

**Towards a model of constructional meaning for natural language
understanding**

Carlos Periñán-Pascual

Abstract

Few researchers in natural language processing are nowadays concerned with linguistically-aware applications. On the contrary, the prevailing trend is towards the search of engineering solutions to practical problems, where researchers are motivated by the immediate gratification from the stochastic paradigm. As a result, there have been few attempts to confront the new challenges in linguistics from the natural language processing approach. The goal of this chapter is to introduce the theoretical foundation underlying ARTEMIS, a knowledge-based system which is intended to simulate natural language understanding in the framework of Role and Reference Grammar. More specifically, we will focus on how to enhance this functional model in order to make argumental constructions play a decisive role in the computational analysis of the deep semantics in the text.

1. Introduction

Natural language understanding constitutes a research field of increasing interest in different disciplines, such as linguistics, cognitive science or natural language processing (NLP). From the NLP perspective, the goal of natural language understanding was early described in the realm of artificial intelligence:

We can describe the process of understanding language as a conversion from a string of sounds or letters to an internal representation of 'meaning'. In order to do this, a language-understanding system must have some formal way to express its knowledge to a subject, and must be able to represent the 'meaning' of a sentence in this formalism. The formalism must be structured so the system can use its knowledge in conjunction with a problem-solving system to make deductions, accept new information, answer questions, and interpret commands. (Winograd 1972: 23-24)

Obviously, it is much easier to build this type of NLP systems when linguistic theories are neglected, but those systems will unavoidably fail from a semantic point of view (Raskin 1987). NLP applications which can work with no foundation in any linguistic theory are deceptively intelligent (Halvorsen 1988), since they don't really allow natural language understanding. Therefore, robust NLP systems require a sound linguistic

model, but what model turns out to be the most beneficial if we intend to convert a sentence into a text meaning representation?

In this regard, we have developed a prototype of NLP system which is grounded in the theoretical model of Role and Reference Grammar (RRG) (Van Valin and LaPolla 1997; Van Valin 2005) and which exploits FunGramKB as its knowledge base (Periñán-Pascual and Arcas-Túnez 2004, 2005, 2007, 2008, 2010a, 2010b; Periñán-Pascual and Mairal-Usón 2009, 2010, 2011; Mairal-Usón and Periñán-Pascual 2009). Although RRG was not devised within computational linguistics, this functional model turns out to be very useful for text meaning representation, which can be described in terms of a logical structure. However, we had to fully integrate constructional meaning into RRG to deepen semantic processing by incorporating the fine-grained constructional schemata from the Lexical Constructional Model (LCM) (Ruiz de Mendoza and Mairal-Usón 2008; Mairal-Usón and Ruiz de Mendoza 2008, 2009) into FunGramKB. In this way, the knowledge base provides a lexico-conceptual architecture in which to anchor a comprehensive model of constructional meaning like the LCM. The aim of this chapter is to describe how an NLP system can derive the semantic representation of a sentence within the RRG framework when argumental constructions occur in the cognitive-linguistic interface. As a result, not only can we gain a better understanding of how language comprehension works, but we can also apply our research to develop enhanced text-based systems (e.g. information extraction, machine

translation or automatic summarizing) and dialogue-based applications (e.g. question-answering or tutoring systems). In essence, we argue that some adjustments of RRG are required in order to make the theory applicable in computational language processing, and particularly in natural language understanding. Further considerations on whether these adjustments are also motivated by the way speakers and hearers process language are out of the scope of this paper, but we expect that future research will address this issue. This chapter is structured as follows. Section 2 briefly describes the two theoretical models which support the linguistic level in FunGramKB, whose main features are in turn presented in section 3. Finally, section 4 gives an account of the way in which the RRG parser manages to integrate constructional meaning by means of FunGramKB.

2. Role and Reference Grammar and the Lexical-Constructional Model

RRG is one of the most relevant functional models of language in current linguistics.¹ RRG was not actually designed for computational linguistics, but this model presents three characteristics which make it suitable for NLP:

¹ Since readers of this book are supposed to be familiar with the RRG framework, we do not intend to provide a detailed account of this functional model.

- a. RRG is a model where morphosyntactic structures and grammatical rules are explained in relation to their semantic and communicative functions.
- b. RRG is a monostratal theory, where the syntactic and semantic components are directly connected through a bidirectional “linking algorithm”.
- c. RRG is a model which owns typological adequacy.

These features are essential for a computational model which aims to provide natural language understanding. First, a functional view of language allows us to capture syntactic-semantic generalizations which are fundamental to explain the semantic motivation of grammatical phenomena. Second, the system is more effectively designed if an algorithm is able to account for both the comprehension and the production of linguistic expressions. Third, typological adequacy becomes an added value when working in a multilingual environment.

RRG is a projectionist theory of language, where many features in the syntactic realization of clause arguments are mapped from the lexical entries of verbs. However, it is important to bear in mind that in the syntax-semantics interface the meaning of the verb is undoubtedly shaped by the meaning of the constructions in which the verb appears. As a result, the meaning of the sentence is determined compositionally by both lexical and constructional meanings. In this respect, the LCM—a usage-based constructionist model of language which goes beyond the core grammar—

allows a bridge between projectionist theories, and more particularly RRG, and constructional theories (Goldberg 1995, 2006; Croft 2001). Indeed, the LCM recognizes the following four levels of constructional meaning:²

- a. Level 1, or argumental layer, accounts for the core grammatical properties of lexical items, as well as argument structure constructions like those postulated by Goldberg (1995, 2006).
- b. Level 2, or implicational layer, is concerned with constructional configurations (such as *What's X doing Y?*) based on low-level situational cognitive models (or specific scenarios), giving rise to meaning interpretations which carry a heavily conventionalized implication.
- c. Level 3, or illocutionary layer, deals with illocutionary constructions (e.g. *Can you (please) X?*), which are considered a matter of high-level situational models (or generic scenarios).
- d. Level 4, or discourse layer, addresses discourse constructions based on high-level non-situational cognitive models (such as reason-result or condition-consequence), with particular emphasis on cohesion and coherence phenomena.

The LCM demonstrates that, although projectionist and constructional approaches are often apparently opposed to each other, “the reality of

² Up to now, the bulk of the work on the LCM has been concerned with the argumental layer (e.g. Baicchi 2007, 2011; Pérez Hernández and Peña Cervel 2009; Peña Cervel 2009; Ruiz de Mendoza and Luzondo Oyón in press) with only some preliminary work on the rest of the levels (e.g. Del Campo Martínez 2011; Ruiz de Mendoza and González 2011).

grammar lies somewhere in the middle between two extremes” (Sugayama 2011: 64-65). Therefore, we have chosen to implement a hybrid model in the linguistic level of FunGramKB, where the bridge is now built between the Lexicon and the Grammaticon.

3. FunGramKB

FunGramKB is a multipurpose lexico-conceptual knowledge base to be implemented in NLP systems, and more particularly for natural language understanding.³ On the one hand, FunGramKB is multipurpose in the sense that it is both multifunctional and multilingual. Thus, FunGramKB has been designed to be potentially reused in many NLP tasks (e.g. information retrieval and extraction, machine translation, dialogue-based systems, etc.) and with many natural languages.⁴ On the other hand, our knowledge base comprises three major knowledge levels, consisting of several independent but interrelated modules:

Lexical level:

³ We use the name “FunGramKB Suite” to refer to our knowledge-engineering tool (www.fungramkb.com) and “FunGramKB” to the resulting knowledge base. FunGramKB Suite was developed in C# using the ASP.NET 2.0 platform and a MySQL database.

⁴ English and Spanish are fully supported in the current version of FunGramKB Suite, although we have just begun to work with other languages, such as German, French, Italian, Bulgarian and Catalan.

- The Lexicon stores morphosyntactic and collocational information about lexical units.⁵ The FunGramKB lexical model is not a literal implementation of the RRG lexicon, although the major linguistic assumptions of RRG are still preserved, i.e. logical structures, macroroles, and the rest of the linking algorithm.
- The Morphicon helps our system to handle cases of inflectional morphology.

Grammatical level:

- The Grammaticon stores the constructional schemata which help RRG to construct the syntax-to-semantics linking algorithm. More particularly, the Grammaticon is composed of several Constructicon modules that are inspired in the four levels of the LCM.

Conceptual level:

- The Ontology is presented as a hierarchical catalogue of the concepts⁶ that a person has in mind, so here is where semantic knowledge is stored in the form of meaning postulates. The Ontology consists of a general-purpose module (i.e. Core Ontology) and several domain-specific terminological modules (i.e. Satellite Ontologies).
- The Cognicon stores procedural knowledge by means of scripts, i.e.

⁵ In this chapter, the term “lexical unit” is used as a synonym of “predicate”, i.e. content words to which morphosyntactic and semantic properties are assigned.

⁶ Terms such as “class”, “category” or “semantic type” are often used in ontology engineering to refer to elements such as FunGramKB “concepts”. However, we prefer the latter, since it better describes the domain of processing in the three-tier model of our NLP knowledge base, i.e. lexical, constructional and conceptual levels.

schemata in which a sequence of stereotypical actions is organised on the basis of temporal continuity, and more particularly on Allen's temporal model (Allen 1983; Allen and Ferguson 1994).

- The Onomasticon stores information about instances of entities and events, such as Bill Gates or 9/11. This module stores two different types of schemata (i.e. snapshots and stories), since instances can be portrayed synchronically or diachronically.

In the FunGramKB architecture, every lexical or grammatical module is language-dependent, whereas every conceptual module is shared by all languages. In other words, linguists must develop one Lexicon, one Morphicon and one Grammaticon for English, one Lexicon, one Morphicon and one Grammaticon for Spanish and so on, but knowledge engineers build just one Ontology, one Cognicon and one Onomasticon to process any language input⁷ conceptually. In this scenario, FunGramKB adopts a conceptualist approach, since the Ontology becomes the pivotal module for the whole architecture.

3.1. Thematic frames and meaning postulates

The FunGramKB ontological concepts are not stored as atomic symbols but are provided with semantic properties such as the thematic frame and the

⁷ Strictly speaking, this is true for any input from the range of languages which are considered to be culturally similar.

meaning postulate. Both of them are conceptual schemata, since they employ concepts—and not words—as the building blocks for the formal description of meaning. Thus, thematic frames as well as meaning postulates become language-independent semantic knowledge representations.

On the one hand, every event in the Ontology is assigned one single thematic frame, i.e. a conceptual construct which states the number and type of participants involved in the prototypical cognitive situation portrayed by the event. To illustrate, we present the thematic frame of +FREEZE_00, to which lexical units such as *freeze* [Eng], *helar*, *congelar* [Spa], *gelare*, *congelare* [Ita], or *geler*, *congeler* [Fre] are linked:

(1) (x1)Theme (x2)Referent

Thus, the thematic frame (1) describes a prototypical cognitive scenario in which “an entity (Theme) freezes another entity (Referent)”.⁸ In FunGramKB, thematic roles are not specific to a given conceptual dimension, or metaconcept, in the Ontology (e.g. #COGNITION, #EMOTION, #POSSESSION or #TRANSFER, among many others), but the metaconcept itself enriches the meaning of thematic roles. In other words, the participants in the thematic frame acquire different interpretations according to the metaconcept under which the given concept is placed. For example, #PERCEPTION involves that Theme refers to the

⁸ It should not be forgotten that, although one or more subcategorization frames can be assigned to a single lexical unit, every concept is provided with just one thematic frame.

entity that perceives another entity (Referent), whereas #CONSTITUTION implies that Theme refers to the entity that is made up of other entities (Referent). In this way, the inventory of thematic roles is dramatically minimized while preserving their semantic informativeness. In this regard, a key requirement for objectivity is to provide thematic roles with accurate definitions on the basis of the location of thematic frames within the Ontology.⁹

In contrast to RRG, FunGramKB thematic roles do have an independent status from the logical structure. They even play a paramount role in the text meaning representation; indeed, the lexico-conceptual linkage can only be performed once the constituents in the parse tree are tagged with the FunGramKB thematic roles, as described in section 4. Neither RRG nor FunGramKB thematic relations are assigned on an arbitrary basis, but their nature and scope are distinctly different. Whereas RRG establishes the thematic roles of the verb through the argument positions in the logical structure, which is created on the basis of the Aktionsart resulting from the application of linguistic tests, FunGramKB thematic roles are determined once the concept to which the verb has been linked is placed into a given ontological metaconcept. As a result, RRG logical structures do only take into account the thematic relations which have an impact on the syntax of the verb (i.e. grammatical relevance),

⁹ Perrián-Pascual and Mairal-Usón (2010) described the semantic interpretation of thematic roles in each FunGramKB metaconcept.

whereas FunGramKB thematic frames encapsulate the thematic roles which are assigned to participants whose presence is required by the cognitive scenario portrayed by the event (i.e. conceptual relevance). This is why a verb of motion such as *march* in ‘Troops also marched to burn an armoury’ has a single argument position (i.e. Mover) in the logical structure, but the whole event could not be understood cognitively unless the Origin, Location and Goal are also born in mind. Therefore, since argument variables in logical structures cannot be automatically linked to variables in thematic frames, the mapping should take place in the lexical entry, as described in the next section.

On the other hand, a meaning postulate is a set of one or more logically connected predications ($e_1, e_2 \dots e_n$), i.e. cognitive constructs carrying the generic features of concepts.¹⁰ Consider (2) as a representation of the meaning postulate of +FREEZE_00:

(2) +(e1: +COOL_00 (x1)Theme (x2)Referent (f1: +MUCH_00)Quantity (f2: (e2: +BECOME_00 (x2)Theme (x3: +SOLID_00)Attribute))Result)

That is, an entity (Theme) cools another entity (Referent) so much that the latter becomes solid. Unlike some other approaches in NLP (e.g. WordNet, among many others), FunGramKB adopts a deep semantic approach which strongly emphasizes the commitment to provide meaning definitions via

¹⁰ Perrián-Pascual and Arcas-Túnez (2004) described the formal grammar of well-formed predications for the FunGramKB meaning postulates.

meaning postulates. At first sight, thematic frames could be deemed to be redundant because they are indeed fully integrated into meaning postulates, i.e. every participant in the thematic frame is referenced by co-indexation to some participant in the meaning postulate. However, the motivation for explicitly building thematic frames lies in the need to bring to the fore those participants which will be potentially involved in the mapping between RRG logical structures (linguistic level) and FunGramKB thematic frames (cognitive level). In fact, if thematic frames did not exist, it would not be possible for linguists and knowledge engineers to perform this mapping, and consequently the lexico-conceptual linkage would eventually be non-existent. In this sense, the relevance of thematic frames becomes manifest.

3.2. Lexical entries

In the FunGramKB Lexicon, lexical entries are provided with the following types of information:¹¹

- a. Basic: headword, index, and language.
- b. Morphosyntax: graphical variant, abbreviation, phrase constituents, category, number, gender, countability, degree, adjectival position, verb paradigm and constraints, and pronominalization.
- c. Core grammar: Aktionsart, lexical template and constructions.

¹¹ Mairal-Usón and Perrián-Pascual (2009) presented the anatomy of the FunGramKB Lexicon by describing the different types of features which form part of a predicate's lexical entry.

d. Miscellaneous: dialect, style, domain, example and translation.

In the case of verbs, the most important lexical component is the core grammar, which contains those attributes whose values allow the system to build the basic logical structure of verbs automatically. Table 1 presents a brief description of these attributes.

Table 1. Attributes in the core grammar.

Attribute		Description
Aktionsart		The most representative RRG verb class in which the verb can occur
Lexical Template	Variables	Variables <i>x</i> , <i>y</i> and <i>z</i> represent the prototypical arguments of the verb
	Thematic-frame mapping	Binding the previous variables to some of the participants in the thematic frame of the concept to which the verb is linked
	Idiosyncratic features	Exceptions to the RRG Default Macrorole Assignment Principle
Constructions		Inventory of argumental constructions in which the verb can take part

To illustrate, Table 2 presents the core grammar of the lexical unit *freeze*.

Table 2. The core grammar of *freeze*.

Attribute		Value
Aktionsart		Causative accomplishment
Lexical Template	Variables	<i>x</i> , <i>y</i>
	Thematic-frame mapping	<i>x</i> = Theme, <i>y</i> = Referent
	Idiosyncratic features	MR2
Constructions		INCH (Inchoative Construction) MIDD (Middle Construction) RESU (Transitive Resultative Construction) RESI (Intransitive Resultative

	Construction)
--	---------------

It should be noticed that knowledge on constructions is not stored in the Lexicon. As described in the following section, it is the Grammaticon that holds the constructional schemata, i.e. machine-tractable representations of constructions, but the lexical entry should have pointers to all those constructions in which a given verb can take part. Thus, FunGramKB enables efficient management of cross-linguistic constructional generalizations—i.e. the Grammaticon is a repository of types of constructions—and constructional variability—i.e. those types of constructions can be instantiated in some languages by means of the pointers located in the lexical entries.

In addition to the constructions derived from the Grammaticon, every verb in the Lexicon is provided with one and only one Kernel Construction, which is built on the basis of the knowledge in the core grammar, primarily the Aktionsart and the lexical template. Depending on the number of variables in the lexical template, the verb will typically occur in a Kernel-1, Kernel-2 or Kernel-3 Construction.¹² For instance, the system can directly derive the Kernel-2 Construction from the core grammar of *freeze*.

3.3. Constructional schemata

¹² In fact, these Kernel Constructions correspond to intransitive, monotransitive and ditransitive constructions respectively.

Constructional schemata are stored in the FunGramKB Grammaticon, which enables a multi-tiered approach to represent the various dimensions of text meaning. A key issue in this module is the definition of “construction”. In the Golbergian model, practically any linguistic pattern is recognized as a construction, being made up of a “form” linked to a “meaning”, as can be seen in the following definitions:

C is a CONSTRUCTION iff_{def} C is a form-meaning pair $\langle F_i, S_i \rangle$, such that some aspect of the form F_i or some aspect of S_i is not strictly predictable from C's component parts or from other previously established constructions. (Goldberg 1995: 4)

Any linguistic pattern is recognized as a construction as long as some aspect of its form or function is not strictly predictable from its component parts or from other constructions recognized to exist. In addition, patterns are stored as constructions even if they are fully predictable as long as they occur with sufficient frequency. (Goldberg 2006: 5)

In fact, constructions serve to capture “our grammatical knowledge *in toto*” (Goldberg 2006: 18). Thus, a sentence such as “He fried the egg in the pan” comprises the constructions “the egg” and “in the pan”, but every single

word in the sentence as well as the suffix *-ed* can also be seen as constructions. As this example demonstrates, it is arguable that “construction” is such a broad a term that an accurate definition is not possible. Moreover:

There is no precise definition of (i) the notion of a productive unit in CxG, (ii) the way productive units are acquired step by step from incoming input utterances, and (iii) the combination operations that combine constructions into (an open-ended number of) new utterances. (Bod 2009: 130)

This criticism is compounded by the fact that frequency serves to determine the stability of any form-meaning pairing as a construction:

(...) what if a form-meaning pairing is produced only once by a communicatively competent native speaker and not only understood but also regarded as highly natural output by other equally competent native speakers within a community of speakers? Would that not be a construction? (Ruiz de Mendoza this volume)

Indeed, it is precisely Ruiz de Mendoza (this volume) who proposes the usage-based notion of construction underlying the LCM:

(...) the LCM defines a construction as a form-meaning (or function) pairing where form affords access to meaning and meaning is realized by form to the extent that such processes have become entrenched, through sufficient use, in the speaker's mind and are generally recognized by competent speakers of the language in question to be stably associated or are at least potentially replicable by other competent speakers of the same language with immaterial variation in its form and meaning.

We share this view which highlights productivity, bi-univocity and replicability as crucial properties to determine whether a form-meaning pairing is regarded as a construction, but our computational approach to constructional meaning requires, first and foremost, a clear-cut distinction between “construct” and “construction”. On the one hand, a construct refers to any form-meaning pairing which serves as a building block in the compositionality of the sentential semantics. Therefore, the FunGramKB constructs can be found in both the linguistic realization (i.e. the input text) and the conceptual representation (i.e. the COREL scheme), where the minimal constructs take the form of lexical units¹³ and ontological concepts respectively. On the other hand, a construction refers to any linguistic

¹³ In FunGramKB, lexical units include simple and derived words as well as multi-word expressions such as idioms. Derivative morphemes are not processed as linguistic objects up to now, but we do not rule out the possibility of treating both inflectional and derivational morphology in the Morphicon.

construct whose meaning cannot be fully derived from the sum of the lexical meanings of the individual constructs taking part in the utterance. Therefore, the notion of construction is viewed from a holistic approach, since “the meaning of the whole is greater than the meaning of the parts” (Lakoff 1977: 239). We can claim that any construction is a construct itself, but not all constructs can be deemed to be constructions. Accordingly, constructs can be categorized as constructional and non-constructional; however, we prefer to use the term construct to exclusively refer to the latter subtype, and construction to the former. Therefore, from the FunGramKB approach, the sentence “He fried the egg in the pan” only consists of the Kernel-2 Construction. The remaining components can only be perceived as constructs, whose meanings are directly derived from their meaning postulates. Unlike Construction Grammar, the building blocks of linguistic realizations are constructs, where some of them can attain a constructional status.

More clarity is also required regarding the scope of constructions in our model of language. As Rappaport Hovav and Levin (1998) stated, both projectionist and constructivist accounts of language acknowledge the existence of constructional meaning and lexical meaning: whereas the former helps to determine the structuring of argument realization, the latter becomes an idiosyncratic part of the word and serves to distinguish that word from others in the same semantic class. However, the main difference between these two approaches actually lies in the interface between syntax

and semantics: morphosyntactic structures can be mapped from the lexical semantics of the verb or, by contrast, the meaning of the sentence is derived from constructions which can override the typical verbal semantics. In terms of the FunGramKB model, constructs get their meaning from the meaning postulates stored in the Ontology, whereas constructional meaning is obtained from the schemata in the Grammaticon. Therefore, the constructional schema serves as a machine-tractable representation of the construction. In contrast to Goldberg's Construction Grammar, in which the formalization of constructional knowledge is rather underspecified, the FunGramKB constructional schemata are defined in terms of descriptors and constraints, where the latter licenses compositionality with other constructs or constructions. In the LI-Constructicon, for instance, every constructional schema is described by means of an Aktionsart, the number and type of variables in the logical structure, the thematic role corresponding to the new variables, the macroroles which cannot be inherited from the core grammar of the verb, and the conceptual contribution to the COREL scheme of the sentence in which the construction is embedded. With regard to the variables in the logical structure, the constraints are focused on phrase realizations¹⁴ and selectional preferences.¹⁵ Figure 1 serves to illustrate the interface of the Intransitive

¹⁴ In the case of prepositional phrases, the most typical prepositions can also be stated.

¹⁵ These selectional preferences take the form of concepts from the FunGramKB Ontology.

Resultative Construction, whose corresponding attribute-value matrix is presented in Figure 2.

Figure 1. The interface of the L1-Constructicon.

Figure 2. The attribute-value matrix of the Intransitive Resultative Construction.

As can be seen, argumental constructions are formalized by means of descriptors and constraints which are oriented to the logical structure and the COREL scheme, since variation in the syntactic context of the verb should eventually involve a variation in the aspectual meaning (i.e. Aktionsart) and/or in the conceptual meaning (i.e. COREL scheme). By contrast, higher-level constructions (i.e. implicational, illocutionary and discursive), which are correspondingly exemplified in (3), do not alter the logical structure but can only extend their corresponding COREL scheme.

- (3) a. What's the child doing in the kitchen with the carving knife? [L2-construction]
- b. I wonder if you could give me the dictionary. [L3-construction]
- c. You can have the day off tomorrow on condition that you work on Saturday. [L4-construction]

The LCM constructions are essentially meaning-bearing devices, regardless of whether their semantic burden lies in the Aktionsart or the COREL scheme. Therefore, the *raison d'être* of a construction is its semantic contribution to that meaning of the sentence which cannot be derived from the lexical units.

We can conclude that FunGramKB adopts a hybrid approach to constructional meaning, i.e. halfway between projectionism and constructivism. On the one hand, FunGramKB shows a clear-cut separation between the linguistic modules, i.e. the Lexicon and the Grammaticon, where the projection from syntax to semantics goes through the pointers in the lexical entries. Moreover, the assumption that “all levels of grammatical analysis involve constructions” (Goldberg 2006: 5) cannot be applied in our model of language, since constructional meaning should be located at one of the four levels of the LCM, where constructions are not found below the argumental layer. On the other hand, the Grammaticon provides meaningful abstract representations of constructions, rather independent from language so as to determine cross-linguistic generalizations, where morphosyntactic constraints on the variables in the logical structure serve to license a given construction.¹⁶ In this framework, the meaning of the sentence is determined by the core grammar of the verb, together with the meaning of argumental, implicational, illocutionary and discursive constructions. In the following

¹⁶ This lexico-constructional stance is in line with Boas's (2008) proposal to pay careful attention to individual verb senses in order to solve the problem of constructional overgeneration.

section, we portray how constructional meaning from the LCM can be fully integrated into the RRG semantic representation through the lexico-constructional knowledge in FunGramKB. Due to space limitations, the section focuses on argumental constructions.

4. Building constructional meaning in RRG with FunGramKB

One of the major contributions of FunGramKB to the RRG theoretical model consists in the shift of the logical structure into the conceptual logical structure (CLS), i.e. a language-independent formalism whose role is to be a text meaning representation serving as the bridge between the linguistic realization and the conceptual realm. To illustrate, we present the RRG logical structure (4) and the FunGramKB CLS (5), which are derived from the sentence “The juice froze black in the refrigerator”:

(4) <_{IF} DEC <_{TNS} PAST < be-in' (refrigerator, [[do' (juice, [freeze' (juice)])] CAUSE [BECOME black' (juice)]]>>>

(5) <_{IF} DECL <_{Tense} past <_{CONSTR-L1} RESI <_{CONSTR-L1} INCH
<_{AKT} ACC [+FREEZE_00 (+JUICE_00-Referent,
+BLACK_00-Result)] (+REFRIGERATOR_00-Location)
>>>>>

This conceptual shift involves a number of changes affecting the standard model of semantic representation:

- a. The instantiation of variables does not take the form of predicates but ontological concepts (e.g. +FREEZE_00 or +BLACK_00). As a result, CLSs now become real language-independent representations.
- b. Every concept linked to a variable is assigned a thematic role—e.g. +JUICE_00 is the Referent and +REFRIGERATOR_00 is the Location in the cognitive scenario introduced by the event. Thus, the CLS can be mapped into a COREL scheme via the thematic frame of the event.
- c. The constructional operator is incorporated (i.e. CONSTR-L1), which plays a prominent role in the syntax-semantics linkage.¹⁷
- d. Since an Aktionsart operator has also been introduced (i.e. AKT), the semantic skeleton originating in the Aktionsart is now replaced by an argument pattern headed by the event (i.e. [+FREEZE_00 (+JUICE_00-Referent, +BLACK_00-Result)]).¹⁸

Feature (d) was motivated by the fact that the RRG decompositional system turns out to be excessively noisy from a computational view, since the semantic burden of the sentence is not actually carried by the CLS but by its corresponding COREL scheme. That is, when some kind of reasoning with the input is required, the CLS should be transduced into a COREL

¹⁷ Indeed, every argumental construction is embodied in a constructional operator whose scope is the core in the RRG layered structure of the clause.

¹⁸ Despite the name of “argument pattern”, a further nucleus can also be introduced in the case of nuclear cosubordination, just as occurs in the Resultative Construction.

representation, so that it can be enriched by the conceptual knowledge from any cognitive module in FunGramKB. In this COREL mapping process, the operators, the concepts and their thematic roles are the only CLS elements taken into account. Thus, the CLS (5) is modeled into the COREL scheme (6), which can be extended to (7) through the meaning postulate of the verb.

(6) +(e1: past +FREEZE_00 (x1)Theme (x2: +JUICE_00)Referent (f1: (e2: +BECOME_00 (x2)Theme (x3: +BLACK_00)Attribute))Result (f2: +REFRIGERATOR_00)Location)

(7) +(e1: past +COOL_00 (x1)Theme (x2: +JUICE_00)Referent (f1: +MUCH_00)Quantity (f2: (e2: +BECOME_00 (x2)Theme (x3: +SOLID_00 & +BLACK)Attribute))Result (f3: +REFRIGERATOR_00)Location)

According to Figure 2, the Intransitive Resultative Construction has contributed with the predication (8) to the COREL scheme (6). Thus, we demonstrate that the meaning of the construction is independent from the meaning of the verb.

(8) (f1: (e2: +BECOME_00 (x2)Theme (x3: +BLACK_00)Attribute))Result

As can be noticed, the process of extending the meaning postulate of the verb with conceptual knowledge from constructional schemata can involve the readjustment of indices for variables *e* (predication), *x* (argument) and *f* (satellite) as the result of unification.

Consequently, the CLS, which is able to account for a wide range of linguistic phenomena within the RRG framework, serves as the pivot language between the input text and the COREL representation, whereas the COREL scheme, which provides the background knowledge from the FunGramKB conceptual modules, serves as the pivot language between the CLS and the automated reasoner. Thus, the division of labor between the conceptual and the linguistic level is still maintained as one of the central methodological axioms in FunGramKB.

In order to build automatically the CLS (5) and the COREL scheme (6), we developed FunGramKB ARTEMIS, whose interface is shown in Appendix 1.¹⁹ ARTEMIS is an NLP prototype whose current goal is to demonstrate that argumental constructions can be captured in the CLS, so that constructional meaning contributes to modeling the COREL scheme of the sentence.²⁰ ARTEMIS consists of three main components, i.e. the Grammar Development Environment (GDE), the CLS Constructor, and the COREL-scheme Builder. The remainder of this section deals with the construction of the CLS from the RRG approach.²¹

¹⁹ FunGramKB ARTEMIS, which stands for “Automatically Representing Text Meaning via an Interlingua-based System”, is also part of the FunGramKB Suite.

²⁰ In this prototype, we considerably reduced the syntactic complexity of the input. In fact, the system is confined to the universal aspects of a single-clause sentence, i.e. the core and the periphery, where the former is affected by argumental constructions. Moreover, with regard to clausal operators, only present and past tenses are adequately treated.

²¹ The computational implementation of the CLS Constructor and the COREL-scheme Builder remains out of the scope of this study. We further refer the reader to Periñán-Pascual and Arcas-Túnez (in press) for the technical details inside ARTEMIS.

As described above, the CLS involved a number of changes in the RRG framework, affecting especially the layered structure of the clause (LSC), the syntactic templates, and the semantic roles. First, it was essential to incorporate the construction as a category of grammar, so we integrated the L1-CONSTRUCTION node into the LSC. More particularly, the clause is configured as one or more L1-constructions which are recursively arranged, serving to address the issue of constructional compositionality. As shown in Figure 3, the innermost construction introduces the core, which can be modeled by other L1-constructions, typically contributing with a further argument.²²

Figure 3. Enhanced model of LSC (unrefined tree).

In fact, the integration of the L1-CONSTRUCTION node into the LSC is in line with the condition that “a theory of clause structure should capture all of the universal features of clauses” (Van Valin 2005: 3).²³ Structurally, although the parse tree may appear to differ from the standard model, the differences are indeed not so remarkable, since those arguments that constructions bring forward are really part of the core from a logical

²² However, there is the possibility for some constructions to incorporate another nucleus, resulting in nuclear cosubordination.

²³ Although the category of “construction” is universal as such, it is important to bear in mind that a given construction can be specific only to one or a few languages, e.g. the *Time-away* Construction evidenced in the English sentence “Twistin’ the night away” (Jackendoff 1997).

perspective. In this way, Figures 3 and 4 should be deemed to be identical after tree refinement.

Figure 4. Enhanced model of LSC (refined tree).

As in the case of RRG, we do not aim to build a representation in terms of purely syntactic features, so we chose to build the parser upon a feature-based grammar.²⁴ As a result, nodes in the parse tree are represented by means of feature structures. For example, Appendix 2 shows the feature-based parse tree corresponding to the sentence “The juice froze black in the refrigerator”.²⁵

Second, the RRG syntactic analysis is based on an inventory of templates, i.e. syntactic trees which do not explicitly state the order of constituents but just their hierarchical organization. On the contrary, the GDE relies on feature-based production rules, which are subject to the linear order of their constituents.²⁶ One of the main problems was certainly how to handle those peripheral adjuncts which can be located between core constituents (e.g. nucleus or argument) according to the linearity of the text. The solution was simply to allow the system to generate the tree and then to

²⁴ Computationally speaking, the feature-based grammar was parsed by using the well-known Earley chart parser.

²⁵ Although feature structures are usually represented as attribute-value matrices or directed acyclic graphs, ARTEMIS employs the bracketed notation, as shown in Appendix 3.

²⁶ We chose this type of rules due to their ability to model more complex phenomena than context-free grammars.

reconstruct its organization by relocating the displaced constituents (i.e. tree refinement).

The grammar in the GDE consists of three types of production rules:²⁷

- a. Syntactic rules, which build the enhanced framework of the LSC (Figure 4), provided with syntactic units such as nucleus, core, construction, periphery and clause. For example:

(9) PP[p=?p] -> p[p=?p] NP
 NP[Num=?n] -> n[Num=?n] | det[Num=?n] n[Num=?n] |
 det[Num=?n] adj n[Num=?n]

- b. Constructional rules, which serve to embed constructional schemata into the LSC. To illustrate, we present the rules for the Intransitive Resultative Construction, whose scheme was shown in Figure 1:

(10) CONSTR-L1[Tense=?t, Template=RESI, Akt=ACC, Weight=3] -> CORE[Tense=?t, Template=RESI] NUC-S[Phrase=ADJP, Role=Result, Macrorole=n] | CONSTR-L1[Tense=?t] NUC-S[Phrase=ADJP, Role=Result, Macrorole=n]
 CONSTR-L1[Tense=?t, Template=RESI, Akt=ACC, Weight=3] -> CORE[Tense=?t, Template=RESI] NUC-S[Phrase=PP, p=into, Role=Result, Macrorole=n] |

²⁷ For the sake of clarity, we have simplified the production rules shown in this section.

CONSTR-L1[Tense=?t] NUC-S[Phrase=PP, p=into,
Role=Result, Macrorole=n]

CORE[Tense=?t, Template=RESI] -> ARG[Num=?n,
Type=y, Phrase=NP, Role=Referent, Macrorole=U]
NUC[Tense=?t, Num=?n, Template=RESI]

c. Lexical rules, which provide morphosyntactic and semantic information about words. For example:

(11) n[Num=sg, Concept='+JUICE_00'] -> 'juice'

n[Num=sg, Concept='+REFRIGERATOR_00'] ->
'refrigerator'

v[Tense=past, Template=RESI, Concept='+FREEZE_00'] ->
'froze'

adj[Concept='+BLACK_00'] -> 'black'

p[p=in] -> 'in'

Whereas syntactic rules are pre-defined through the GDE, constructional and lexical rules are created in runtime in accordance with the tokens from the input stream. This dynamic process of rule elaboration, which expedites significantly the syntactic parsing, needs to retrieve knowledge from the database in order to complete the attribute-value features. More particularly, constructional rules are generated with the aid of the Lexicon and the Grammaticon (i.e. the core grammar of the verb together with all its

constructional schemata), and lexical rules mainly require the Lexicon and the Ontology.²⁸

In the elaboration of the constructional rules, another relevant aspect was the assignment of macroroles according to the Default Macrorole Assignment Principle (Van Valin 2005: 63), which was slightly adapted to the characteristics of the CLS. However, default values can be overridden by those previously assigned in the Lexicon core grammar—e.g. in the lexical entry of *kick*, the *y* variable is the Undergoer—or in the Grammaticon constructional schemata—e.g. the Inchoative Construction assigns the Undergoer to the *y* variable.

It is important to note that, just as every language has its own inventory of RRG syntactic templates, our formal grammar consists of language-specific rules. As an example of this we may take the constructional rules: constructional schemata in the Grammaticon are shared by all languages in FunGramKB Suite, but constructional rules also state the ordering of nuclei and arguments, so they should be language-specific.

Third, in contrast to RRG, thematic roles do play a paramount role in the CLS. Indeed, only by tagging the constituents in the parse tree with the FunGramKB thematic roles can ARTEMIS perform the lexico-conceptual

²⁸ Predicate conceptualization involves the problem of word-sense disambiguation: since lexical information in FunGramKB is linked to the senses of words (i.e. sense-oriented approach), a word-sense disambiguator should firstly tag the lemmas with a single conceptual label from the Ontology, or, in the case of proper nouns, from the Onomasticon. This disambiguator is still work in progress, so now users must disambiguate polysemous words from the ARTEMIS interface before the parsing occurs.

linkage, i.e. the construction of a fully-fledged conceptual representation in the form of a COREL scheme.

After the parser returns a feature-based tree of the input sentence, the CLS basically results from the extraction of the most relevant semantic units together with their attributes. In other words, the syntax-driven semantics is so embedded in the parse tree itself, certainly much more than in the RRG model, that the system will do nothing but remove the morphosyntactic units of the LSC and relocate the operators according to their scope.

5. Conclusions and future research

In this chapter, we have described how argumental constructions can be fully integrated into the RRG model with the aim to develop knowledge-based NLP systems for language comprehension. As it is widely accepted in the linguistic literature, the verb plays a key role in determining the meaning of the clause, but this meaning is usually shaped by argumental constructions. By implementing the LCM into the linguistic level of FunGramKB, and more particularly in the realm of the Lexicon and the Grammaticon, we have succeeded in narrowing the gap between the so-called projectionist and constructivist approaches to language processing. This has given rise to the CLS, which involves a conceptual shift of the RRG logical structure, allowing now to deal with argumental constructions

as a key component of the semantic representation. To reach this goal, the CLS Constructor requires a constraint-based parser that relies on a robust knowledge base, such as FunGramKB. It is well known that RRG also allows argumental constructions to be represented in terms of constructional schemata. The problem lies in the fact that sentential semantics relies solely on the logical structure. RRG can be semantically enriched by means of CLSs, which construct a bridge between FunGramKB conceptual knowledge, i.e. common-sense, cultural and personal knowledge, and the particular idiosyncrasies as coded in linguistic expressions.²⁹ This conceptualist shift to language processing affects not only the standard model of logical structure but also that of the constructional schema. Despite these modifications, we intend to keep our computational model of language processing close to RRG functional premises, since we also aim to interpret and analyse linguistic realizations in the framework of communication and cognition.

In the last few years we tested our theoretical assumptions on the CLS as a paper prototype, but now ARTEMIS has been released as a proof-of-concept prototype application which intends to demonstrate the feasibility of our approach. As it needs to be thoroughly tested, directions of future research should aim to provide suitable treatment for complex syntactic phenomena. For example, we should extend the scope of the parser

²⁹ Mairal-Usón, Perrián-Pascual and Pérez Cabello de Alba (2012) described the benefits of adopting an ontological approach to the RRG logical structure.

to include both the pre- and post-core slots and the right and left detached positions, as well as giving a wider coverage to operators, particularly in the nucleus (e.g. aspect and negation) and in the core (e.g. modality). Moreover, our stratificational approach to argumental constructions in the LSC undoubtedly simplifies constructional merger, but further analysis of the constraint-based operations is required to restrict the co-occurrence of constructions. Finally, there is still a need to develop a linguistically-aware model which can solve prepositional sense disambiguation problems. This research should focus on the semantic interpretation of predicative prepositional phrases acting as adjuncts, since the remaining cases are already dealt by FunGramKB, through the core grammar of the verb or through the constructional schemata. The semantic interpretation of this type of prepositional phrase clearly involves two issues: the thematic role to be assigned to the adjunct, and the way in which the semantic burden of the preposition will affect the COREL scheme of the sentence.

To conclude, it should be noted that our incipient experimentation with ARTEMIS has yielded such promising results that we expect the CLS to bring numerous benefits to many different NLP fields, from information retrieval to machine translation.

6. Acknowledgments

Financial support for this research has been provided by the DGI, Spanish Ministry of Education and Science, grant FFI2011-29798-C02-01. I would also like to thank Christopher Butler for valuable comments on the first draft and Ricardo Mairal-Usón for fruitful discussion on RRG issues. Any error is mine.

7. References

- Allen, James F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11): 832-843.
- Allen, James F. & Ferguson, George. 1994. Actions and events in interval temporal logic. *Journal of Logic and Computation* 4(5): 531-579.
- Baicchi, Annalisa. 2007. The subsumption process of the intransitive-transitive migration. In *Migrations of Forms, Forms of Migration*, Marina Dossena, Domenico Torretta & Annamaria Sportelli (eds), 21-41. Bari: Progedit.
- Baicchi, Annalisa. 2011. Metaphoric motivation in grammatical structure. The case of the caused-motion construction from the perspective of the Lexical-Constructional Model. In *Motivation in Lexicon, Grammar, and Discourse*, Guenter Radden, Klaus Uwe Panther & Peter Koch (eds), 149-170. Amsterdam/Philadelphia: John Benjamins.

- Boas, Hans C. 2008. Determining the structure of lexical entries and grammatical constructions in Construction Grammar. *Annual Review of Cognitive Linguistics* 6: 113-144.
- Bod, Rens. 2009. Constructions at work or at rest? *Cognitive Linguistics* 20(1): 129-134.
- Croft, William. 2001. *Radical Construction Grammar*. Oxford: Oxford University Press.
- Del Campo Martínez, Nuria. 2011. Cognitive modeling in illocutionary meaning. *Review of Cognitive Linguistics* 9(2): 392-412
- Goldberg, Adele. 1995. *A Construction Grammar Approach to Argument Structure*. Chicago: University of Chicago Press.
- Goldberg, Adele. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford: Oxford University Press.
- Halvorsen, Per-Kristian. 1988. Computer applications of linguistic theory. In *Linguistics: The Cambridge Survey. II Linguistic Theory: Extensions and Implications*, Frederick Newmeyer (ed), 198-219. Cambridge: Cambridge University Press.
- Jackendoff, Ray. 1997. Twistin' the night away. *Language* 73: 534-559.
- Lakoff, George. 1977. Linguistic gestalts. *Chicago Linguistic Society* 13: 236-287.
- Mairal-Usón, Ricardo & Perrián-Pascual, Carlos. 2009. The anatomy of the lexicon component within the framework of a conceptual knowledge base. *Revista Española de Lingüística Aplicada* 22: 217-244.

- Mairal-Usón, Ricardo, Perriñán-Pascual, Carlos & Pérez Cabello de Alba, María Beatriz. 2012. La representación léxica. Hacia un enfoque ontológico. In *El Funcionalismo en la Teoría Lingüística: La Gramática del Papel y la Referencia*, Ricardo Mairal-Usón, Lilián Guerrero & Carlos González Vergara (eds), 85-102. Madrid: Akal.
- Mairal-Usón, Ricardo & Ruiz de Mendoza, Francisco J. 2008. New challenges for lexical representation within the Lexical-Constructional Model (LCM). *Revista Canaria de Estudios Ingleses* 57: 137-158.
- Mairal-Usón, Ricardo & Ruiz de Mendoza, Francisco J. 2009. Levels of description and explanation in meaning construction. In *Deconstructing Constructions*, Christopher Butler & Javier Martín Arista (eds), 153-198. Amsterdam/Philadelphia: John Benjamins.
- Peña Cervel, Sandra. 2009. Constraints on subsumption in the caused-motion construction. *Language Sciences* 31, 740-765.
- Pérez Hernández, Lorena & Peña Cervel, Sandra. 2009. Pragmatic and cognitive constraints on lexical-constructional subsumption. *Atlantis* 31(2): 57-73.
- Perriñán-Pascual, Carlos & Arcas-Túnez, Francisco. 2004. Meaning postulates in a lexico-conceptual knowledge base. In *Proceedings of the 15th International Workshop on Databases and Expert Systems Applications*, 38-42. Los Alamitos: IEEE Computer Society.

- Periñán-Pascual, Carlos & Arcas-Túnez, Francisco. 2005. Microconceptual-Knowledge Spreading in FunGramKB. In *Proceedings of the 9th IASTED International Conference on Artificial Intelligence and Soft Computing*, 239-244. Anaheim-Calgary-Zurich: ACTA Press.
- Periñán-Pascual, Carlos & Arcas-Túnez, Francisco. 2007. Cognitive modules of an NLP knowledge base for language understanding. *Procesamiento del Lenguaje Natural* 39: 197-204.
- Periñán-Pascual, Carlos & Arcas-Túnez, Francisco. 2008. A cognitive approach to qualities for NLP. *Procesamiento del Lenguaje Natural* 41: 137-144.
- Periñán-Pascual, Carlos & Arcas-Túnez, Francisco. 2010a. Ontological commitments in FunGramKB. *Procesamiento del Lenguaje Natural* 44: 27-34.
- Periñán-Pascual, Carlos & Arcas-Túnez, Francisco. 2010b. The architecture of FunGramKB. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, 2667-2674. Malta: European Language Resources Association.
- Periñán-Pascual, Carlos & Arcas-Túnez, Francisco. In press. The implementation of the FunGramKB CLS Constructor. In *Language Processing and Grammars: The Role of Functionally Oriented Computational Models*, Carlos Periñán-Pascual & Brian Nolan (eds). Amsterdam/Philadelphia: John Benjamins.

- Periñán-Pascual, Carlos & Mairal-Usón, Ricardo. 2009. Bringing Role and Reference Grammar to natural language understanding. *Procesamiento del Lenguaje Natural* 43: 265-273.
- Periñán-Pascual, Carlos & Mairal-Usón, Ricardo. 2010. La gramática de COREL: un lenguaje de representación conceptual. *Onomázein* 21: 11-45.
- Periñán-Pascual, Carlos & Mairal-Usón, Ricardo. 2011. The COHERENT methodology in FunGramKB. *Onomázein* 24: 13-33.
- Rappaport Hovav, Malka & Levin, Beth. 1998. Building verb meanings. In *The Projection of Arguments: Lexical and Compositional Factors*, Miriam Butt & Wilhelm Geuder (eds), 97-134. Stanford: CSLI Publications.
- Raskin, Victor. 1987. Linguistics and natural language processing. In *Machine Translation: Theoretical and Methodological Issues*, Sergei Nirenburg (ed), 42-58. Cambridge: Cambridge University Press.
- Ruiz de Mendoza, Francisco J. Meaning construction, meaning interpretation and formal expression in the Lexical Constructional Model. To appear in this volume.
- Ruiz de Mendoza, Francisco J. & González, Francisco. 2011. Constructional integration in the Lexical Constructional Model. *British and American Studies* 17: 75-95.
- Ruiz de Mendoza, Francisco J. & Luzondo Oyón, Alba. In press. Lexical-constructional subsumption in resultative constructions in English. In

Cognitive Linguistics. Between Universality and Variation, Mario Brdar, Milena Zic Fuchs, Ida Raffaelli, Mateusz-Milan Stanojevic, Nina Tudjman Vukovic (eds). Newcastle: Cambridge Scholars Publishing.

Ruiz de Mendoza, Francisco J. & Mairal-Usón, Ricardo. 2008. Levels of description and constraining factors in meaning construction: an introduction to the Lexical Constructional Model. *Folia Linguistica* 42(2): 355-400.

Sugayama, Kensei. 2011. The main determinants of sentence meaning: verbs or constructions? In *Proceedings of the 12th Chinese Lexical Semantics Workshop*, 58-66. Taipei.

Van Valin, Robert D. Jr. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge: Cambridge University Press.

Van Valin, Robert D. Jr. & LaPolla, Randy J. 1997. *Syntax, Structure, Meaning and Function*. Cambridge: Cambridge University Press.

Winograd, Terry. 1972. *Understanding Natural Language*. New York/London: Academic Press.

Appendix 1. FunGramKB ARTEMIS interface.

Appendix 2. Graphical representation of the parse tree of the sentence

“The juice froze black in the refrigerator”.

Appendix 3. Bracketed representation of the parse tree of the sentence

“The juice froze black in the refrigerator”.

Figure 1.

a) CLS:

Aktionsart: ACC Variables: y,w

Variable: w

Thematic role: Result

Macrorole:

Phrases: ADJP, PP

Syntax: Nucleus

Prepositions: into

Preferences:

b) COREL scheme:

```
+(e1: <EVENT> (f1: (e2: +BECOME_00 (x1: y)Theme (x2: w)Attribute))Result)
```

Done

Figure 2.

L1-constr	Type	RESI			
	CLS	Aktionsart	ACC		
		Variables	Type	y	
Type	w				
Role	result				
Phrase	ADJP PP				
Syntax	nucleus				
Prep	into				
	COREL scheme	+(e1: <EVENT> (f1: (e2: +BECOME_00 (x1: y)Theme (x2: w)Attribute))Result)			

Figure 3.

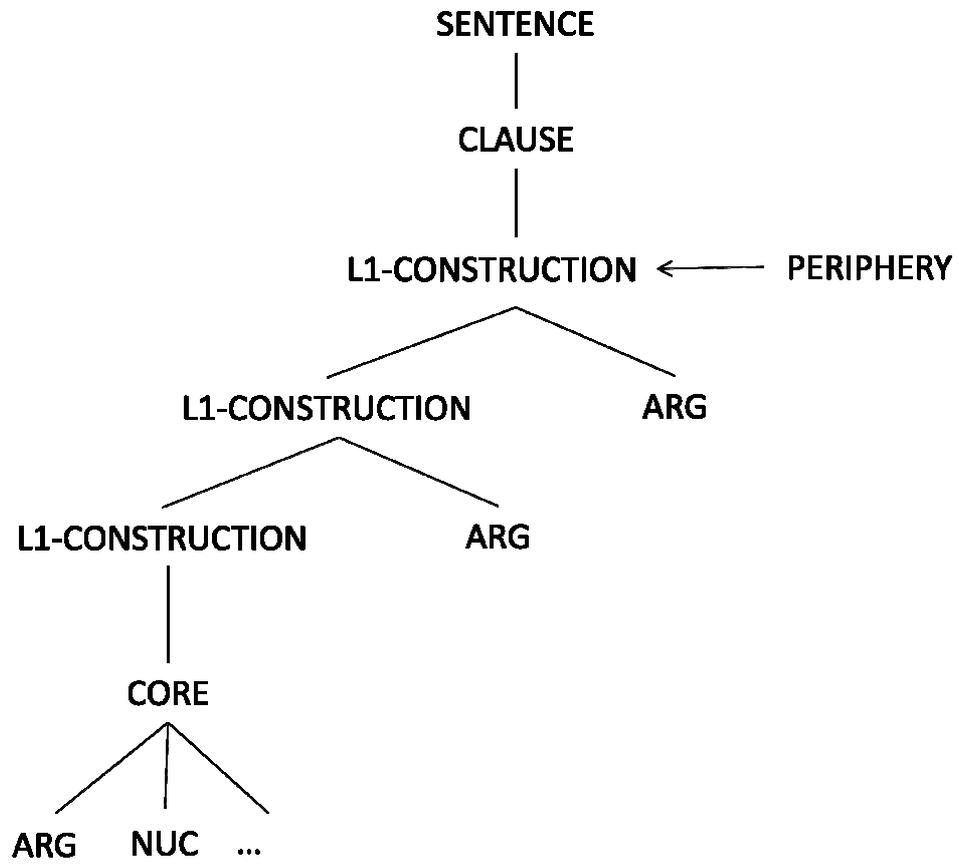
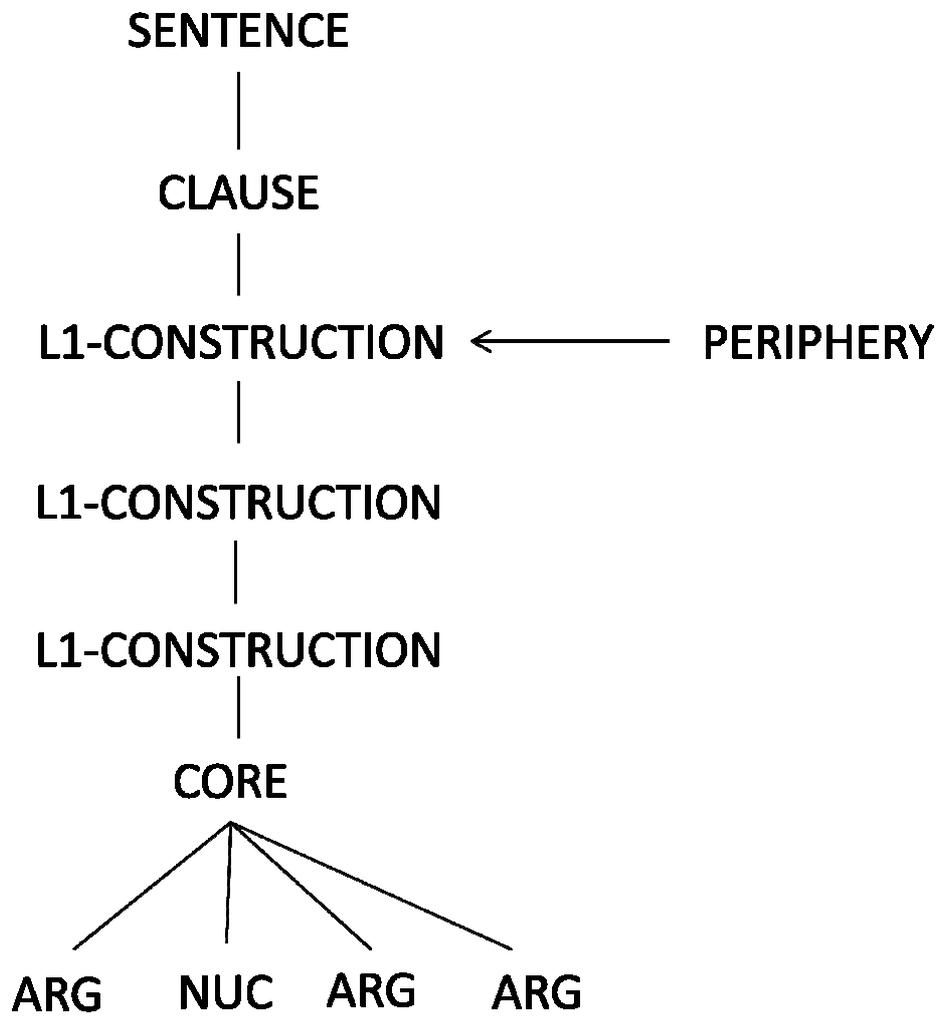


Figure 4.





FunGram KB ART E M I S

Input:
the juice froze black in the refrigerator

Reset

Parse tree:

```
Tree 01
(S)
  (CL[Tense='past']
    (CONSTR-L1[Akt=ACC, Template='REST', Tense='past', Weight=3]
      (CONSTR-L1[Act=ACC, Template='INCH', Tense='past', Weight=2]
        (CORE[Template='INCH', Tense='past']
          (ARG[Macrorole='U', Phrase='NP', Role='Referent', Type='f']
            [+REFRIGERATOR_00-n]
          )
        )
      )
    )
  )

```

Constructional priority Tree refinement

```
S
  CL [Tense=past]
    CONSTR-L1 [Akt=ACC, Template=REST, Tense=past, Weight=3]
      CONSTR-L1 [Akt=ACC, Template=INCH, Tense=past, Weight=2]
        CORE [Template=INCH, Tense=past]

```

CLS:

```
<F DECL <Tense PAST <CONSTR-L1 RES1 <CONSTR-L1 INCH <AKT ACC [+FREEZE_00n(+JUICE_00-Referent, +BLACK_00)] (+REFRIGERATOR_00-n) >>>>
```

COREL scheme:

Grammar Development Environment:

```
Syntaxic rules
# start S
#####
# Grammar Productions
#####
S -> CL
CL[Tense=?t] -> CONSTR-L1[Tense=?t] | CONSTR-L1[Tense=?t] PER
#####
## [add constructional rules] ##
PER -> ADJUNCT | ADJUNCT ADJUNCT | ADJUNCT ADJUNCT ADJUNCT
ADJUNCT [Phrase=ADVf] -> ADVP
ADJUNCT [Phrase=PP, p=?p] -> NUC-A[p=?p] ARG[Type=x, Phrase=NP, Role=n, Macrorole=n]
NUC-A[p=?p] -> FRED-A[p=?p]
FRED-A[p=?p] -> p[p=?p]
```

Save

Download

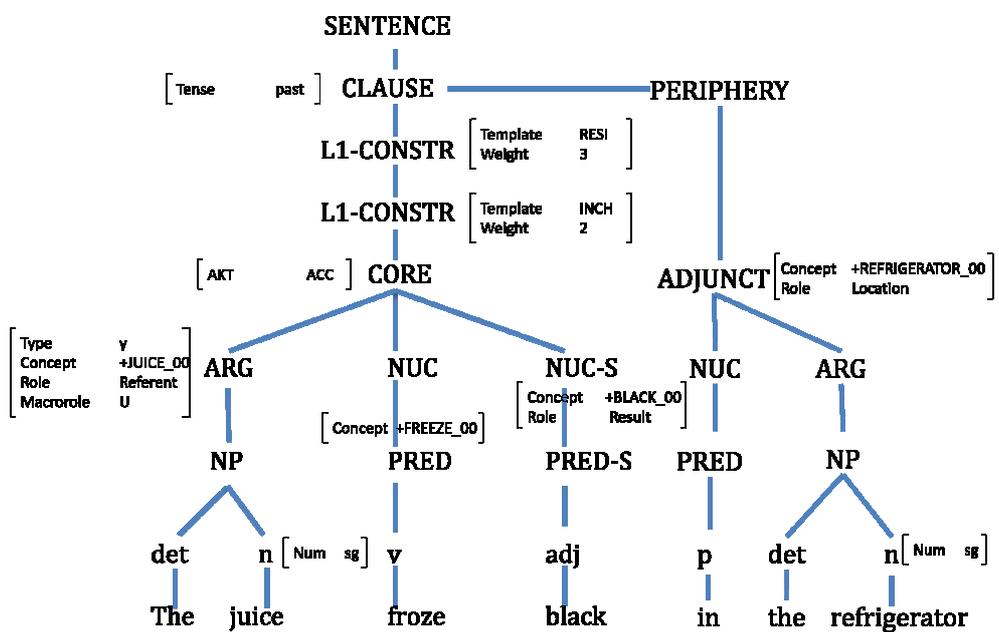
Constructional rules

```
CONSTR-L1[Tense=?t, Template=KER2, Akt=CGCC, Weight=4] -> CORE [Tense=?t, Template=KER2]
CORE [Tense=?t, Template=KER2] > ARG[Num=?n, Type=x, Phrase=NP, Role=Referent, Macrorole=U]
NUC [Tense=?t, Num=?n, Template=KER2] ARG[Type=?y, Phrase=NP, Role=Referent, Macrorole=U]
CONSTR-L1[Tense=?t, Template=INCH, Akt=ACC, Weight=2] -> CORE [Tense=?t, Template=INCH]
CORE [Tense=?t, Template=INCH] -> ARG [Num=?n, Type=y, Phrase=NP, Role=Referent, Macrorole=U]
NUC [Tense=?t, Num=?n, Template=INCH]
CONSTR-L1 [Tense=?t, Template=MIDD, Akt=STA, Weight=3] -> CORE [Tense=?t, Template=MIDD] ARG [Type=y, Phrase=ADVP, Role=Manner, Macrorole=n] | CONSTR-L1 [Tense=?t] ARG [Type=y, Phrase=ADVP, Role=Manner,
```

lexical rules

```
n[Num=?g, Concept='+JUICE_00'] -> 'juice'
v[Tense=past, Template=KER2, Concept='+FREEZE_00'] -> 'froze'
v[Tense=past, Template=INCH, Concept='+FREEZE_00'] -> 'froze'
```

Appendix 2.



Appendix 3.

```

(S[]
  (CL[Tense='past']
    (CONSTR-L1[Template='RESI', Weight=3]
      (CONSTR-L1[Template='INCH', Weight=2]
        (CORE[Akt='ACC']
          (ARG[Type='y', Concept='-JUICE_00', Role='Referent', Macrorole='U']
            (NP[]
              (det[] the)
              (n[Num='sg'] juice)))
            (NUC[]
              (PRED[Concept='-FREEZE_00']
                (v[] froze))))))
          (NUC-S[]
            (PRED-S[Concept='-BLACK_00', Role='Result']
              (adj[] black))))))
    (PER[]
      (ADJUNCT[Concept='-REFRIGERATOR_00', Role='Location']
        (NUC-A[]
          (PRED-A[]
            (p[] in)))
          (ARG[]
            (NP[]
              (det[] the)
              (n[Num='sg'] refrigerator)))))))))

```