

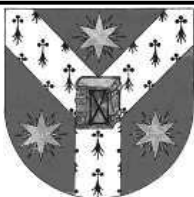


**Learning to unlearn in lattices of concepts:
A case study in
Fluid Construction Grammars**

Liviu Ciortuz, Vlad Saveluc

TR 11-01, October 2011

ISSN 1224-9327



Learning to unlearn in lattices of concepts: A case study in Fluid Construction Grammars

Liviu Ciortuz, Vlad Saveluc
Department of Computer Science
“Al.I. Cuza” University, Iași, Romania

Abstract

This report outlines a couple of lattice-based (un)learning strategies proposed in a recent development of unification-based grammars, namely the Fluid Construction Grammar (FCG) setup.

These (un)learning strategies are inspired by two linguistic phenomena occurring in a dialect spoken in the Banat area of Romania. Children from that region — where influences produced over centuries by Serbian, a Slavic language, are obvious — learn in school the modern Romanian language, which is a Romance language.

This particular setup offers us the possibility to model in FCG a two-step learning process: the first step is that of learning a (perfective) verbal aspect similar to the one already presented by Kateryna Gerasymova in her MSc thesis, while the second one is concerned with un-learning (or, learning another linguistic “construction” over) this verbal aspect. Thus, the interesting issue here is how learning could continue beyond learning the verbal aspects. We will first give linguistic facts, after which we will outline the way in which FCG could model such a linguistic process.

From the computational point of view, we show that the heuristics used in this grammar repairing process can be automatically derived since the meanings associated to words and phrases are organized in a lattice of feature structures, according to the underlying constraint logics.

We will later discuss the case of another verbal marker in the dialect spoken in Banat. It will lead us to sketch a composite, quite elaborated (un)learning strategy.

1 Introduction

Construction Grammars [12] are unification-based grammars [17] that emphasize the strong link between syntax and semantics throughout the parsing and generation/production process. Several versions of Construction Grammars exist: Berkeley Construction Grammar [9], Embodied Construction Grammar [2], Radical Construction Grammar [8], Sign-Based Construction Grammar [16], and Fluid Construction Grammar [21]. The last of them, FCG, benefits from a nice implementation and experimentation platform (www.fcg-net.org/Download.html). What distinguishes FCG from other types of construction grammars is that its design centers on the idea that grammars do evolve and therefore we must be able to study the language evolution in a computational setup.

In the recently finished EU FP7 project “ALEAR” (www.alear.eu), FCG researchers used autonomous robots to study which mechanisms are needed in order to make these artificial agents develop certain traits of language (possibly resembling certain phenomena in the human language). Several studies on the acquisition of interesting linguistic competences, based on FCG implementations, have been recently presented for German [13] [18], Russian [11] and Hungarian [3]. Among these studies, the closest to our work is the one due to Gerasymova [11]. She studied how a rather intricate piece of linguistic mechanism — the one represented by the *aspects* of perfective verbs in Russian, in which the meaning of a past action is further determined by a prefix added to the corresponding verb —, can be learned through so-called language games [19] [20] played by artificial agents.

In a *linguistic game*, two agents having different grammars face the same environment. One agent is the teacher and the other is the learner. There is no grammar exchange between agents; the learner should acquire new linguistic competence only through communication with the teacher. When communication ends in failure (i.e. non-understanding), certain grammar repair/extension procedures are triggered for/in the learner. If this process is successful, it amounts to learning.

This paper outlines a couple of learning strategies proposed in the FCG (in fact the FCGLight) setup based on us-

ing lattices of concepts (meanings of words and phrases).¹ These strategies were derived starting from the analysis of a linguistic evolution phenomenon in the Romanian language.

The Romanian language has a regional dialect (“grai”, in old Romanian) which is spoken in Banat, a region situated in the western part of Romania. This dialect is called “*graiul bănăţean*”, henceforth abbreviated as GB. More precisely, in this paper we will refer to the dialect spoken in southern part of Banat, known as “Banatul sârbesc” (the Serbian Banat) since it shares a border with Serbia. As expected, the Romanian dialect spoken there, was influenced over the centuries by the Serbian language.^{2 3}

The organization of this paper is as follows: Section 2, which is the largest one in this work, will present an interesting strategy for learning one more level beyond the learning (à la Gerasymova [10]) of the ‘do’ prefix that marks a perfective aspect of verbs in GB. What makes it interesting from the computational learning point of view is that this strategy has a heuristical backbone which can be directly inferred from the lattice of feature structures that represent the meanings produced in FCG during so-called language games [19] [20].

Section 3 contains supplementary linguistic facts about the ‘do’ prefix in “*graiul bănăţean*”.

In Section 4 we will show that the prefix ‘pro’ which occurs in a number of verbs in modern Romanian — but has a different role and meaning than the prefix ‘pro’ of verbs in “*graiul bănăţean*” — is un-learnable using Gerasymova’s strategy. The explanation is that ‘pro’-prefixed verbs in modern Romanian do not exhibit the (simple) homologous compositionality of form and meaning assumed by Gerasymova. However, the prefix ‘pro’ in modern Romanian is learnable (over the prefix ‘pro’ used by GB-speaking people) by using an interesting mixture of simpler strategies. Finally, Section 5

¹Machine learning algorithms working on lattices of terms in the propositional logic are presented in [14].

²“*Graiul bănăţean*” was also influenced (mostly at the lexical level) by the Hungarian language — since the principality of Transylvania, which included the Banat region, belonged for a long time to the Hungarian crown — and also more recently by the German language, since Transylvania was incorporated into the Austro-Hungarian empire. Transylvania became part of modern Romanian in 1918, at the end of the First World War.

³The first author of this paper learned this dialect during his childhood, when he spent his summer vacations in Banat.

will summarize the work here presented and draw the conclusions, while Section 6 will propose (as possible further work) an interesting extension to the learning procedure here presented.

Other interesting linguistic phenomena, together with FCG implementations, have been recently presented for German [13] [18], Russian [11] and Hungarian [3].

2 The case of the ‘do’ aspect marker

Due to the Serbian influence, “*graiul bănăţean*” exhibits the use of the Slavic prefix ‘do’ [22], which does not exist in modern Romanian (henceforth designated MR). This prefix marks a perfective aspect of verbs in a similar way to the Russian verb aspects as presented by Gerasymova [10].

2.1 Linguistic facts

When ‘do’ is attached as a prefix to a verb in GB, it indicates (like in Russian [22]) that the action to which the speaker is referring is/was well under way, but not yet terminated. Moreover, according to our empirical observations, ‘do’ is mainly (but not exclusively) used in the negative *form* of the perfect *tense* in the indicative *mood*,⁴ and in the *affirmative form* of the *imperative mood*.

For instance:

- i. [Eu/Noi] N-am domâncat. (GB)
[Eu/Noi] N-am terminat de mâncat. (MR)
I/We haven’t yet finished to eat. (EN)
- ii. Domânăncă! (GB)
Termină de mâncat! (MR)
Finish to eat! (or, Finish your meal!) (EN)
- iii. Lasă-mă să domânănc! (GB)
Lasă-mă să termin de mâncat! (MR)
Please let me finish to eat! (EN)

Note: In the *iii* example, the particle ‘să’ is the subjunctive marker (that there precedes the first person singular form) of the verb ‘a mânca’, which means ‘to eat’. ‘A’ is the marker of the infinitive mood in Romanian. We also

⁴Note that the ‘do’ preposition can be attached to verbs in any tense of the indicative mood, not only in the case of perfective tense.

like to inform the reader that the Romanian language is highly inflectional, therefore the personal pronouns ('eu', 'noi' etc) are usually omitted when playing the subject role; they can be easily inferred from the finite form of the verb which expresses the predicate.

It is not difficult to see that the 'do' preposition used in GB can be encoded and/or learned in FCG in a similar way to the prepositions that mark different verb aspects in Russian (see [10] and [6]).

Our *goal* here is to show how, after having encoded or learned in FCG a grammar for a fragment of GB comprising the preposition 'do', it can subsequently evolve. Actually, we will sketch a learning process at the end of which the role played by the preposition 'do' will be relegated to a certain type of phrasal construction.

The design of this *learning strategy* is inspired by the linguistic facts. Namely, when entering into contact with the modern Romanian (MR) language — for instance when going to school as children or, in the last decades, by listening to the media —, the inhabitants of the Serbian Banat region learn an alternative way of the use of the 'do' preposition. The reason why they have to do this is natural: MR, a Romance language, lacks this preposition.⁵⁶ Concretely, the 'do'-replacement mechanism amounts to using in MR a phrase which is composed mainly of

- the verb 'a termina' (to finish) or one of its synonyms ('a sfârși', 'a finaliza')
- the (non-finite) *supine* form of the verb to which (in GB) the 'do' marker was attached.

As a matter of illustration, the reader should see the correspondence between the GB and MR versions of the *i*, *ii* and *iii* examples given above. There, the preposition 'de' is the marker of the supine mood, thus 'de mâncat' is the supine form of the verb 'a mânca' (to eat).

⁵The Banat people completely renounce to using 'do' when moving outside their region.

⁶Moreover, there is no other preposition in MR that corresponds to the preposition 'do' in GB. We will show in the Section 4 that another preposition used in GB, 'pro', which has also a perfective(-related) role is replaced in certain cases by a preposition ('re') which was imported in modern Romanian via its Latin roots. We mention that those Latin roots of Romanian were heavily emphasized/revived through a large-scale cultural movement that took place in the 18th and 19th centuries.

2.2 Conceptualizing the learning strategy

Figure 1 shows the simple *transformation rule* that synthetically expresses what children raised in the Serbian Banat area (supposedly) learn when "writing" modern Romanian in school (or by listening to the media) "over" the dialect (GB) that they usually use at home.⁷ The supine form itself is generally obtained (with very few exceptions) by attaching the preposition 'de' to the masculine singular form of the verb's participle. For instance, the supine of the verb 'a mânca' (to eat) is 'de mâncat'.⁸

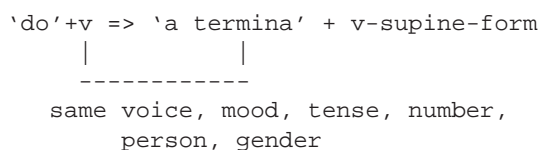


Figure 1: The (general) transformation rule that governs the replacement of the 'do' (GB) aspect marker when learning MR.

Here follows an example of application of the transformation rule shown in Figure 1:

- i'*. [Eu] Domănânc. (GB)
- [Eu] Termin de mâncat. (MR)
- I am finishing to eat. (EN)
- (or: I am finishing my/the meal. (EN))

We note that the example *i'* is the indicative, present tense, singular number, affirmative form of the example *i*.

2.3 Operationalization of learning, using FCG

While not being quite straightforward, it is not very difficult either to design a *strategy for learning* in the FCG setup. We base our strategy firstly on exploiting the double correspondence between two different forms (GB and

⁷When the first author occasionally in goes vacation in that region, he uses to play language games with children, by emphasizing the correspondence between the GB and MR forms of simple sentences like those in the examples *i*, *ii*, *iii* and *i'*. Such language games are based on the transformation rules similar to the one shown in Figure 1.

⁸Another supine marker is the preposition 'la', but there above only the marker 'de' can be used.

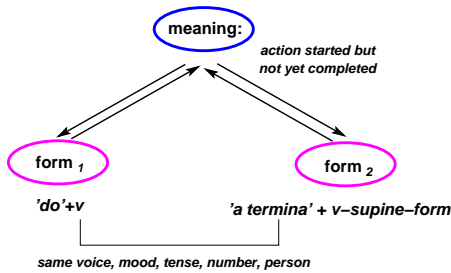


Figure 2: The case of the ‘do’ aspect-like marker: the double correspondence between meaning and two forms (one in GB, the other in MR). Here above, the verb v is assumed, for simplicity reasons, to be intransitive.

MR) and one meaning that may result from language games (see Figure 2), and secondly on integrating these (and other) meanings in a lattice of FSs.

The association of one meaning to two different forms, only one of which is acceptable/parsable by the learner’s grammar, may easily show up during language games. Such a *conflict situation* is meant to trigger at the learner agent a grammar repair/extension process. Indeed, the *aim* of our strategy is to modify the learner’s grammar so as to be able to parse the second (MR) form.

2.3.1 The Learner’s Initial Grammar

Before presenting the *core procedure* of our proposed *grammar learning strategy*, let us state that — besides the two types of constructions necessary to parse/produce sentences like “Domănânc”, namely the FCG constructions for (a.) ‘do’ and (b.) ‘mănânc’⁹ — the *learner’s starting grammar* should already contain the following three (sets of) constructions:

c. A lexical construction l_1 for each inflection of the mono-transitive verb ‘a termina’ (to finish). This verb takes as its direct object a noun phrase. The *meaning* associated to an inflected form of ‘a termina’, for instance ‘mănânc’ (active voice, indicative mood, present tense, first person, singular number) is

⁹These two constructions are very much similar to those presented for ‘na’ and respectively ‘risova’+‘/la’ in Chapter 3 of Gerasymova’s MSc thesis [10].

```
((finish ?event)(agent ?event 1s)
(object ?event ?object)) (t)
```

In the LIGHT (and FCGLight) notation [5] — that is close to the notation used in OSF-logic [1] and HPSG [15] —, t is written as the FS

```
#event:finish[ agent 1s,
object #object]
```

d. A phrase construction (r) that combines two types of constructions: one for an inflected form of ‘a termina’, and one for a noun phrase, thus enabling the learner to process phrases like “Termin lucrul” (meaning: I am finishing the work).

e. A construction l_2 that, given the lexical construction for the base form of a verb, derives its *supine* form.¹⁰ The *meaning* associated to the supine, for instance ‘de măncat’ is

```
((eat ?event-1)(agent ?event-1
?agent)) (s)
```

which in LIGHT notation corresponds to the FS

```
#event-1:eat[ agent #agent].
```

Note that in the t and s sets of meaning constraints, the variables $?object$ and $?agent$ are uninstantiated.

Figure 3A offers a synoptic view for the resulting correspondence between the *form* constraints and units on one side and the *meaning* constraints on the other side, when parsing/producing the sentence “Domănânc” using the source grammar (GB).

2.3.2 Specifications for the Core Learning Procedure: Input and Output

The *input* of our core learning procedure should be given as $(m, (form_1, form_2))$, where

– m is a set of constraints expressing a meaning,¹¹ like:

```
((eat ?event-2)(agent ?event-2 1s)
(event-type ?event-2 under-completion)) (m)
```

¹⁰Learning or inventing the supine mood in FCG is not the subject of this paper.

¹¹More exactly, the meaning associated to a certain configuration of the environment in which the respective language game takes place.

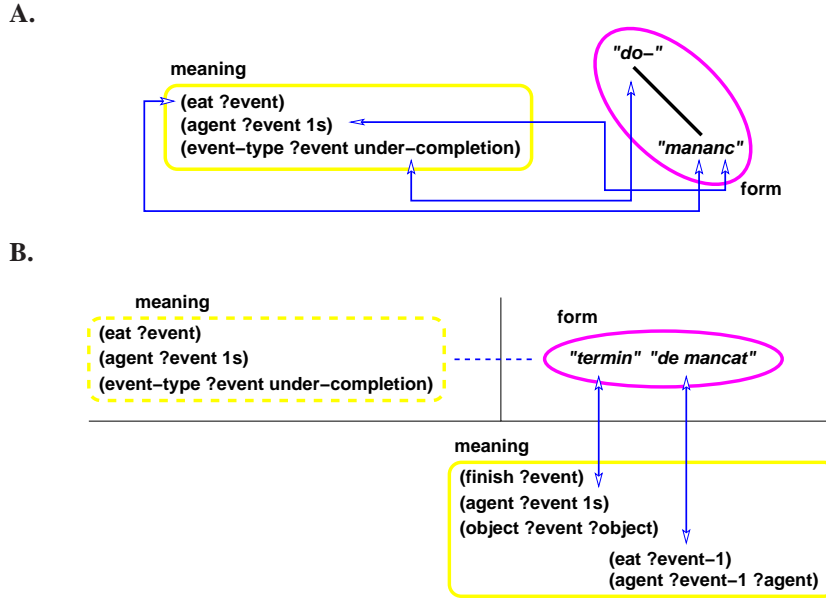


Figure 3: A schematic view on the compositional correspondence between form and meaning when parsing/producing the sentences (A.) “Domăănânc” and (B.) “Termin de mâncat” using the input grammar. In fact, the figure B. shows an improved correspondence, which would be obtained using a holophrasis construction, as suggested by Gerasymova in Chapter 4 of her MSc thesis [10].

which in LIGHT notation corresponds to

#event-2:eat[agent 1s,
event-type under-completion];

– $(form_1, form_2)$ is a pair of forms strings¹² associated to the meaning m — in our case $form_1$ is “domăănânc”, and $form_2$ is “termin de mâncat” — such that $form_1$ is parsable by the learner’s grammar and the generated meaning is m , while $form_2$ has not this property.

Figure 3B shows the meaning produced after parsing the sentence “Termin de mâncat” using the input/“source” (GB) grammar, as assumed above. Note that in this case the parsing process is only partially completed, since the verb `termin` accepts only a noun phrase as complement. Moreover, the created meaning differs from the intended meaning (indicated in the upper left dotted case in Figure 3B).

¹²For the sake of simplicity, we do not use the `meets` constraint-based version used in FCG coding in order to express strings like “termin de mâncat”.

Someone who is acquainted with the learning strategy proposed by Gerasymova could perhaps ask himself whether using holophrastic constructions could help us obtain the target/“output” grammar. Following such an approach, a holophrasis would basically replace the meaning m' produced by the source grammar (see below) with m , the meaning indicated by the teacher. Figure 3B illustrated this (unfortunately, rough) correspondence between `form` and `meaning` based on the holophrasis usage. A possible further step would be to generalize over such holophrases with respect to the verb itself and its mood, tense, person and number features. However, the fact that this approach does not contains in itself the indices for how to affect the source grammar itself, leads us to consider it an (almost) impossible path to follow.

We will propose another approach, in which the way to do the necessary changes in the grammar will emerge naturally. This approach is based on using lattices of concepts (here FSs) expressing meanings and constructions, like in FCGlight that uses OSF-logic to express feature

constraints [1].

The *output* of the core learning procedure is a new (repaired/evolved) version of the learner’s grammar that is able to parse $form_2$ so as to generate/produce m , and vice versa.

2.3.3 Illustrating the Idea of the Core Learning Procedure

Considering m , $form_1$ and $form_2$ instantiated as above, the *core* of the *learning procedure* itself, works as follows:

Parsing the string $form_2$ fails (or: does not terminate) because the application of the rule r is blocked. Nonetheless, after applying the constructions for ‘termin’ (to finish, first person, singular, present tense) and ‘de mâncat’ (to eat, supine), the parsing process has produced the meaning m' which is the union of (t) and (s) seen as sets of constraints:

```
((finish ?event)(agent ?event 1s)
(object ?event ?object)
(eat ?event-1)(agent ?event-1 ?agent))
(m')
```

The *question* one may ask is: What could the learner do starting from this point, and knowing that he/she should have produced the meaning m (instead of m')? The *answer* to this question is provided below:

By *comparing* the meanings m and m' , the learner will impose certain transformations on m' — and subsequently/backwardly on the constructions l_1 (for ‘termin’) and the phrase construction r — such that $form_2$ become fully parsable and generate the meaning m .

But how is this *comparison* concretely performed? The *idea* is to use the *subsumption* (i.e. generalization), or its opposite (specialization) relation among FSs as presented for instance by [1] [4].¹³

As such, the FSs m and m' are not comparable w.r.t. the subsumption relation (here denoted \sqsupseteq), i.e. neither $m \sqsupseteq m'$ nor $m' \sqsupseteq m$.

However, one could make the following *transformations* on m' :

(1) *generalization*: Eliminating the *finish* predicative constraint on the variable $?event$, and the *object* feature constraint on the same variable — due to the fact that the two constraints are not found in m — the result (denoted m'') is much closer to m .

```
((agent ?event 1s)
(eat ?event-1)(agent ?event-1 ?agent))
(m'')
```

The *problem* now is that m'' is not a single-rooted FS (unlike m), due to the fact that it contains two *event*-sorted variables, namely $?event$ and $?event-1$.

(2) *specialization*: Imposing the equation constraint on these two variables, yields a single-rooted FS:

```
((agent ?event-1 1s)(eat ?event-1)
(agent ?event-1 ?agent))
(m_1)
```

which in LIGHT notation is expressed (simpler!) as:

```
#event-1:eat[ agent #agent:1s].
```

Moreover, now $m_1 \sqsupseteq m'$, i.e. m_1 is strictly more general than m' .

(3) *specialization*: Adding the one “missing” constraint (*event-type ?event-1 under-completion*) to m_1 makes it become exactly m (up to variable renaming)!

These three operations on the meaning constraints entail the following three *transformations* on the constructions t and r (see the above *c.* and *d.* prerequisites for the input grammar to the learning procedure):

(1') Eliminate the two constraints (*finish ?event*) and (*object ?event ?object*) from t , the meaning of the the verb ‘a termina’(or, better, from a copy of this verb’s construction). Then modify this verb’s valence list so as to accept supine verbs instead of (only) noun phrases.

(2') Using the valence list of the verb ‘a termina’, make the rule construction r (better, a copy of it) equate the two event variables, one being the agent value for the verb ‘termin’, the other the agent value for the supine verb.

(3') Make the rule r impose the constraint (*event-type ?event-1 under-completion*) on the event corresponding to the supine verb.

¹³We have already shown [6] that learning FCG grammars could be seen (in FCGLight) as a search process in a space of grammar versions, using the subsumption relation.

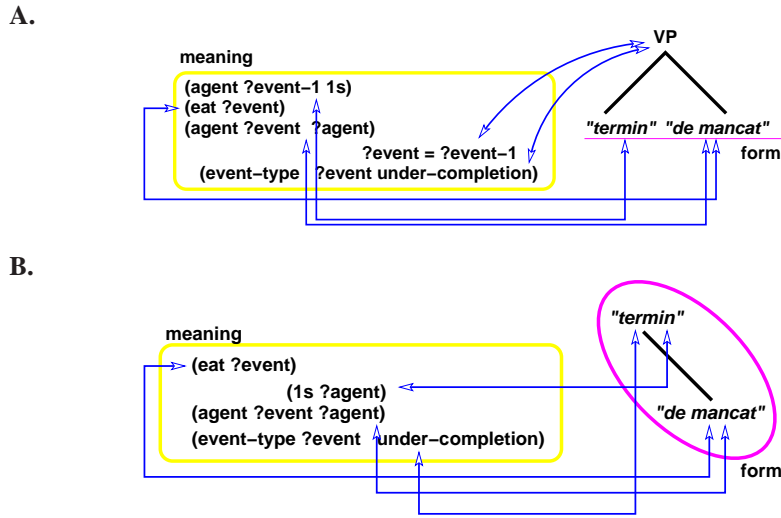


Figure 4: A schematic view on the compositional correspondence between form and meaning when parsing/producing the sentence “Termin de mâncat” with the learnt/“target” grammar. In the first version of the learnt grammar (A.) the constructions for the phrase (r) and the verb `termin` are affected. In the second version (B.) only the `termin` construction is affected. Our paper elaborates the A. alternative.

To sum up, the transformations (1’), (2’) and (3’) induced from the constructions found in the learner’s initial grammar (G) have solved our problem. Now, $form_2$ is parsable by G' (which denotes G altered by (1)–(3)), and the generated meaning is m , as desired.

Important note: An alternative to the points (2’) and (3’) from above would be to let the rule r unchanged and apply those operations only on the (copy) construction for the verb ‘termin’.¹⁴ Figures 4A and 4B graphically show the correspondence between `form` and units on one side and meaning constraints on the other side for the two versions of the target grammar (MR) whose learning was presented above.

Finally, a number of *generalizing steps* will complete the learning process that corresponds to acquiring the transformation rule presented in Figure 1:

- generalization over the voice, mood, tense, number and person of the verb;
- generalization over the verb itself (and its valence list);

¹⁴In fact, two other versions are possible, in which the r (VP) rule retains one of the two indicated meaning constraints, while the remaining constraint is passed to the `termin` construction.

– the verb ‘a termina’ (better: a new version/copy of it) will have the valence list defined as `v-supine-form + v-valence-list`,¹⁵

– the ‘perfective’ property will be deleted from the lexical description of those verbs in GB that accept ‘do’ as prefix.

Now, *revisiting* the steps (1) (2) and (3) from above, we could give one more (somehow higher level) justification of *why* we proceeded the way we did.

The FS lattice that one would get by representing m , t , s , m' and $m'' = \text{LUB}(t, s)$ and the subsumption relationships among them is shown in Figure 5. LUB is abbreviation for the *least upper bound relation* among FSs. We note that Figures 5 and 6 employ LIGHT (and FCGLight)

¹⁵In FCG this change would also require the modification of the rule that governs the combination of the verb ‘a termina’ with a list/set of complements.

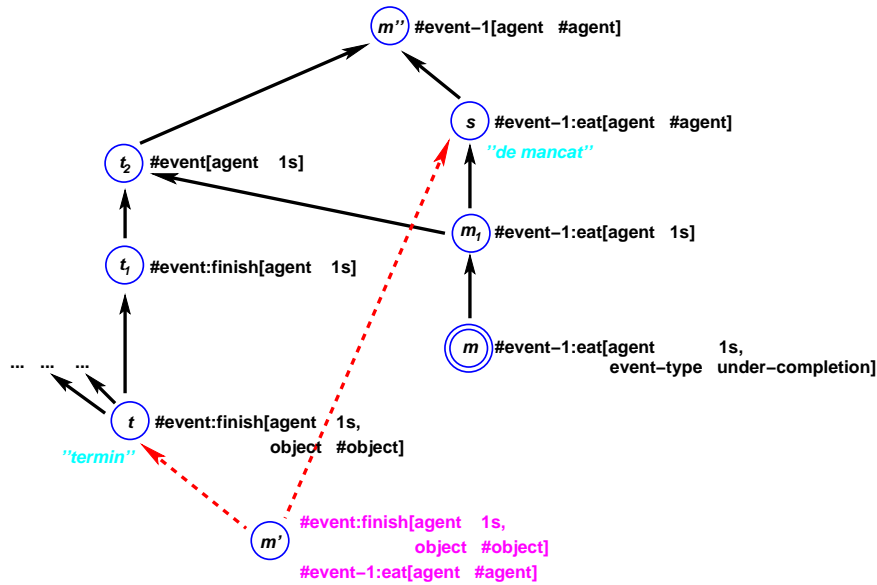


Figure 5: The lattice of meaning FSs for the sentence “Termin de mâncat”. The FS t encodes the meaning of the verb ‘a termina’ (to finish), while s corresponds to ‘a mânca’ (to eat). Denoted by m is the meaning associated to the given sentence; it has to be obtained (via parsing) by the learnt/output grammar.

notation.^{16 17}

¹⁶In Figure 5 the dotted arrows correspond to *generalized subsumption*, seen as a relationship among possibly multi-rooted feature structures. In this acceptance, one could view $m' = \text{GLB}(s, t)$, where GLB denotes the greatest lower bound among possibly multi-rooted feature structures. Also, $m'' = \text{LUB}(m, m')$.

¹⁷Instead of considering generalized LUB/GLB relationships between multi-rooted FSs, one could decide to stick — at least for examples whose complexity compares to the present example — to single-rooted FSs, on which to apply the basic (i.e. non-generalized) LUB/GLB operations. In this case, for our example the following discussion comes into framework.

If m' is to be put under the form of a single-rooted feature structure, the two *possibilities* are:

- either to instantiate the variable $?event$ to $?agent$ or
- to instantiate the variable $?object$ to $?event-1$.

Sort constraints in FCGlight — or, in FCG, predicative constraints — on these variables should block the first of these two possibilities. The second possibility — moreover, assuming that sort/predicative constraints on $?object$ and $?event-1$ are relaxed (which is acceptable), or alternatively assuming that the constraint on the verb’s valence in the rule r is generalized (again acceptable) — leads to a single-rooted FS version of m' which obviously differs from m . The resulting rooted FS, in LIGHT notation is:

#event:finish[agent 1s,

Interestingly, this lattice shows that in order to get the meaning m in the target grammar (for the sentence “Termin de mâncat”) there is no need to affect s , the meaning produced by the supine; only the meaning associated to the verb ‘termin’ must be changed.

As shown in the annotated version of this lattice shown in Figure 6, each directed arc in the lattice corresponds to eliminating a certain elementary constraint. Conversely, each downward dotted arrow corresponds to the introduction of an elementary constraint. As a consequence, one can “read” in the lattice a *path* to follow from t to m . In fact there are two such paths. For convenience, we chose the shortest one of them. When we presented above the

object #event-1:eat[agent #agent].

One further step would be to coreferenciate the to agent roles, thus getting the FS

#event:finish[agent #agent:1s,
object #event-1:eat[agent #agent]].

However, subsequent modifications to the grammar so as to get the desired result would be rather complicated, or at least more complicated than the solution that will be built in the sequel.

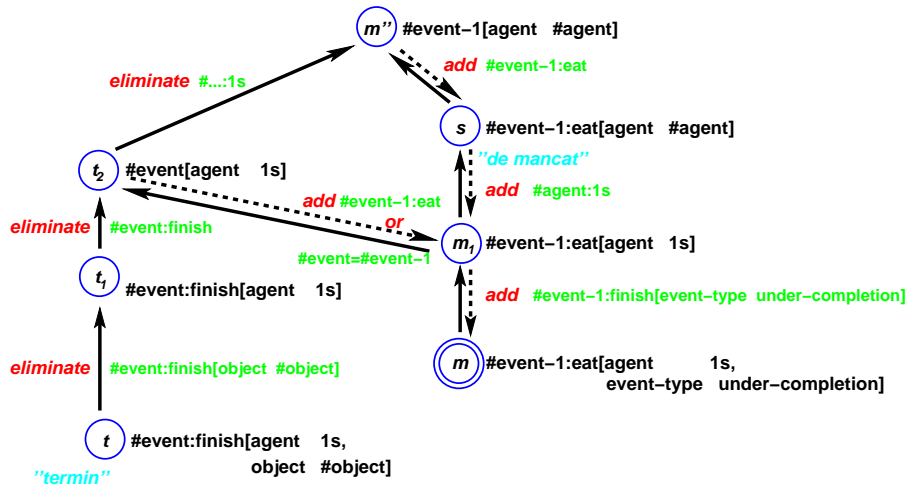


Figure 6: Operationalizing the search for the target meaning by walking on the path(s) from t to m in the lattice of meaning FSs (shown in Figure 5). Each arc traversed on such a path corresponds to adding or removing an elementary constraint.

steps (1'), (2') and (3'), we in fact obtained the *procedural* version of “reading” this path.

2.3.4 The Learning Procedure

Here we *summarize* our proposed grammar learning strategy:

1. Collect (*meaning*, (*form*₁, *form*₂)) pairs during the language game.
2. Given one such pair, organize the lattice of meaning FSs (the target, source, LUB, and in between FSs) based on the subsumption/generalization relation between FSs.
3. “Read” the path(s) in this lattice between the source and the target meaning FSs as transformations — i.e. either pushing or eliminating elementary constraints — in the encompassing constructions in the source grammar.
4. Generalize the resulting construction(s) if/whenever it is possible.

3 Other linguistic issues related to the ‘do’ marker in GB

This is a section oriented toward presenting more linguistic data related to the ‘do’ marker in “graiul bănațean”. Therefore, the reader who is only interested in FCG-based strategies for grammar repairing/evolution could skip this section and proceed directly to the next one.

3.1 The case of ‘do’ attached to ‘a găta’ (GB)

There is one remarkable case in which the translation of the ‘do’ particle form GB into modern Romanian works under a rule which is different from the general one introduced in Figure 1. This is the case of the verb ‘a găta’ which is an equivalent of the verb ‘a termina’ (MR) and ‘to finish’ (EN). When the ‘do’ prefix is attached to one of the forms of the verb ‘a găta’, it simply lets its meaning unchanged. However, it seems to us that it adds a sense of closeness in the case of past tense and a sense of urgency when the future tense is used.

‘do’ + ‘a găta’ (GB)

= 'a termina [imediat]' (MR)
 = 'to finish [immediately]' (EN)

For example:

- iv. [Eu/Noi] Am dogătat lucrul. (GB)
 [Eu/Noi] [Tocmai] Am terminat lucrul. (MR)
 I/We have [just] finished the/our work. (EN)

If we were applying the (GB-to-MR) transformation rule introduced in Figure 1, we would get a sentence that is not acceptable in MR:

- v. (*) [Eu/Noi] Am terminat de terminat lucrul. (MR)
 I/We have finished to finish the work. (EN)

When it is used in the imperative voice, the GB verb 'a dogăta' conveys even more evidently the request of finishing immediately the action:

- vi. Do-gată treaba/lucrul! (GB)
 Termină [imediat] treaba/lucrul! (MR)
 [You have to] Finish [immediately] your work! (EN)

Finally, we note that the regional verb 'a găta' was retained in MR only as the passive form: 'a fi gata' (to be ready).

3.2 The case of 'do' attached to participles

There is one more special case, when 'do' is attached as a prefix to the participle form of a verb. For instance 'doreparat' (GB) is translated as 'reparat în totalitate' (MR) and 'completely repaired' (EN). As an exception, the past participle form of the verb 'a dogăta', namely 'dogătat' or 'dogata' (GB) is translated simply as 'terminat'/'gata' (MR) = 'finished' (EN).

For example:

- vii. [Lucrul/el] Nu-i dogătat. (GB)
 [Lucrul/el] Nu-i [încă / complet] gata. (MR)
 [The work / It] is not yet/completely finished. (EN)

In general, the transformation rule for 'do'+ v-past-participle could be expressed as:

'do'+v-participle => [complet]
 v-participle

while its negative counterpart is

'ne'+ 'do'+v-participle => [încă]
 'ne'+v-participle /

[incomplet] + v-participle

For instance,

- viii. [Eu/Noi] Am lăsat mașina nedoreparată. (GB)
 [Eu/Noi] Am lăsat mașina incomplet reparată. (MR)
 I/We let the car not fully repaired. (EN)

3.3 The case of 'do' modified by the adverb 'mai'

There is another composed form of the past participle 'dogata': 'maidogata'. It means: aproape gata (MR), nearly finished (EN). Therefore, 'mai' acts as a modifier of 'do': almost, or nearly (finished).¹⁸

- ix. [Noi] Suntem maidogata. (GB)
 [Noi] Suntem aproape gata. (MR)
 We are nearly ready (en.)
- x. Treaba/Lucrul este maidogata. (GB)
 Treaba/Lucrul este aproape gata. (MR)
 The work is almost done. (EN)

A couple of *remarks* can be made here:

The preposition 'mai' cannot be attached to the prefix 'do' in the finite forms of verbs:

- iv'. (*) [Eu/Noi] Am maidogătat lucrul. (GB)
 vi'. (*) Maidogată treaba/lucrul! (GB)

However, the preposition 'mai' cannot be eliminated in the passive voice:

- ix'. (*) [Noi] Suntem dogata. (GB)
 x'. (*) Lucrul este dogata. (GB)

¹⁸In modern Romanian, the 'mai' adverb may indicate the repetition

[Eu/noi] Am mai vazut-o. (MR)
 I/We have seen it one more time. (EN)

or it may be a reinforcer for adverbs and adjectives, similarly to 'more' in English. It can also be used to mark an abandoned or aborted action:

[Eu/Noi] N-am mai terminat treaba. (MR)
 I/We let the work unfinished [although I/we shouldn't]. (EN)

4 The case of the ‘pro’ prefix in GB

The ‘pro’ preposition that can be attached as a prefix to different verbs in ‘graiul bănăţean’ (GB) bears the following simple meaning: the action (*a*) corresponding to the verb (*v*) has occurred at least one more time in the past.

For instance:

- xi.* [Eu] promănânc. (GB)
I am eating again. (en.)

The fact that a similar/identical action *a* has already took place in the past makes us think that the ‘pro’ preposition in GB has a perfective (or at least a perfective-like) role.

Encoding and learning the ‘pro’ preposition in FCG could be done easily, again following Gerasymova’s learning strategy.

Learning the modern Romanian language in the Serbian Banat (for instance when children go to school) implies the following *three transformations* concerning the use of the ‘pro’ preposition.

A. Similarly to the case of (un-learning) the ‘do’ preposition attached as a prefix to participles of verbs in GB, the ‘pro’ preposition may *generally* be replaced in MR by an adverb (‘iar’/‘iarăşi’/‘din nou’ = again). Thus the original verb phrase in GB gets an attached modifier in MR.

$$\text{'pro'+v + ...} \Rightarrow \text{v+'iar' + ...}$$

The place of the adverb ‘iar’ is not strict, as MR enjoys a large freedom of phrase constituents’ order.

B. In many cases (but not in general, as it was stated for the A case) one can replace ‘pro’ in GB with the preposition ‘re’ in MR. ‘Re’ originated from Latin, therefore it is quite common in MR. (We make the remark that ‘re’ is not used in GB.) For instance:

- xii.* [Eu/Noi] Am profăcut zidul. (GB)
I/We have remade/rebuilt the wall. (en.)

Simply, this transformation is expressed as:

$$\text{'pro'+v + ...} \Rightarrow \text{'re'+v + ...}$$

However, as already suggested above, not all verbs admit this simple transformation. For instance, while in GB the verb ‘a promănca’ is perfectly acceptable, the form

that it yields via the above transformation rule, namely ‘a remănca’, is perfectly understandable in MR but totally unacceptable.

Conversely, some ‘re’-prefixed verbs in MR have a meaning which is not necessarily indicating the re-iteration of an action. For instance:

- ‘a relua’ (‘a lua’ means ‘to take’) can be used with the sense ‘to take again’, but it is usually employed with the sense ‘to resume’;
- ‘a reda’ (‘a da’ means ‘to give’) is only rarely used with the sense ‘to give again’, since in general it acts like ‘to tell’ (a story), ‘to play’ (a magnetic tape) etc.

As a consequence, the above (B) rule is limited to certain verbs. As a consequence, learning its correct application implies also the necessity to learn the set of verbs to which it is applied. This means that the hierarchy/set of verbs accepting the prefix ‘pro’ in GB is over-written in MR by another hierarchy/set of verbs that accept the ‘re’ prefix, as suggested in Figure 7.

C. There is an important number of verbs (in general neologisms) in MR, which exhibit the prefix ‘pro’.¹⁹ Unlike the ‘pro’ preposition in GB (and the ‘re’ preposition in MR) which can be attached quasi-freely to many verbs expressing an action that can be repeated, one cannot use in MR a rule $v \Rightarrow \text{'pro'+v}$ for generating new verbs.

Up to our knowledge, there is no one verb in modern Romanian that accepts the prefix ‘pro’ and has the same sense that it has in ‘graiul bănăţean’. Otherwise said, it seems to us that each ‘pro’-prefixed verb that have been imported into MR and happen to have a homonym in GB has a meaning which is different from the latter’s.

For instance:

- ‘a propune’ in GB means ‘a pune din nou’ in MR and, respectively, to put [something] again [somewhere], in English,

¹⁹In general, a ‘pro’-prefixed verb has as associated meaning an action *a*, that takes place before another action *b*, the later one being usually of a different kind. (It should be noted that in the case of the Latin-rooted ‘re’ in MR or the preposition ‘pro’ in GB the two actions are identical or very similar.) The form of the ‘pro’-prefixed verb was derived by association with the verb corresponding to the subsequent action (*b*). However, further on the two verbs associated with *a* and respectively *b* have separate lives. Therefore it is possible that when transferred into another language (this was the case of MR), the second verb (*b*) might not make it into the target language.

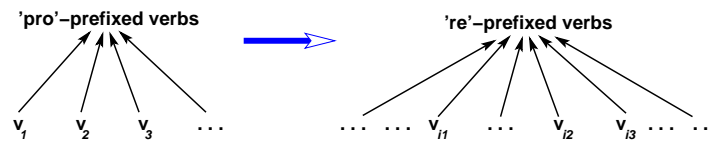


Figure 7: The hierarchy/set of verbs accepting the prefix ‘pro’ in GB is over-written in MR by another hierarchy/set of verbs that accept the ‘re’ prefix.

- ‘a propune’ in MR means to propose (or, to make a proposition) in English;
- ‘a produce’ in GB means ‘a duce din nou’ in MR and, respectively, to carry again [something, somewhere] in English,
- ‘a produce’ in MR means to produce in English.

Therefore, when a child in Serbian Banat learns such MR ‘pro’-prefixed verbs when going to school (or listening to the media) he/she has to replace a meaning (used “at home”) with another, completely different meaning used at school and in the media.

Beyond the above three cases (A, B and C), it is quite interesting to note one more issue:

D. One could ask whether the ‘pro’ prefix in MR could be learned using Gerasymova’s strategy. Before answering this question, we make the observation that rather few of the ‘pro’-prefixed verbs in MR admit a morphological decomposition (we don’t refer here to etymology):

a produce = a pro+duce; a propune = a pro+pune; a proscie = a pro+scie; a proclama = a pro+clama; a procrea = a pro+crea.

while most of such verbs do not this property:

a promova (to promote); a procura (to procure); a promulga (to promulgate); a propaga (to propagate); a procrastina (to procrastinate); a promite (to promise); a provoca (to provoke); a procesa (to process); a proiecta (to make a project, or to design).

The answer to the above question is the following: Abstracting over the sense of the ‘pro’-prefixed verbs in MR — even limiting ourselves to those ‘pro’-prefixed verbs

that admit a morphological decomposition — is not possible using Gerasymova’s learning strategy simply because there is no structural form–meaning (or syntax–semantics) correspondence. More formally, there is no homologous compositionality of form and meaning in the sentences/phrases containing ‘pro’-prefixed verbs in MR.

A summary of the discussion in this section concerning learning-beyond-learning the ‘pro’ prefix in GB, is schematically provided in Figure 8.

5 Conclusions

The grammar learning strategy that we designed in Section 2 of this paper constitutes another example (besides Gerasymova’s previously introduced one) that puts into light the useful capabilities of FCG to technically illustrate a language evolution process. We thus gave an FCG-based solution for modelling a significant problem in the interesting puzzle that occurs at the interference between a Romance language and a Slavic influence.

A second linguistic phenomenon related to the same interference area allowed us to exemplify the possibility to produce a combined strategy out of two simpler grammar evolving strategies.

Below we state the main conclusions we drew at the end of this work, firstly from the computational linguistics point of view:

1. *Learning beyond learning* of aspect-indicating prepositions can be done in FCG (and we have exemplified it accordingly). See the ‘do’ case in GB.

2. *Composite learning* (i.e. combined learning rules) may work in other cases involving (aspect-like indicating) prepositions. In such a case, operationalizing the learning in FCG is a sensibly more elaborated process. See the ‘pro’ case in GB.

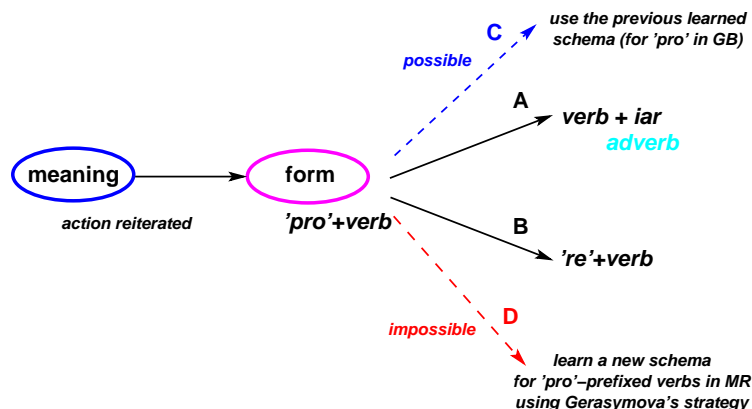


Figure 8: A schematic view on the composite strategy for learning-beyond-learning the ‘pro’ prefix in GB.

3. A given learning strategy may succeed in a certain setup but it may not (and should not!) succeed in another setup. See the ‘pro’ case in MR, when using Gerasymova’s learning strategy is attempted.

In particular, when using the learning strategy designed by Gerasymova, *learning* of the ‘pro’ (repetition) marker in GB on the one hand and the *impossibility of learning* the prefix ‘pro’ (bearing the sense of precedence) in MR on the other hand, shows us that one important *condition* should be satisfied:

form and meaning should be homologously structured.

This condition is clearly met in the case of Russian verb aspects:

```
'po'+cita -> read + for-a-while
'po'+risova -> draw +
for-a-while
```

4. *Limiting the search space* for the new construction/grammar versions is crucial. Learning is always done locally; its validation (wrt a test suite) should be holistic.

5. Beyond all these issues, up to our understanding, the context — and, more generally, the *language game concept*, especially the careful presentation of taught sentences — is a very important matter when it comes to learning in an FCG-like setup.

From the computational learning perspective, we showed that the grammar repair heuristics employed in

FCG can be naturally derived in a (FCGlight-like) setup that takes advantage of the fact that meaning FSs can be organized as a lattice based on the subsumption relation.

6 Further work

We question whether it is really necessary that a new form — like the one presented in Section 2 —, namely for an event whose type is *under-completion* be learned through a language game. Our opinion is that the ‘do’-transformation rules (see Figure 1) can be “invented” by one agent alone, through examination of a test suite restricted to sentences built around ‘do’-prefixed verbs like “domănânc” (I’m finishing to eat), “docitesc” (I’m finishing to read), “doscriu” (I’m finishing to write) etc and the associated meaning FSs.

Here we sketch a [*single-agent-based*] *grammar-evolving strategy* whose output is similar to the one produced by the [*two-agent-based*] strategy that we presented in this paper:

Step 1. Abstracting over the common event-type, i.e. *under-completion*, in direct connection with the prefix ‘do’, may prompt the agent to propose a semantically related verb, for instance ‘complete’, to take the place of ‘do’, while perfectly preserving the previously associated meaning for each sentence in the test suite.

Step 2. In order to maintain the simplicity over the pair of verbs ‘complete’+‘eat’, ‘complete’+‘read’, ‘complete’+‘write’ etc, the agent may decide to impose inflectional constraints/features on only one of them (for instance ‘complete’), while the other verb will be put in an unfinite form.²⁰

Step 3. If the unfinite verb form that was chosen at the above step is not appropriate — with respect to the agent’s history of parsed and generated sentences —, the agent may create a new unfinite form for instance by using an already existing one and marking it using a (possibly new) marker/preposition that would designate its newly acquired functionality.²¹

Step 4. Apply the steps 1, 2 and 3 of the strategy we described at the end of Section 2.

Step 5. To fully repair (i.e. make operational) its grammar following the above changes, the agent should

- make a copy of the ‘complete’ verb’s construction and change its valence list according to the new role (‘de’, v-past-participle, valence-of-v);
- remove certain old features (like *aspect*, *perfective*) from the syntactic categories of the verbs *v* (‘a mânca’, ‘a citi’, ‘a scrie’ etc) that previously admitted the ‘do’-prefixation.

Step 6. Alternatively to the above step, the agent could

- invent a simple (*supine- νp*) rule to link the chosen marker (‘de’) to the νp expressed by the verb *v* and its complements;
- modify the ‘complete’ verb’s valence list so that its direct object can be made either of an *n p* (assuming that it was as such before applying this procedure) or the newly created verb phrase (*supine- νp*).

²⁰MR chose to put inflectional constraints upon the ‘complete’/‘a termina’ verb and let the other verb (for instance ‘a mânca’) in a unfinite form, namely the supine.

²¹MR uses the supine marker ‘de’ to this aim, as a preposition (to the singular, masculine form of) the verb’s past participle.

The reader should note that in fact the above steps 1, 2 and 3 replace the work done by the teacher in the language game on which the procedure given in Section 2 is based, while *Step 4* is work common to both procedures.

Acknowledgement

This work was supported by the “ALEAR” research project carried in the EU FP7 programme and also by its counterpart “ALEAR” 37EU funded by the Romanian Ministry of Education and Research.

This technical report is an extended version of the [7] paper.

References

- [1] Hassan Ait-Kaci and Andreas Podelski. Towards a meaning of LIFE. *Journal of Logic Programming*, 16:195–234, 1993.
- [2] Benjamin Bergen and Nancy Chang. Embodied Construction Grammar in Simulation-Based Language Understanding. In *Construction Grammar(s): Cognitive and Cross-Language Dimensions*. John Benjamins, 2005.
- [3] Katrien Beuls. Verbal agreement in Hungarian. In Luc Steels, editor, *Fluid Construction Grammar*. Springer Verlag, To appear, 2011.
- [4] Bob Carpenter. *The Logic of Typed Feature Structures – with applications to unification grammars, logic programs and constraint resolution*. Cambridge University Press, 1992.
- [5] Liviu Ciortuz. LIGHT — A constraint language and compiler system for typed-unification grammars. In *KI-2002: Advances in Artificial Intelligence*, volume 2479, pages 3–17, Berlin, Germany, 2002. Springer-Verlag.
- [6] Liviu Ciortuz and Vlad Saveluc. FCGLight: Making the bridge between Fluid Construction Grammars and main-stream unification grammars using feature constraint logics. In Luc Steels, editor, *Fluid*

- Construction Grammar*. Springer Verlag, To appear, 2011.
- [7] Liviu Ciortuz and Vlad Saveluc. Learning to unlearn in lattices of concepts: A case study in Fluid Construction Grammars. In *Proceedings of SYNASC 2011*, Timișoara, Romania, 2011. IEEE Computer Society. (In print.).
- [8] William Croft. *Radical construction grammar: syntactic theory in typological perspective*. Oxford Linguistics. Oxford University Press, 2001.
- [9] Charles Fillmore. Grammatical Construction Theory and the familiar dichotomies. In *Language Processing in Social Context*, pages 17–38. North-Holland/Elsevier, 1989.
- [10] Kateryna Gerasymova. Acquisition of aspectual grammar in artificial systems through language games, 2009. Humboldt Universitaet zu Berlin, Germany, MS thesis.
- [11] Kateryna Geraysmova. Expressing grammatical categories in FCG. In Luc Steels, editor, *Fluid Construction Grammar*. Springer Verlag, To appear, 2011.
- [12] Adele Goldberg. *Constructions: A Construction Grammar Approach to Argument Structure*. University Of Chicago Press, 1995.
- [13] Vanessa Micelli. Information structure and field topology with a case study for German. In Luc Steels, editor, *Fluid Construction Grammar*. Springer Verlag, To appear, 2011.
- [14] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
- [15] Carl Pollard and Ivan A. Sag. *Head-driven Phrase Structure Grammar*. CSLI Publications, Stanford, 1994.
- [16] Ivan A. Sag. Sign-Based Construction Grammar: An informal synopsis. In H. Boas and I. A. Sag, editors, *Sign-Based Construction Grammar*. CSLI Publications, The Center for the Study of Language and Information, Stanford University, 2010. Version of August 23, 2010.
- [17] S. Shieber. *An introduction to unification-based approaches to grammars*. CSLI Publications, Stanford, 1986.
- [18] Michael Spranger and Martin Loetzsch. Processing distributed information the case of German locative phrases. In Luc Steels, editor, *Fluid Construction Grammar*. Springer Verlag, To appear, 2011.
- [19] Luc Steels. Language games for autonomous robots. *IEEE Intelligent Systems*, 16(5):16–22, September 2001.
- [20] Luc Steels. Constructivist development of grounded construction grammars. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA, 2004. Association for Computational Linguistics.
- [21] Luc Steels. Introducing Fluid Construction Grammar. In Luc Steels, editor, *Fluid Construction Grammar*. Springer Verlag, To appear, 2011.
- [22] Boris Unbegaun, Paul Falla, and Marcus Wheeler. *Oxford Russian Dictionary*. Oxford University Press, 2007. Fourth Edition.

Appendix

Here we give the FCG code for the constructions in the the source (GB) grammar and the target grammar (MR, both versions) whose learning was presented in Section 2. For the full code, the reader should consult www.info.uaic.ro/~ciortuz/ALEAR/BanatPrefixes.

```
;;;;; GB grammar

(def-lex-cxn eat-cxn
  (def-lex-skeleton eat-cxn
    :syn-cat (==1 (lex-cat verb) (person 1) (nr sg))
    :meaning (==1 (pronoun-lsg ?individual ?context)
                  (eat ?event ?context) (agent ?event ?individual))
    :sem-cat (==1 (arg ?event ?context))
    :string "mananc"))

(def-lex-cxn finish-cxn
  (def-lex-skeleton finish-cxn
    :syn-cat (==1 (lex-cat verb) (person 1) (nr sg)
                  (accept-direct-object +) (accept-supine -))
    :meaning (==1 (pronoun-lsg ?individual ?context)
                  (finish ?event ?context) (agent ?event ?individual))
    :sem-cat (==1 (arg ?event ?context))
    :string "termin"))

(def-lex-cxn work-cxn
  (def-lex-skeleton work-cxn
    :syn-cat (==1 (lex-cat noun) (nr sg))
    :meaning (==1 (work ?work-obj ?context))
    :sem-cat (==1 (arg ?work-obj ?context))
    :string "lucrul"))

(add-cxn (make-cxn prefix-do (:label prefix)
  ((?top-unit
    (footprints (==0 prefix-do))
    (tag ?meaning (meaning (== (event-type ?event under-completion))))
    (sem-subunits (== ?verb-unit)))
  (?verb-unit (sem-cat (== (arg ?event ?context))))
  ((j ?prefix-unit ?verb-unit)
    (footprints (==1 prefix-do))
    ?meaning))
  <-->
  ((?top-unit
    (footprints (==0 prefix-do))
    (syn-subunits (== ?verb-unit))
    (tag ?form
      (form (== (string ?prefix-unit "do-")
              (meets ?prefix-unit ?verb-unit))))))
  (?verb-unit
    (syn-cat (==1 (lex-cat verb))))))
```

```

      ((j ?prefix-unit ?verb-unit)
       (footprints (==1 prefix-do))
       (syn-cat (== (lex-cat prefix)))
       ?form))
) *constructions*)

(add-cxn (make-cxn direct-object (:label object)
  ((?top-unit
    (footprints (==0 direct-object))
    (tag ?meaning (meaning (== (object ?event ?individual))))
    (sem-subunits (== ?verb-unit ?object-unit)))
   (?verb-unit
    (sem-cat (==1 (arg ?event ?context))))
   (?object-unit
    (sem-cat (==1 (arg ?individual ?context))))
   ((j ?object-unit)
    (footprints (==1 direct-object))
    ?meaning))
  <-->
  ((?top-unit
    (footprints (==0 direct-object))
    (syn-subunits (== ?verb-unit ?object-unit))
    (tag ?form (form (== (meets ?verb-unit ?object-unit))))
   (?verb-unit
    (syn-cat (==1 (lex-cat verb) (accept-direct-object +)))
   (?object-unit
    (footprints (==0 direct-object))
    (syn-cat (==1 (lex-cat noun))))
   ((j ?object-unit)
    (footprints (==1 direct-object))
    ?form))
  ) *constructions*)

;;;;; MR grammar, 1st version

(def-lex-cxn eat-cxn
  (def-lex-skeleton eat-cxn
    :syn-cat (==1 (lex-cat verb) (mood supine))
    :meaning (== (eat ?event ?context))
  :sem-cat (==1 (arg ?event ?context))
  :string "de mancat"))

(def-lex-cxn finish-cxn
  (def-lex-skeleton finish-cxn
    :syn-cat (==1 (lex-cat verb) (mood indicative) (person 1)
                (nr sg) (accept-direct-object -) (accept-supine +))
    :meaning (== (pronoun-1sg ?individual ?context)
                (agent ?event ?individual)
                (event-type ?event under-completion))
    :sem-cat (==1 (arg ?event ?context))

```

```

:string "termin"))

(add-cxn (make-cxn supine (:label supine)
  ((?verb-unit
    (sem-cat (==1 (arg ?event ?context))))
  (?supine-unit
    (footprints (==0 supine))
    (sem-cat (==1 (arg ?event ?context))))
  ((j ?supine-unit)
    (footprints (==1 supine))))
  <-->
  ((?top-unit
    (footprints (==0 supine))
    (syn-subunits (== ?verb-unit ?supine-unit))
    (tag ?form (form (== (meets ?verb-unit ?supine-unit))))))
  (?verb-unit
    (syn-cat (==1 (lex-cat verb) (mood indicative) (accept-supine +))))
  (?supine-unit
    (syn-cat (==1 (lex-cat verb) (mood supine))))
  ((j ?supine-unit)
    (footprints (==1 supine-object))
    ?form))
) *constructions*)

;;;; MR grammar, 2nd version

(def-lex-cxn eat-cxn
  (def-lex-skeleton eat-cxn
    :syn-cat (==1 (lex-cat verb) (mood supine))
    :meaning (== (eat ?event ?context))
    :sem-cat (==1 (arg ?event ?context))
    :string "de mancat"))

(add-cxn (make-cxn finish-cxn (:label finish)
  ((?top-unit
    (footprints (==0 finish-cxn))
    (tag ?meaning (meaning (== (event-type ?event under-completion))))
    (sem-subunits (== ?verb-unit))))
  (?verb-unit
    (sem-cat (== (arg ?event ?context))))
  ((j ?prefix-unit ?verb-unit)
    (footprints (==1 finish-cxn))
    ?meaning))
  <-->
  ((?top-unit
    (footprints (==0 finish-cxn))
    (syn-subunits (== ?verb-unit))
    (tag ?form
      (form (== (string ?prefix-unit "termin")
        (meets ?prefix-unit ?verb-unit))))))

```

```
(?verb-unit (syn-cat (==1 (lex-cat verb) (mood supine))))  
((j ?prefix-unit ?verb-unit  
  (footprints (==1 finish-cxn))  
  (syn-cat (== (lex-cat prefix)))  
  ?form))  
) *constructions*
```