultural emergence of other aspects of grammar or syntax [108, 14, 79], the main novelty of the thesis is that the grammar is driven by computational needs instead of by semantics, where by semantics refers to the need for expressing a particular kind of meanings. e.g. event roles, colour space, quantification, etc.

In these models, the focus is made on how agents manage to negotiate a grammar that covers the meanings. In the models reported in the thesis instead, agents are given the ability to express all the meanings they encounter, which are abstract and arbitrary, and the main motivation to negotiate a grammar comes from decreasing the cognitive effort in parsing and interpreting an utterance, and of course, the communicative need to interpret what the speaker intended.

There is a model for the emergence of agreement systems [11] that does emphasize the role of computational needs in the motivation for the system to arise. However, the variability of agreement systems requires a lower level description of the strategies and doesn't allow a comparison between language strategies in the way provided in the thesis, but rather a mechanistic validation for some particular ones.

Some other models of the cultural emergence of phrase structure [57] as well do consider more general and abstract meanings, but they assume syntactic categories are given and the focus is on the competition between a grammatical strategy and a strategy using holistic words.

Finally, there are other models accounting for the cultural emergence of hierarchy and recursion. In [82], the emergence of hierarchical phrase structure is also a derivation of the meanings in use, for which logical expressions are considered, syntactic categories derive from the grammar formalism of categorial grammar that it is used, and the main driving force towards convergence is again exclusively communication. In [26], recursive phrases again are compared to holistic languages and their usages motivated only by communicative needs, syntactic categories are given, and all the cases of syntactic recursion that arise are derived from cases of semantic recursion.

Notice that in the models where phrase structure strategies are compared to holistic languages, agents start from the scratch with no grammar and no lexicon, whereas in the models of the thesis agents build a grammar on top of a compositional lexicon, and they are biased towards phrases (by the CI operator). Moreover, semantic recursion is not a requirement for the emergence of syntactic recursion, the role of discrimination vs description games is relevant, and not only communicative pressures are used but also computational pressures.

Moreover, other selective pressures than the reduction of computational complexity have been used on previous studies on the cultural emergence of lexical systems and grammatical systems. Such as pressures for coordinating actions in a given world, pressures for reducing mistakes in communication, learnability pressures or pressures for categorizing a physical world.

All theses models showed different ways in which a compositional lexicon may originate from the scratch as a self-organised process by a population of artificial agents. This is relevant to the thesis given that the models proposed assume a compositional lexicon, because the way in which the lexical system covers the the conceptual system gives potential compositional power. Therefore, the models of the thesis rely on previous results on the emergence of compositional lexical systems.

There are more efforts in research devoted to the study of self-organised lexical systems than to self-organised grammatical systems, however there is general consensus on the will to move in the direction of putting more efforts there too [62].

7.2.2 Models of the Origins of Recursion

Also, there are many other approaches to investigate the origins or emergence of recursion in language beyond the simulation of its cultural evolution. Many authors, generally agree on the fact that: *While there* *are significant recurrent patterns in* (language) *organization, these are better explained as stable engineering solutions satisfying multiple design constraints, reflecting both cultural-historical factors and the constraints of human cognition* [67]. Therefore, results following other approaches are likely to be complementary to the models studied in the thesis.

Some hypotheses that have given successful results consider the faculty of recursion as a by-product of other phenomena, e.g. a usage-based byproduct [109, 20] or an extension of the cognition related to numerals [115]. Another approach that has been widely explored is about the function of recursion as a requirement or as a positive impact for language acquisition [5] or for cultural transmission [63]. Moreover, others are looking for the differences and similarities of language properties and bird songs [10] or other non-human species [74] to discover the keys to the origins of recursion.

7.2.3 Models of Language Processing

There is a widespread consensus in the communities of Natural Language Processing, Computational Linguistics and Artificial Intelligence that an accurate model of human languages will require both symbolic and sub-symbolic models. In the thesis I used only symbolic models based on the FCG formalisation because this facilitates the implementations of models supporting adaptation, e.g. keeping competing constructions.

Advanced language models based on artificial neural networks (ANN) are better in abstracting features from linguistic data than previous symbolic models, but as end-to-end models they are usually built from the scratch (or from some cleverly chosen prewired network) and trained against some specific dataset. There are few attempts to explore ANN language models that iteratively learn from external data.

7.3 Future Directions

This section overviews both new avenues open by the thesis and complementary avenues that could be explored.
In the first place, the models proposed can be easily extended to study derivations of the research questions. First, the innovative way in which the selective pressures are implemented opens up the perspective to a more general framework for studying selective pressures in linguistic aspects beyond communicative pressures. The core of this new framework lies on the exploration of alignment operators which are directly influenced by the online performance of the constructions in language processing, instead of just their final communicative outcome. And also interpreting as linguistic pressures the criteria used by the operators to modify the priorities among constructions by favouring some constructions over others.
Second, the same implementation used to study the models can be adapted to account as well for other cultural and linguistic phenomena, as it could be:

1. studying the role of the *cultural structure* by comparing experiments using different distributions of possible situations, i.e. modifying the way in which situations are generated.

2. adding a mortality rate in the model allowing agents to *die* and *get born* and let the newborns invent new grammar in order to study the cultural transmission across generation.

3. extending the ontologies to account for the distinction between intensional and extensional meanings, i.e. in the noun phrase *a beautiful dancer*,“beautiful” could refer to the dancer as *a person* or as *the way he dances*. Supporting this extension requires only increasing by one the number of variables that the meaning predicates have.

4. studying the intrinsic evolution of meaning given by the grammar itself. If agents are allowed to build up meanings directly from their grammars, and produce utterances without the constraints of

a context, a notion of metaphor can be modelled, and the space of semantic types gets modified subsequently.

5. keeping the original constructions when a grammar update is made, i.e. when the construction is modified for example by the application of syntactic coercion. This would be a fairer approach to an evolutionary framework for grammars.

And third, the model in chapter 6 showed the limitations of phrase structure in discriminating some of the elements in an utterance, motivating the use of other grammatical features such as anaphoras, morphology (e.g. agreement markers) or function words (e.g. *that*). Extending the linguistic module of the model and studying phrase structure in combination with these grammatical features will give new insights into the way our languages evolve.
Futhermore, language strategies for lexical systems [112] could be combined with language strategies for phrase structure in order to study the evolution of grammars from the scratch.
Another extension from another point of view is to endow grammatical constructions with the ability to add new meaning (and not uniquely binding existing variables).
Other perspectives that should be considered are using case studies from real language examples [30], or approaching the study of the origins of the learning strategies, by assuming only more basic operations that agents combine to explore language strategies. The latter can also open the possibility to study how given language strategies can get more general. On the other side, the study of additional functions of phrase structure should also be explored.
Finally, there are several questions that I started to follow during my thesis time but didn't lead to conclusive results yet. One of them is the study of the evolution of grammatical constructions in an agent's grammar. I started to define useful metrics within and across agents to keep track of the evolvable grammars, considering distances between grammatical constructions and between grammars [49], but many efforts are still needed. Moreover, I approached this issue by considering the reuse of

constructions information to create new constructions as a reproductive system and applied techniques for visualization and analysis from biology in order to understand better the nature of these changes, see figure 7.1. It is interesting to find similarities between other systems, particularly biological systems, and consider the application of analyses techniques specific to those systems. Another approach followed was to consider grammatical constructions used in the same grammar application as constructions cooperating and those wrongly used as constructions competing, and again tried to make use of techniques from biology. In essence, all the approaches have to do with the idea of considering the grammar or some of its derivations as a living population.

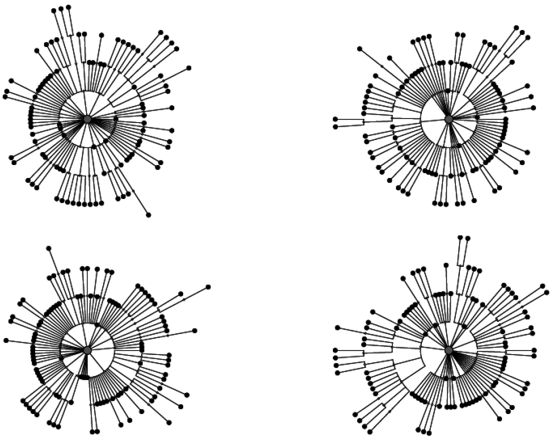


Figure 7.1: The internal grammar evolution of agents focussing on the updating of constructions by the means of learning operators. Each node represents a constructions and the offspring consists of new construction instances that arise after the application of a learning operator to the original construction.

In figure 7.1, having a single child means that one construction was modified (or created) after the application of the operator, while having two children means that two constructions were modified (or created) after the application of an operator, for example in the creation of a new pattern in combination with syntactic coercion. When constructions are created without any previous construction as a referent (e.g. contextual inference without previous constructions) they hang from the center.

On another front, part of my time during the last months of the thesis has been devoted to work on a neuronal plausible implementation of the fundamental operations used in the learning operators proposed. These efforts have lead to a model based on the theory that learning guides evolution [31] which uses artificial attractor networks with learning. The model is a Naming Game model and hence it approaches the study of strategies to play language games by considering implicit representations of language strategies and language processing by using neural models. Concretely, the successful neural implementation of this language game proposes a framework to study grammar evolution (of an agent) integrating learning and evolution in the same structural representation. This opens an interesting connection to the models of the thesis, where a set of learning operators for the same language strategy is used as well for learning (acquisition) and evolving (emergence).

Additionally, I built several tools to visualize the output of the models and some of them will be soon included as components of the new Babel Platform, and they will be accessible to other researchers that can keep using them and adapting them to their needs.

7.4 Final Remarks

This thesis is another proof that sophisticated grammars aren't needed to study relevant properties of artificial systems and their use as models for our languages. Instead, what is important is to state clearly the assumptions of the models (including those implicit on the implementations) and state and narrow the particular aspects of language that are analysed. This will allow to continue the work done on the models and expand the interest for the research field of artificial language evolution.

In this case, remember that all simulated data and the implementation of the models are available for downloading in

www.biologiaevolutiva.org/lsteels/tesi-emilia.zip
The learning operators studied in the thesis are models for cognitive capacities. However, the way in which they are used in the models doesn't subscribe to any concrete hypothesis about their origin. This means that it is not claimed to what extend they are biologically determined or culturally acquired.

Moreover, the ontologies proposed model mental representations, and it is unlikely they happened to be there before a grammar able to express them was there. The approach followed in the thesis is a first step and future models should explore more deeply the relation between the evolution of ontologies (and their representations) and language systems.

The models of the thesis are meant to be the starting ground for further research on the research questions. The models are to a high extend modular and this facilitates the integration of alternative or more extensive hypotheses that can be compared. E.g any sort of word order bias, bias to process words from the left or from the right, take into account biased situations (i.e. some objects more common than others), and of course scaling up the proposed models both for agents and ontology size.

My perspective on the topic changed substantially from the beginning to the end of the thesis, the problems faced during the design, development and testing of models open up the box for many new ideas and a broader comprehension on the original question. That's why I hope that the work devoted to the thesis will be worth it and that the models and results can also help other researchers exploring and validating further and more detailed hypotheses about the cultural evolution of grammar.

Part IV

Appendices

Appendix A

PUBLICATIONS

In this appendix I list the peer-reviewed publications that I have published during my thesis time, along with a brief description on how they relate to the thesis.

- Steels, L. & Garcia-Casademont, E. (2015). Ambiguity and the origins of syntax. *The Linguistic Review*, 32(1), 37-60.

  It introduces and discusses aspects of the first language game proposed. Directly related to chapters 4, 5 and 6.

- Garcia-Casademont, E. (2015). A case study in the emergence of recursive phrase structure. *Proceedings of the First Complex Systems Digital Campus World E-Conference 2015*

  It introduces a part of the content of chapter 3 and 6.

- Garcia-Casademont, E. (2014). Information transfer is not enough to preserve systematicity. *Proceedings of the Student Conference of Complexity Science*

  It introduces relevant results for the discussion in chapter 7.

- Steels, L. & Garcia-Casademont, E. (2015). How to play the Syntax Game. *Proceedings of the European Conference on Artificial Life*, 479-486.

It discusses relevant parts of chapters 5 and 7.

- Garcia-Casademont, E. & Steels, L. (2014). Strategies for the emergence of first-order phrase structure. In E. A. Cartmill, S. Roberts, H. Lyn & H. Cornish (Eds.), *The evolution of language - Proceedings of the 10th International Conference (Evolang-X)*, 50-57. World Scientific Parts of chapters 3, 5 and 7.
- Garcia-Casademont, E. & Steels, L. (2016). Insight Grammar Learning. *Journal of Cognitive Science*, 17(1), 27-62. Part of chapter 4.
- Garcia-Casademont, E. & Steels, L. (2015). Usage-based Grammar Learning as Insight Problem Solving. *Proceedings of the EuroAsian-Pacific Joint Conference on Cognitive Science*, 1419:0039. Part of chapter 4.
- Steels, L. & Garcia-Casademont, E. The Origins of Recursive Grammar. Part of chapter 6.
- Garcia-Casademont, E. (2014). Interpretation Processes: Analysis of the complexity of different language systems. *Evolang X Workshop: How Grammaticalization Processes Create Grammar*. Part of chapter 3.
- Garcia-Casademont, E. (2015). Tracking Language Evolution: Construction Distance. Essence Workshop I introduces relevant aspects for the discussion in chapters 5 and 7.

Publications outside the scope of the thesis:

- de Vladar, H. P., Garcia-Casademont, E. & Steels, L. (2016). Language Imitation Games with a Darwinian Cognitive Architecture (submitted)
- Khalid, F., Garcia-Casademont, E., Laborde, S., Lagesse, C., & Lusczek, E. (2014). How Artificial Intelligence Can Inform Neuroscience: A Recipe for Conscious Machines? *Proceedings of CSSS 2014*.

- Mathis, C., Yu, L., Aguilar-Rodríguez, J., Davidovic, S., Mehta, R., Garcia-Casademont, E., Qiao, Z., Kharrazi, A., Vroomans, R., & Gibbons, S. M. (2014). The tradeoff between division of labor and robustness in complex, adaptive systems is shaped by environmental stability. *Proceedings of CSSS 2014.*

- Garcia-Casademont, E., Gorsky, S., Graebner, C., Laborde, S. & Martínez-Vaquero, L. (2014). One complex system, three approaches: conceptual and methodological insights. *Proceedings of CSSS 2014.*

Appendix B

VISUALIZATION AND WEB DEMONSTRATIONS

There are 4 interactive web demonstrations, one referred to in chapter 2 and the other three referred to in chapters 4, 5 and 6. The first one describes two syntax game interactions in detail and the latter illustrate the main aspects of the models discussed in the corresponding chapters, these ones are divided in two parts:

1. Learning Operators: Without operators, operators in flexible grammar application and operators in grammar updating.
2. Interactions: Failure and successful interactions, and interactions in different phases of the experimental runs.

Interactive Web Demonstrations

The web demonstrations are self-contained, however some tips are useful in order to take advantage of all their functionalities.

- Click on elements or nodes in language processing (plus and dot symbols), one time and two times to expand/collapse them in order to visualize/hide their internals.

- Click on variables to see the equalities as color equalities: By clicking on symbols and variables, you can easily see how different parts of a construction or transient structure are linked to each other, or which values are shared across feature structures.

- Constructions are blue, and the application result of a construction or a learning operator in flexible grammar is green.

- syn-cat coercion, syn-cat group and footprints are FCG technical features that enable the low level implementation of some of the aspects of the operators, but that have no direct mention in the thesis.

The web demonstration is built under the platform Babel2, you can find more information and the usage of the demo in the site https://www.fcg-net.org/projects/web-demonstration-guide/

Visualization

Most of the tools and graph types to visualize data that I used in the thesis were specifically created for this research. In order to make them accessible to everyone willing to do research in Artificial Language Evolution they will get integrated to the Babel2 platform in the coming releases.

Appendix C

LISP CODE OF SOME OF THE CORE MECHANISMS

Generate Situations

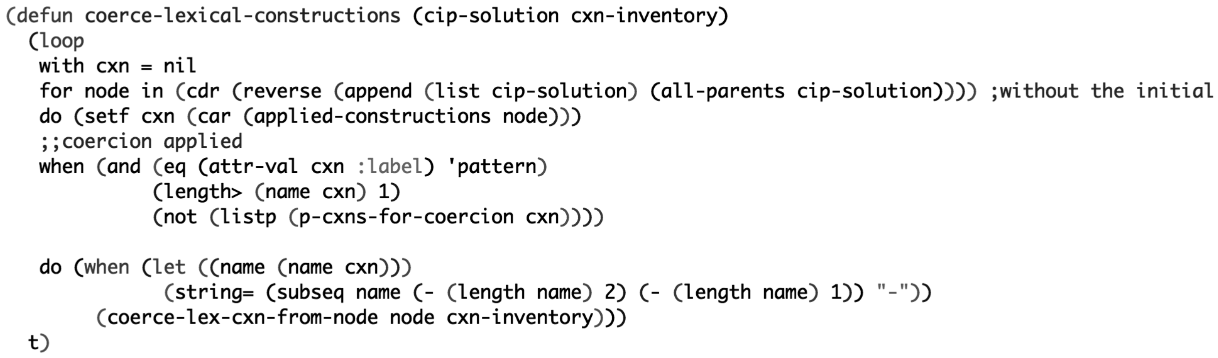
```
(defun generate-situation-fixed-length (nb-of-words objs-in-situation world)
  (labels ((get-random-word-nb-sequence (nb-of-words)
             (cond ((= nb-of-words 2) '(2))
                   ((= nb-of-words 3) (random-elt '((1 2) (3))))
                   ((= nb-of-words 4) (random-elt '((2 2) (3 1) (4))))
                   ((= nb-of-words 5) (random-elt '((2 3) (4 1) (5)))))))
    (let ((nb-of-words-per-object (get-random-word-nb-sequence nb-of-words)))
      (loop
       for nb-of-words-object in nb-of-words-per-object
       for object = (loop for obj = (create-object (random-elt (mapcar #'car (attributes world)))
                                                   world :nb-of-words nb-of-words-object)
                          if (not (member (mapcar #'first-and-second obj) (append situation objs-in-situation)
                                          :test #'(lambda(x y) (permutation-of? x y :test #'equalp))
                                          :key #'(lambda(x) (mapcar #'first-and-second x))))
                          do (return obj))
       when (not (find-if #'(lambda(attr) (find attr (reduce #'append situation) :key #'first-and-second)) object))
       collect object into situation
       finally return (if (member nil situation) nil situation)))))
```

```
(defun generate-situation (nb-of-words nb-of-object-groups world)
  (loop
   with situation = nil
   repeat nb-of-object-groups
   do (loop for sit = (generate-situation-fixed-length nb-of-words situation world)
            when sit do (setf situation (append situation sit))
            until sit)
   finally (return situation)))
```

Create New Phrasal Constructions

```
(defun create-new-patterns (cip-solution cxn-inventory)
  (let* ((left-pole (left-pole-structure (car-resulting-cfs (cipn-car cip-solution))))
         (groups (find-all-if  #'(lambda(x) (and (unit-feature-value x 'subunits)
                                                 (or (not (unit-feature-value x 'syn-cat))
                                                     (listp (unit-feature-value x 'syn-cat))))) left-pole)))
    (loop
     with groups-created = nil
     for group in groups
     for subunits = (subunits group left-pole)
          do
          (let ((cxn-name (build-cxn-name subunits)))
            (when (and cxn-name (not (member nil cxn-name)))
              (add-phrasal-cxn-sb cxn-name cxn-inventory)
              (setf groups-created t)))
          finally (when groups-created (notify learning-event-creation (agent cxn-inventory))))
    t))
```

Coerce Lexical Constructions

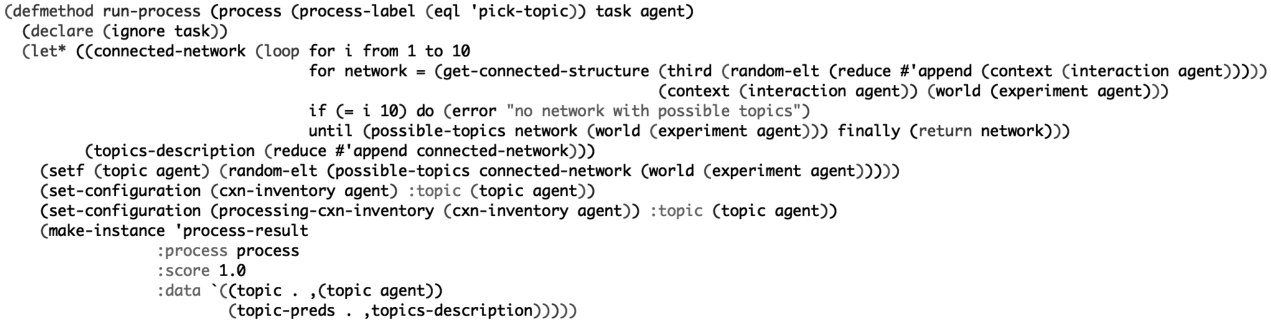


```
(defun coerce-lexical-constructions (cip-solution cxn-inventory)
  (loop
   with cxn = nil
   for node in (cdr (reverse (append (list cip-solution) (all-parents cip-solution)))) ;without the initial
   do (setf cxn (car (applied-constructions node)))
   ;;coercion applied
   when (and (eq (attr-val cxn :label) 'pattern)
             (length> (name cxn) 1)
             (not (listp (p-cxns-for-coercion cxn))))

   do (when (let ((name (name cxn)))
              (string= (subseq name (- (length name) 2) (- (length name) 1)) "-"))
        (coerce-lex-cxn-from-node node cxn-inventory)))
  t)
```

Online Interpretation

```
(defmethod cip-node-test ((node cip-node) (mode (eql :check-situation)))
  (labels ((match-with-the-situation (meaning-passed bindings)
             (setf (fcg-emilia::cipn-bindings node) bindings)
             (setf (fcg-emilia::cipn-bindings-source node) (fcg-emilia::cipn-bindings (car (all-parents node))))
             (setf (fcg-emilia::cipn-meaning node) meaning-passed)
             t)
            (no-match-with-the-situation ()
             (push 'check-situation (statuses node))
             (setf (fully-expanded? node) t)
             nil))
    (if (or (eq (label-cxn (car (applied-constructions node))) 'pattern) (eq (label-cxn (car (applied-constructions node))) 'group))
      (ecase (get-configuration (construction-inventory node) :current-direction)
        ('-> t)
        ('<-
           (let* ((new-meaning (get-new-meaning-from-node node))
                  (accumulated-meaning (get-accumulated-meaning-from-node node new-meaning)))
             (let ((situation-preds (meaning-objects-from-set (get-configuration (original-cxn-set (construction-inventory node)) :situation)))
                   bindings
                   meaning-passed)
               (if (fcg-emilia::cipn-bindings (car (all-parents node)))
                 (progn
                   (setf bindings (remove-duplicates
                                   (loop for bindings in (fcg-emilia::cipn-bindings (car (all-parents node)))
                                         append (unify (cons '== accumulated-meaning)
                                                       situation-preds
                                                       (list (or (remove-if #'(lambda(x)
                                                                                 (not (member (car x) accumulated-meaning
                                                                                              :key 'third)))
                                                                            bindings) +no-bindings+))))
                                   :test #'(lambda(x y) (permutation-of? x y :test #'equalp))))
                   (setf meaning-passed accumulated-meaning))
                 (progn
                   (setf bindings (unify (cons '== new-meaning) situation-preds))
                   (setf meaning-passed new-meaning)))
               (if bindings
                 (match-with-the-situation meaning-passed bindings)
                 (no-match-with-the-situation)))))) t)))
```

Speaker topic selection (for chapter 6)



```
(defmethod run-process (process (process-label (eql 'pick-topic)) task agent)
  (declare (ignore task))
  (let* ((connected-network (loop for i from 1 to 10
                                  for network = (get-connected-structure (third (random-elt (reduce #'append (context (interaction agent))))))
                                                                         (context (interaction agent)) (world (experiment agent)))
                                  if (= i 10) do (error "no network with possible topics")
                                  until (possible-topics network (world (experiment agent))) finally (return network)))
         (topics-description (reduce #'append connected-network)))
    (setf (topic agent) (random-elt (possible-topics connected-network (world (experiment agent)))))
    (set-configuration (cxn-inventory agent) :topic (topic agent))
    (set-configuration (processing-cxn-inventory (cxn-inventory agent)) :topic (topic agent))
    (make-instance 'process-result
                 :process process
                 :score 1.0
                 :data `((topic . ,(topic agent))
                         (topic-preds . ,topics-description)))))
```

Bibliography

[1] ABOITIZ, F. Gestures, vocalizations, and memory in language origins. *Front. Evol. Neurosci. 4*, 2 (2012).

[2] ARBIB, M. Grounding the mirror system hypothesis for the evolution of the language-ready brain. In *Simulating the Evolution of Language*, A. Cangelosi and D. Parisi, Eds. Springer Verlag, London, 2002, pp. 229–254.

[3] BARONCHELLI, A. Role of feedback and broadcasting in the naming game. *Physical Review E 83*, 4 (2011), 046103.

[4] BARONCHELLI, A., DALL'ASTA, L., BARRAT, A., AND LORETO, V. Topology-induced coarsening in language games. *Phys. Rev. E 1*, 73 (2006), 015102.

[5] 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*, T. Briscoe, Ed. Cambridge University Press, Cambridge, 2002.

[6] BAXTER, G. J., BLYTHE, R. A., CROFT, W., AND MCKANE, A. J. Utterance selection model of language change. *Phys. Rev. E 73* (2006).

[7] BELL, E. T. The iterated exponential integers. *The Annals of Mathematics 39* (1938), 539–557.

[8] BELPAEME, T., AND BLEYS, J. Explaining universal color categories through a constrained acquisition process. *Adaptive Behavior 13*, 4 (2005), 293–310.

[9] BERGEN, B. K., AND CHANG, N. Embodied Construction Grammar. In *Construction Grammars: Cognitive Grounding and Theoretical Extensions*, J.-O. Östman and M. Fried, Eds. John Benjamins, Amsterdam, 2005.

[10] BERWICK, R., OKANOYA, K., BECKERS, G., AND BOLHUIS, J. Songs to syntax: the linguistics of birdsong. *Trends Cogn Sci. 15 (3)* (2013), 113–21.

[11] BEULS, K., AND STEELS, L. Agent-based models of strategies for the emergence and evolution of grammatical agreement. *PLOS ONE 8(3)* (2013).

[12] BEULS, K., VAN TRIJP, R., AND WELLENS, P. Diagnostics and repairs in Fluid Construction Grammar. In *Language Grounding in Robots*, L. Steels and M. Hild, Eds. Springer Verlag, Berlin, 2012.

[13] BICKERTON, D. Language evolution: A brief guide for linguists. *Lingua 117* (2007), 510–526.

[14] BLEYS, J. *Language Strategies for the Domain of Colour*. PhD thesis, Vrije Universiteit Brussel, 2010.

[15] BLEYS, J., AND STEELS, L. *Linguistic Selection of Language Strategies*. Springer Berlin Heidelberg, 2011, pp. 150–157.

[16] BOD, R., AND SCHA, R. *Data-oriented Parsing*. Center for the Study of Language and Information, Stanford, Palo Alto, 2003.

[17] BOWERMAN, M., AND CHOI, S. Shaping meanings for language: Universal and language-specific in the acquisition of spatial semantic categories. In *Language Acquisition and Conceptual Development*, M. Bowerman and S. C. Levinson, Eds. Cambridge University Press, Cambridge, 2001, pp. 132–158.

[18] CHOMSKY, N. *Syntactic Structures*. Mouton, The Hague, 1957.

[19] CHRISTIANSEN, M. H., AND CHATER, N. Language as shaped by the brain. *Behavioral and Brain Sciences 31* (2008), 489–558.

[20] CHRISTIANSEN, M. H., AND MACDONALD, M. C. A usage-based approach to recursion in sentence processing. *Language Learning 59* (2009), 126–161.

[21] CORBALLIS, M. C. The recursive mind: The origins of human language, thought, and civilization, 2014.

[22] CROFT, W. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford UP, Oxford, 2001.

[23] CROFT, W. Evolutionary linguistics. *Annual Review of Anthropology 37*, 4 (2008), 219–234.

[24] DALL'ASTA, L., BARONCHELLI, A., BARRAT, A., AND LORETO, V. Non-equilibrium dynamics of language games on complex networks. *Phys. Rev. E 1*, 74 (2006), 036105.

[25] DANIEL L. EVERETT. Cultural constraints on grammar and cognition in pirahã another look at the design features of human language. *Current Anthropology 46*, 4 (2005), 621–646.

[26] DE BEULE, J. *Compositionality, Hierarchy and Recursion in Language. A Case Study in Fluid Construction Grammar*. PhD thesis, Vrije Universiteit Brussel, 2007.

[27] DE BEULE, J., AND DE VYLDER, B. Does language shape the way we conceptualize the world? In *the XVII Annual Conference of the Cognitive Science Society* (2005), B. G. Bara, L. Barsalou, and M. Bucciarelli, Eds., pp. 559–564.

[28] DE JONG, E. D. *Autonomous Formation of Concepts and Communication*. PhD thesis, Vrije Universiteit Brussel, Brussels, 2000.

[29] DE LA HIGUERA, C. Grammatical inference: learning automata and grammars, 2010.

[30] DE VELDE F., V. The emergence of modification patterns in the dutch noun phrase. *Linguistics 47*, 4 (2009), 1021–1049.

[31] DE VLADAR, H. P., AND SZATHMÁRY, E. Neuronal boost to evolutionary dynamics. *Interface Focus 5*, 6 (2015).

[32] DE VYLDER, B. *The Evolution of Conventions in Multi-Agent Systems*. PhD thesis, Vrije Universiteit Brussel, Brussels, 2007.

[33] DE VYLDER, B., AND TUYLS, K. How to reach linguistic consensus: A proof of convergence for the naming game. *Journal of Theoretical Biology 242*, 4 (October 2006), 818–831.

[34] DEUTSCHER, G. *The Unfolding of Language. An Evolutionary Tour of Mankind's Greatest Invention*. Metropolitan, New York, 2005.

[35] DI SCIULLO, M., AND BOECKX, C., Eds. *The Biolinguistic Enterprise. New perspectives on the Evolution and Nature of the Human Language Faculty.* Oxford University Press, 2011.

[36] EVANS, N., AND LEVINSON, S. C. The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and brain sciences 32*, 05 (2009), 429–448.

[37] FERBER, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman, Harlow, UK, 1999.

[38] FERRER I CANCHO, R., AND SOLE, R. Least effort and the origins of scaling in human language. In *Proceedings of the National Academy of Sciences USA* (2003), vol. 100, pp. 788–791.

[39] FILLMORE, C. J., KAY, P., AND O'CONNOR, M. C. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language* (1988), 501–538.

[40] FITCH, W. *The Evolution of Language*. Approaches to the Evolution of Language. Cambridge University Press, 2010.

[41] FRIEDERICI, A. Processing local transitions versus long-distance syntactic hierarchies. *Trends in Cognitive Science 8* (2004), 245–47.

[42] FRIEDERICI, A. D. Towards a neural basis of auditory sentence processing. *Trends in Cognitive Sciences 6*, 2 (2002), 78 – 84.

[43] FRIEDRICH, P. Semantic structure and social structure: an instance from russian. *Explorations in cultural anthropology* (1964).

[44] FUJITA, K., AND BOECKX, C. A., Eds. *Advances in Biolinguistics: The Human Language Faculty and Its Biological Basis*. Routledge, 2016.

[45] FUTRELL, R., MAHOWALD, K., AND GIBSON, E. Large-scale evidence of dependency length minimization in 37 languages. *PNAS* (2015).

[46] GALANTUCCI, B. Experimental semiotics: A new approach for studying communication as a form of joint action. *Topics in Cognitive Science 1*, 2 (2009), 393–410.

[47] GARCIA-CASADEMONT, E. Information transfer is not enough to preserve systematicity. *Proceedings of the Student Conference of Complexity Science* (2014).

[48] GARCIA-CASADEMONT, E. A case study in the emergence of recursive phrase structure. In *Proceedings of the First Complex Systems Digital Campus World E-Conference 2015* (2015), Springer Proceedings in Complexity.

[49] GARCIA-CASADEMONT, E. Tracking language evolution: Construction distance. *Essence Workshop* (2015).

[50] GARCIA-CASADEMONT, E., AND STEELS, L. Strategies for the emergence of first-order phrase structure. In *The Evolution of Language. Proceedings of the 10th International Conference (EVOLANG X Vienna)* (2014), A. D. Smith, K. Smith, and R. Ferrer i Cancho, Eds., World Scientific Press, pp. 50–57.

[51] GARCIA-CASADEMONT, E., AND STEELS, L. Insight grammar learning. *Journal of Cognitive Science 17-1* (2016), 27–62.

[52] GÄRDENFORS, P. Cooperation and the evolution of symbolic communication. In *Evolution of Communication Systems: A Comparative Approach*, D. K. Oller and U. Griebel, Eds. MIT Press, Cambridge, MA, 2004, pp. 237–256.

[53] GENTNER, T. Q., FENN, K., D., M., AND NUSBAUM, H. Recursive syntactic pattern learning by songbirds. *Nature 440*, 7088 (2006), 1204–1207.

[54] GOLDBERG, A. E. Construction grammar: a construction grammar approach to argument structure, 1995.

[55] GOLDBERG, A. E. Constructions: A new theoretical approach to language. *Trends in Cognitive Science 7*, 5 (May 2003), 219–224.

[56] GOLDBERG, A. E. *Constructions at Work, The Nature of Generalization in Language*. Oxford University Press, Great Clarendon Street, Oxford, 2006.

[57] GONG, T. Exploring the roles of horizontal, vertical, and oblique transmissions in language evolution. *Adaptive Behavior 18* (2010), 356–376.

[58] GONG, T., AND MINETT, J. W. A simulation study in word order bias. *Interaction Studies* (2009).

[59] HAUSER, M. D., CHOMSKY, N., AND FITCH, W. T. The faculty of language: What is it, who has it, and how did it evolve? *science 298*, 5598 (2002), 1569–1579.

[60] HERRMANN, E., CALL, J., HERNÁNDEZ-LLOREDA, M. V., HARE, B., AND TOMASELLO, M. Humans have evolved specialized skills of social cognition: the cultural intelligence hypothesis. *Science 317*, 5843 (2007), 1360–1366.

[61] JABLONKA, E., AND LAMB, M. *Evolution in Four Dimensions: Genetic, Epigenetic, Behavioral, and Symbolic Variation in the History of Life*. Bradford books. MIT Press, 2005.

[62] JÄGER, H., STEELS, L., BARONCHELLI, A., BRISCOE, E., CHRISTIANSEN, M. H., GRIFFITHS, T., JAGER, G., KIRBY, S., KOMAROVA, N., RICHERSON, P. J., AND TRIESCH, J. What can mathematical, computational and robotic models tell us about the origins of syntax? In *Biological Foundations and Origin of Syntax*, D. Bickerton and E. Szathmáry, Eds. MIT Press, USA, January 2009, pp. 385–410. originally published in D. Bickerton & E. Szathmáry (Eds.), Biological Foundations and Origin of Syntax, ©2009 MIT and the Frankfurt Institute for Advanced Studies, Cambridge, MA: The MIT Press, (pp. 385-410).

[63] KIRBY, S. Learning, bottlenecks and the evolution of recursive syntax. In *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, T. Briscoe, Ed. Cambridge University Press, Cambridge, 2002, pp. 173–204.

[64] KOEHLER, W. *The Mentality of Apes*. Routledge and Kegan Paul, New York, 1956.

[65] LEVINSON, S. C. Yéli dnye and the theory of basic color terms. *J. Linguistic Anthropology 10* (2000), 3–55.

[66] LEVINSON, S. C., AND EVANS, N. Time for a sea-change in linguistics: Response to comments on ?the myth of language universals? *Psychological Review 120 (12)* (2010), 2733–2758.

[67] LEVINSON, S. C., AND GRAY, R. D. Tools from evolutionary biology shed new light on the diversification of languages. *Trends in cognitive sciences 16*, 3 (2012), 167–173.

[68] LOETZSCH, M., WELLENS, P., DE BEULE, J., BLEYS, J., AND VAN TRIJP, R. The babel2 manual. Tech. Rep. AI-Memo 01-08, AI-Lab VUB, Brussels, 2008.

[69] LORETO, V., BARONCHELLI, A., MUKHERJEE, A., PUGLISI, A., AND TRIA, F. Statistical physics of language dynamics. *Journal of Statistical Mechanics: Theory and Experiment P04006* (2011).

[70] LUPYAN, G., AND DALE, R. Language structure is partly determined by social structure. *Plosone 5* (2010), e8559.

[71] LYON, C., NEHANIV, C. L., AND CANGELOSI, A., Eds. *Emergence of Language and Communication.* Lecture Notes in Computer Science. Springer Verlag, Berlin, 2007.

[72] MALENFANT, J., JACQUES, M., AND DEMERS, F. N. A tutorial on behavioral reflection and its implementation. In *Relfection '96 Conference* (1996).

[73] MARQUES, T., AND BEULS, K. Evaluation strategies for computational construction grammars. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (Osaka, Japan, December 2016), The COLING 2016 Organizing Committee, pp. 1137–1146.

[74] MIYAGAWA, S., BERWICK, R., AND OKANOYA, K. The emergence of hierarchical structure in human language. *Frontiers in Psychology* (2013).

[75] NOWAK, M. A., AND KRAKAUER, D. C. The evolution of language. *Proceedings of the National Academy of Sciences of the United Stated of America 96*, 14 (July 1999), 8028–8033.

[76] NOWAK M.A., PLOTKIN, J., AND JANSEN, V. The evolution of syntactic communication. *Nature 404* (2000), 495–498.

[77] OHLSSON, S. Restructuring revisited: Ii. an information processing theory of restructuring and insight. *Scandinavian Journal of Psychology 25* (1984), 117–129.

[78] OLIPHANT, M., AND BATALI, J. Learning and the emergence of coordinated communication. *The newsletter of the Center for Research in Language 11*, 1 (1997).

[79] PAUW, S., AND HILFERTY, J. The emergence of quantifiers. In *Experiments in Cultural Language Evolution*, L. Steels, Ed. John Benjamins, Amsterdam, 2012.

[80] POLLARD, C., AND SAG, I. A. *Head-Driven Phrase Structure Grammar.* University of Chicago Press, Chicago, 1994.

[81] SCHREIWEIS, C., BORNSCHEIN, U., BURGUIÈRE, E., KERIMOGLU, C., SCHREITER, S., DANNEMANN, M., GOYAL, S., REA, E., FRENCH, C. A., PULIYADI, R., GROSZER, M., FISHER, S. E., MUNDRY, R., WINTER, C., HEVERS, W., PÄÄBO, S., ENARD, W., AND GRAYBIEL, A. M. Humanized foxp2 accelerates learning by enhancing transitions from declarative to procedural performance. *Proceedings of the National Academy of Sciences 111*, 39 (2014), 14253–14258.

[82] SIERRA-SANTIBÁÑEZ, J. An agent-based model of the emergence and transmission of a language system for the expression of logical combinations. *AAAI 2015* (2015), 492–500.

[83] SMITH, K., KIRBY, S., AND BRIGHTON, H. Iterated learning: A framework for the emergence of language. *Artificial life 9*, 4 (2003), 371–386.

[84] SOLE, R., AND SEOANE, L. Ambiguity in language networks. *Linguistic Review 32*, 1 (2015), 5–35.

[85] SPRANGER, M. *The Evolution of Grounded Spatial Language*. PhD thesis, Vrije Universiteit Brussel, 2011.

[86] STEELS, L. A self-organizing spatial vocabulary. *Artificial Life Journal 2*, 3 (1995), 319–332.

[87] STEELS, L. Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. In *Approaches to the Evolution of Language: Social and Cognitive bases* (1997), J. Hurford, C. Knight, and M. Studdert-Kennedy, Eds., Edinburgh: Edinburgh University Press.

[88] STEELS, L. Iterated learning versus language games. two models for cultural language evolution. In *Proceedings of the International Workshop on Self-Organization and Evolution of Social Behaviour* (Monte Verita, Ascona, Switzerland, September 2002), C. Hemelrijk, Ed.

[89] STEELS, L. Evolving grounded communication for robots. *Trends in Cognitive Sciences 7*, 7 (2003), 308–312.

[90] STEELS, L. Language re-entrance and the ‘inner voice’. *Journal of Consciousness Studies 10*, 4-5 (2003), 173–185.

[91] STEELS, L., Ed. *Design Patterns in Fluid Construction Grammar*. John Benjamins, Amsterdam, 2011.

[92] STEELS, L., Ed. *Experiments in Cultural Language Evolution*. Amsterdam: John Benjamins, 2012.

[93] STEELS, L. Self-organization and selection in language evolution. In *Experiments in Cultural Language Evolution*. Amsterdam: John Benjamins, 2012, pp. 1–37.

[94] STEELS, L. Basics of Fluid Construction Grammar. *Constructions and Frames* (2015).

[95] STEELS, L. Do languages evolve or merely change? *Journal of Neurolinguistics 43*, Part B (2017), 199 – 203.

[96] STEELS, L., AND BELPAEME, T. Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences 28*, 4 (2005), 469–489.

[97] STEELS, L., AND GARCIA-CASADEMONT, E. Ambiguity and the origins of syntax. *Linguistic Review* (2015).

[98] STEELS, L., AND GARCIA-CASADEMONT, E. How to play the syntax game. In *European Conference on Artificial Life (ECAL 2015)* (2015).

[99] STEELS, L., AND GARCIA-CASADEMONT, E. The origins of recursive grammar. *In Preparation* (2018).

[100] STEELS, L., AND HILD, M., Eds. *Language Grounding in Robots*. New York: Springer-Verlag, 2012.

[101] STEELS, L., AND KAPLAN, F. Bootstrapping grounded word semantics. In *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, T. Briscoe, Ed. Cambridge University Press, Cambridge, 2002, pp. 53–73.

[102] STEELS, L., AND LOETZSCH, M. Perspective alignment in spatial language. In *Spatial Language and Dialogue*, K. R. Coventry, T. Tenbrink, and J. A. Bateman, Eds. Oxford University Press, 2008.

[103] STEELS, L., AND LOETZSCH, M. Babel: a tool for running experiments on the evolution of language. In *Evolution of Communication and Language in Embodied Agents*, S. Nolfi and M. Mirolli, Eds. Springer-Verlag, Berlin Heidelberg, 2010, pp. 307–313.

[104] STEELS, L., AND LOETZSCH, M. The grounded naming game. In *Experiments in Cultural Language Evolution*, L. Steels, Ed., vol. 3 of *Advances in Interaction Studies*. John Benjamins, 2012, pp. 41–59.

[105] STIVERS, T., ENFIELD, N. J., BROWN, P., ENGLERT, C., HAYASHI, M., HEINEMANN, T., HOYMANN, G., ROSSANO, F., DE RUITER, J. P., YOON, K.-E., ET AL. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences 106*, 26 (2009), 10587–10592.

[106] TINBERGEN, N. On aims and methods of ethology. *Zeitschrift für Tierpsychologie 20*, 4 (1963), 410?433.

[107] TOMASELLO, M., CARPENTER, M., CALL, J., BEHNE, T., AND MOLL, H. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and brain sciences 28*, 05 (2005), 675–691.

[108] VAN TRIJP, R. *Analogy and Multi-Level Selection in the Formation of a Case Grammar. A Case Study in Fluid Construction Grammar*. PhD thesis, Universiteit Antwerpen, 2008.

[109] VERHAGEN, A. Syntax, recursion, productivity – a usage-based perspective on the evolution of grammar. In *Evidence and Counter-Evidence. Festschrift F. Kortlandt*, A. Lubotsky, J. Schaeken, and J. Wiedenhof, Eds., vol. 1 of *SSGL 32*. Rodopi, Amsterdam, 2006.

[110] VERHOEF, T. *Efficient coding in speech sounds: Cultural evolution and the emergence of structure in artificial languages*. PhD thesis, University of Amsterdam, 2013.

[111] VOGT, P., AND COUMANS, H. Investigating social interaction strategies for bootstrapping lexicon development. *Journal of Artificial Societies and Social Simulation 6*, 1 (1 2003).

[112] WELLENS, P. *Adaptive Strategies in the Emergence of Lexical Systems*. PhD thesis, Vrije Universiteit Brussel, 2012.

[113] WELLENS, P., AND LOETZSCH, M. Multi-dimensional meanings in lexicon formation. In *Experiments in Cultural Language Evolution*, L. Steels, Ed., vol. 3 of *Advances in Interaction Studies*. John Benjamins, 2012, pp. 143–166.

[114] WITTGENSTEIN, L. *Philophische Untersuchungen*. Suhrkamp, Frankfurt, 1967.

[115] XU, Y., AND REGIER, T. Numeral systems across languages support efficient communication: From approximate numerosity to recursion. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society* (2014), P. B. et al., Ed.

[116] ZIPF, G. K. *Human behaviour and the principle of least-effort*. Cambridge, MA: Addison-Wesley, 1949.

Index