Revista de LINGÜÍSTICA Y LENGUAS APLICADAS

Revista de Lingüística y Lenguas Aplicadas
Vol. 12 año 2017, 33-44
EISSN 1886-6298
https://doi.org/10.4995/rlyla.2017.5406

THE INTERACTION OF PARSING RULES AND ARGUMENT – PREDICATE CONSTRUCTIONS: IMPLICATIONS FOR THE STRUCTURE OF THE GRAMMATICON IN FUNGRAMKB[1]

María del Carmen Fumero Pérez
Ana Díaz Galán
Universidad de La Laguna

Abstract: *The Functional Grammar Knowledge Base (FunGramKB), (Periñán-Pascual and Arcas-Túnez 2010) is a multipurpose lexico-conceptual knowledge base designed to be used in different Natural Language Processing (NLP) tasks. It is complemented with the ARTEMIS (Automatically Representing Text Meaning via an Interlingua–based System) application, a parsing device linguistically grounded on Role and Reference Grammar (RRG) that transduces natural language fragments into their corresponding grammatical and semantic structures. This paper unveils the different phases involved in its parsing routine, paying special attention to the treatment of argumental constructions. As an illustrative case, we will follow all the steps necessary to effectively parse a For-Benefactive structure within ARTEMIS. This methodology will reveal the necessity to distinguish between Kernel constructs and L1-constructions, since the latter involve a modification of the lexical template of the verb. Our definition of L1-constructions leads to the reorganization of the catalogue of FunGramKB L1-constructions, formerly based on Levin's (1993) alternations. Accordingly, a rearrangement of the internal configuration of the L1-Constructicon within the Grammaticon is proposed.*

Keywords: *FunGramKB, NLP, Parsing, ARTEMIS, Argumental Construction, computational implementation of RRG.*

1. INTRODUCTION

The automatic treatment of language has been one of the outputs of the great development that the field of computational technology has undergone in the last decades. The web era has generated the need for text processing technologies to perform tasks which have now become part of our daily routine. Such common actions as searching the Wikipedia or Google, requesting information from a virtual digital assistant (Siri, Cortana), translating (Google translator, Bing), etc. are possible due to the applications developed in the field of Natural Language Processing (NLP). Within NLP, these activities fall into what is known as data mining, language interface and machine translation, respectively. The increasing demand for text processing applications of this kind calls for rapid and effective solutions in the treatment of natural language; this has resulted in a dominance of approaches which focus on statistical methods. While the advantage of stochastic models is their immediate applicability, one of the frequently mentioned drawbacks of these methods is that they are not based on a theory of linguistic representation. Periñán-Pascual (2012), citing Wintner (2009) and Ferrari (2004), discusses the main reasons that explain the detachment between NLP and linguistics. He argues that from the point of view of computer engineers, the work of linguists has been, in general, too theoretical and difficult to represent formally, thus, lacking practical applicability. This eminently theoretical nature of linguistic research makes it a long term process, not meeting, therefore, the demand for immediacy imposed by the information technology industry.

---

[1] Este trabajo ha sido desarrollado en el contexto del proyecto de investigación: "Desarrollo de un laboratorio virtual para el procesamiento computacional de la lengua desde un paradigma funcional". (UNED) FF2014-53788-C3-1-P.

To cite this article: Fumero Pérez, M. C., Díaz Galán, A. (2017). "The interaction of parsing rules and argument – predicate constructions: implications for the structure of the grammaticon in FunGramKB". *Revista de Lingüística y Lenguas Aplicadas*, *12*, 33-44. https://doi.org/10.4995/rlyla.2017.5406

Received: 2016-04-25 Accepted: 2017-03-14

What would be desirable is a greater interaction and collaboration between linguists -often reluctant to formalize their theories- and computer engineers so that they could benefit from each other's knowledge. It is this type of interdisciplinary approach that lies behind the project within which the present study is developed, the Functional Grammar Knowledge Base (FunGramKB) (Periñán-Pascual and Arcas-Túnez, 2014). Two are the main assets of this multifunctional and multilingual knowledge base:

a. It has a strong linguistic component based on two sound functional models of language: Role and Reference Grammar (RRG) (Van Valin and La Polla, 1997; Van Valin, 2005) and the Lexical Constructional Model (LCM) (Ruiz de Mendoza and Mairal Usón, 2008; Mairal Usón and Ruiz de Mendoza, 2009).

b. It is formalized through its own representation language, COREL (Conceptual Representation Language), a notational system with great expressive power which allows the representation of deep semantics.

Periñán-Pascual and Arcas-Túnez (2010:2667) describe FunGramKB as "a user-friendly online environment for the semiautomatic construction of a multipurpose lexico-conceptual knowledge base for natural language processing (NLP) systems, and more particularly for natural language understanding". Three knowledge levels, namely, conceptual, lexical and grammatical constitute the main components of FunGramKB, as we can see in Figure 1.

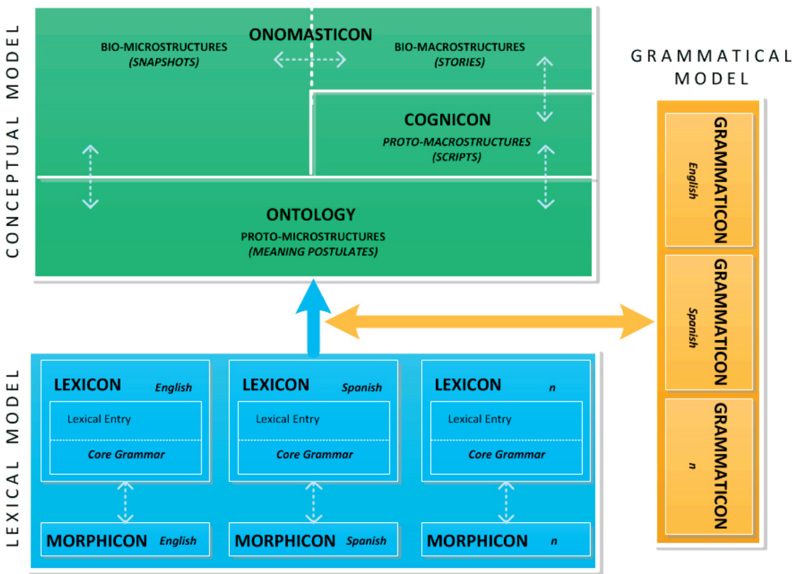

Figure 1. The architecture of FunGramKB (source: www.fungramkb.com).

Conceptual knowledge, which is language independent, is stored in three different modules: the Onomasticon, which includes encyclopedic knowledge; the Cognicon, where procedural knowledge is captured, and, finally, the Core Ontology, which constitutes the central module of the whole system, stores semantic knowledge in the form of meaning postulates (MPs), as is shown in the following entry for the concept +BAKE_00.

(1) +BAKE_00
MP: +(e1: +CHANGE_00 (x1)$_{THEME}$ (x2)$_{REFERENT}$ (f1: (e2: +BECOME_00 (x2)$_{THEME}$ (x3: +HOT_00) $_{ATTRIBUTE}$)) $_{RESULT}$)[2]

In FunGramKB, the language dependent modules are the Lexicon and the Grammaticon. The Lexicon –a compilation of the lexical units of a specific language (so far, English, Spanish, French and Italian)– provides morphosyntactic, pragmatic and collocational information about these units. The Grammaticon –an inventory of constructional schemata– has four Constructicon modules, which reflect the four layers distinguished in the LCM, that is, argumental (L1-Constructicon), implicational (L2-Constructicon), illocutionary (L3-Constructicon) and discursive (L4-Constructicon).

To this date, there are three applications that exploit FunGramKB: the FunGramKB NAVIGATOR, which allows the user to retrieve data from the lexical entries in the English Lexicon and from the conceptual entries in the Core Ontology; DEXTER (Discovering and Extracting Terminology), an online multilingual workbench which is provided with a suite of tools for the compilation and management of small and medium-sized corpora, and ARTEMIS (Automatically Representing Text Meaning via an Interlingua–based System), a parsing device which

[2] In natural language the MP of the concept +BAKE_00 would read as "a participant (x1) changes something (x2), with the result (f1) that (x2) becomes very hot".

is grounded on RRG as its linguistic model and uses FunGramKB as its knowledge base. Its objective is the simulation of natural language understanding by binding natural language fragments with their corresponding grammatical and semantic structures, in other words, it seeks to obtain the appropriate syntactic and semantic underlying representations of any piece of natural language. In order to fulfill this goal, ARTEMIS must resort to the grammatical and semantic material that appears distributed both within and without the prototype.

In the computational representation of natural language, an account of constructional meaning is fundamental, since it can override core verbal semantics and have a direct influence on syntax. Nevertheless, it is only in recent years that we find literature which gives account of studies that attempt to go beyond theoretical descriptions and treat constructions computationally. We may cite those mentioned by Steels (2015) as examples of computationally driven approaches to constructions: Embodied Construction Grammar, Fluid Construction Grammar, Signed-Based Construction Grammar and Template Construction Grammar. Within the FungramKB environment, there are also recent studies such as Luzondo-Oyón and Ruiz de Mendoza (2015) which address this issue. Our paper contributes to this same line of research by adapting the notion of construction to make it computationally effective, trying to arrive at compromise solutions in order to fully articulate the process of identification of constructional structures in FunGramKB and ARTEMIS.

We consider it essential to establish what should count as an argumental construction in FunGramKB, as it will not necessarily encode what may be understood as argumental construction outside our model. This paper will reassess the validity of such structures as FunGramKB constructions and propose a rearrangement of the internal configuration of the L1-Constructicon, which may lead to the dismissal of some constructions, the addition of new ones or the reinterpretation of others. In doing so, we will try to contribute to the reconciliation between NLP and linguistics by presenting a detailed methodology of the parsing process in ARTEMIS, which involves bringing into scene the role of parsing rules in combination with other sources of grammatical information (lexica and constructicons).

In Section 2 we discuss the treatment of constructional meaning in FunGramKB. Section 3 is dedicated to the description of ARTEMIS, and section 4 illustrates, through the analysis of a sample sentence, the actual parsing routine which ARTEMIS follows to arrive at the semantic representation of a construction, in our case, the For-Benefactive Construction. Finally, the practical and theoretical implications of this analysis are summarized in the concluding section.

2. ARGUMENTAL CONSTRUCTIONS IN FUNGRAMKB

An NLP computational model which aims at a satisfactory semantic representation of the sentence must account not only for lexical meaning, derived from the lexical entries of verbs, but also for constructional meaning as it can influence both the syntactic behavior and the core meaning of a verb. To achieve this aim, FunGramKB has adopted a hybrid approach which combines principles from RRG and the LCM. The projectionist nature of RRG allows the connection between syntax and the lexical entries of verbs contained in the Lexicon, while the constructionist model of language proposed by the LCM provides the constructional meaning which is not present in RRG. With this in mind, Periñán-Pascual (2013:214) adopts a conception of construction which separates from the broader notion defended by Goldberg (2006:5):

> 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)

Periñán-Pascual defends a narrower view of construction in which productivity, bi-univocity and replicability are considered crucial properties to decide the constructional nature of a form-meaning pairing, in line with the usage-based notion of construction proposed by 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. (Ruiz de Mendoza, 2013:238)

Furthermore, Periñán-Pascual and Arcas-Túnez (2014:172) consider that a computational approach to the notion of construction makes it necessary to establish a precise distinction between the concepts *construct* and *construction*, which they express in the following way:

> In terms of the FunGramKB model, lexical constructs get their meaning from the meaning postulates stored in the Ontology, whereas constructional meaning is shaped by the Core Grammar in the Lexicon and the constructional schemata in the Grammaticon. (Periñán-Pascual and Arcas-Túnez, 2014:172-173)

The term *construction* is used within FunGramKB in a very restrictive manner if compared with its usage in standard linguistic (especially constructional) models, as we can see in the following definition given by Periñán-Pascual and Arcas-Túnez (2014:172) "[…] a pairing of form and meaning, serving as a building block in the compositionality of sentential semantics, whose meaning cannot be fully derived from the sum of the lexical meanings of the individual constructs taking part in the utterance".

The consequences of the distinction between construct and construction are significant in the design of the L1-Constructicon (argumental level), since only the structures whose grammatical and/or semantic features are not obtained from the grammatical and semantic contribution of its members/components are to be considered as candidates for membership in the constructional module.

Although FunGramKB distinguishes four different constructional levels, at this stage of development of the ARTEMIS parser, only level 1 argumental constructions (CONSTR-L1) can be accounted for. These constructions, based on the linguistic expression of predicate-argument relationships (Goldberg, 1995), are associated with Level 1 constructional templates in the LCM; such structures are made up of elements of semantic interpretation that can be realized syntactically (Mairal Usón and Ruiz de Mendoza, 2009). Examples of these types of constructions as found in the FunGramKB Constructicon are:

(2) The president appointed Smith as press secretary. (As Construction)

(3) Peter sneezed the napkin off the table. (Caused Motion Construction)

(4) Kelly ate herself sick. (Resultative Construction)

Argumental information in FunGramKB can be retrieved from two sources: the Core Grammar in the Lexicon or the Grammaticon. The Core Grammar stores the information related to what Periñán-Pascual calls *Kernel constructions*, in such a way that every verb listed in the Lexicon is provided with one such Kernel construction. Depending on the number of variables in the lexical template, the verb will occur in a Kernel-0 (zero-argument verbs), Kernel-1 (intransitive), Kernel-2 (monotransitive) or Kernel-3 (ditransitive). As we can see in Figure 2, a simplified representation, the Core Grammar of a verb such as *bake* would correspond to a Kernel-2 (monotransitive verb), presenting, therefore, two variables x (theme) and y (referent) as the following example taken from FunGramKB illustrates:

(5) …we $_{\text{(THEME)}}$ also bake a selection of other French breads and pastries $_{\text{(REFERENT)}}$ (13/CFS/W_advert)

Lexicon	English
Lexical unit	bake
Ontological concept	+BAKE_00
Aktionsart	Active Accomplishment (ACC)
Variables	x, y
Macroles	2, Undergoer: no value selected
Thematic frame mapping	x: theme, y: referent
Constructions:	*Benefactive Object Construction*
	For Benefactive Construction
	Instrument Subject Construction
	Material Subject Construction
	Unexpressed Second Argument Construction

Figure 2. Simplified representation of the Core Grammar of *bake.*
(source: http://www.fungramkb.com//edit/lexicon//lexicon.aspx)

In our opinion, this use of the term *Kernel construction*, reveals somewhat contradictory in relation to the distinction made by Periñán-Pascual and Arcas-Túnez between construct and construction, which we cited above. In this respect, we share the position defended by Luzondo-Oyón and Ruiz de Mendoza (2015:79) that it would be more appropriate to use the term *Kernel construct* rather than *construction* to refer to these structures, since they are not of compositional nature and their meaning resides in the Core Grammar in the Lexicon. This would allow us to make a distinction between this type of structures and *Non-Kernel argumental constructions*, those not accounted for in the Lexicon, but stored and described in the form of constructional schemata in the L1-Constructicon in the Grammaticon. This distinction would also be more in line with Periñán-Pascual's (2013:215) own assertion that although constructs can be either constructional or not, it is preferable to use the label *construct* to refer to non-constructional structures and *construction* for those of constructional nature.

What we propose here is to consider as L1-construction only those constructs which alter the Core Grammar of the verb. This means adopting a restrictive notion of L1-construction within FunGramKB to meet the requirements of their computational treatment. It is important to highlight that this means no challenge against the concept of construction as understood in linguistic models, typically the different variants of construction grammars, including here the LCM. The importance of L1 or non-Kernel constructions from a computational perspective is that they have direct impact on the formal representation of meaning in ARTEMIS, the so-called Conceptual Logical Structure (CLS), since, as we will prove, they either modify the argumental structure by substracting arguments or adding non-optional constituents (Argument Adjuncts or Secondary Nuclei) and/or by changing aspectual meaning (Aktionsart).

This involves a drastically different methodology from the one originally adopted in the design of the L1-Constructicon. The inventory of non-Kernel L1-constructions was initially based on Levin's (1993) alternations, with some additional constructions from other sources. In its present stage, however, alternations have been substituted for constructional schema, the rationale for this change being that alternations focus on the modification of an input sentence pattern into a different one, while what the parser needs is the actual description of each of the constructional patterns in which a verb can enter and a pointer in the Lexicon to lead the parser to the description of these constructions in the Grammaticon.

In order to show how meaning is encoded in ARTEMIS, in the following section we will describe its parsing routine and show the importance of constructional meaning in this process. In our view, the requirements of such a codification confirm the need to modify the concept of construction and, consequently, alter the catalogue of FunGramKB L1-constructions.

3. GRAMMATICAL ANALYSIS WITHIN ARTEMIS

The ARTEMIS application is, technically, a bottom-up chart parser with top-down prediction which transforms or transduces the natural language input it receives to its equivalent grammatical and semantic structures (Periñán-Pascual and Arcas-Túnez, 2014). The parser consists of three components: a) The Grammar Development Environment (GDE) b) The Conceptual Logical Structure (CLS) Constructor and c) The COREL-Scheme Builder.

The GDE comprises the set of rules (syntactic, constructional and lexical) necessary for the parsing of natural language expressions. Such rules yield, as a result, a parsed tree (illustrated in Figure 3) following the format, with some modifications, of the Layered Structure of the Clause (LSC) as proposed in RRG. The most relevant variation is the introduction of an L1-construction node between the clause and the core, which will account for those arguments that are not directly derivable from the main predicate argument's structure. The effects of constructional interaction with verbal predicates will be considered below.

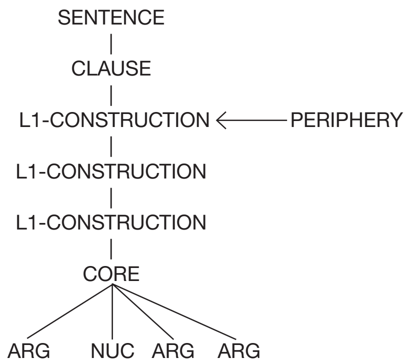

Figure 3. Enhanced model of the Layered Structure of the Clause (refined tree).

The ultimate aim of ARTEMIS is to transduce meaning into a CLS, the text meaning representation which results from the modification of the RRG logical structure. The changes made to this logical structure result in the enrichment of the representation by a) using FunGramKB ontological concepts instead of predicates, b) assigning the concepts a thematic role, c) specifying the type of event with an Aktionsart operator and, lastly, d) encoding constructional meaning through constructional operators (CONSTR-L1). Finally, the CLS is transduced into a purely semantic conceptual representation in COREL, the formal FunGramKB representation language, the so called extended COREL scheme. Example sentence 6 below illustrates the whole process:

(6) Louise had baked a cake for the kids.

RRG Logical structure:

(7) [[do´(Louise, Ø)] CAUSE [BECOME baked´(cake)]] PURP [BECAME have´(the kids, cake)]

CLS:

(8) $<_{IF}^{DECL}<_{TENSE}^{PAST}<_{CONSTR\_L1}^{FBEN}<_{CONSTR\_L1}^{KER2}<_{AKT}^{CACC}$<[+BAKE_00(%LOUISE_00-Agent, +CAKE_00-Referent, +CHILD_00-Beneficiary)] >>>>>>

Extended COREL scheme:

(9) +(e1: +BAKE_00 (x1: %LOUISE_00)$_{THEME}$ (x2: +CAKE_00)$_{REFERENT}$ (f1: (e2: +DO_00 (x1)$_{AGENT}$ (e1)$_{REFERENT}$ (f2: +CHILD_00)Beneficiary))Purpose)

In what follows, we will spell out the process followed by the CLS constructor to implement the syntax-semantics linking algorithm within FunGramKB, as described by Periñán-Pascual and Arcas-Túnez (2014:177ff). To describe this procedure and how constructional meaning is incorporated we will use sample sentence (6), which features a Kernel 2 and what we will label a For-Benefactive construction (FBEN). This example will allow us to show, in a stepwise fashion, how the grammatical information encoded in the sentence is obtained by means of the interplay of the different components involved in the parsing procedure. The significance of this sample sentence lies in the fact that, to codify it, ARTEMIS has to be able to understand that the structure overrides the information provided by the Core Grammar (see Figure 2) in which *bake* only has two arguments x (theme) and y (referent), not justifying, therefore, the presence of a third constituent, *for the kids*. In RRG terms (Van Valin and La Polla, 1997), this third element meets the requirements for its analysis as an argument adjunct, namely, that the logical structure of *for* shares an argument, *cake*, with the logical structure of *bake*, as we can see in example 7 above.

In the next section we will provide a detailed description of the steps followed by the parser to account for the presence of this argument adjunct introduced by the FBEN construction, as a demonstration of how the machine processes constructional meaning.

4. A STEP BY STEP METHODOLOGY

The phases involved in the syntax-semantics linking algorithm are outlined in Figure 4 taken from Periñán-Pascual and Arcas-Túnez (2014:178).

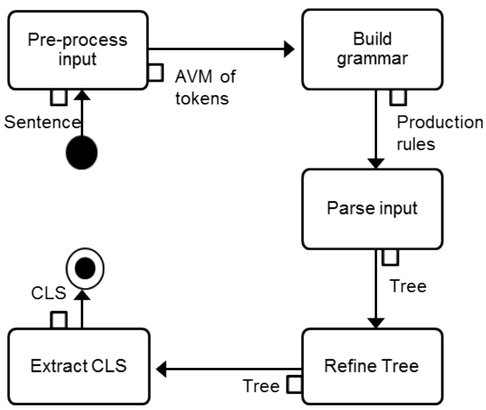

Figure 4. The ARTEMIS process.

The first stage of this process is the AVM extraction for tokens, where the input sentence is automatically separated into basic units of analysis, which are then labelled with a part of speech tag described in the form of feature bearing devices, the so called Attribute Value Matrixes (AVMs)[3], thus, providing the first initial morphosyntactic characterization of the lexical items, as the following AVM for the verb form *baked* illustrates.

[3] In line with Unification Grammars (Boas and Sag, 2012; Sag, Wasow and Bender, 2003), in ARTEMIS parsing operations are guided not only by syntactic rules, but also by constraint-checking operations on the grammatical and semantic information encoded in the AVMs.

Form	baked
Lemma	bake
POS	verb
Tense	past
Concept	+BAKE_00

Figure 5. AVM of the word token *baked.*

The second step applies GDE rules to ascribe an appropriate syntactic/semantic contour to the sentences, which, ultimately, yields a syntactic tree. In order to arrive at the conceptual logical structure of the predicate, ARTEMIS first resorts to the Core Grammar of the verb to check the grammatical information provided in the lexical entry of the corresponding predicate. The AVM in Figure 6 summarizes the information (syntactic and semantic) contained in the Core Grammar of the verb *bake*: the Aktionsart corresponds to an active accomplishment (ACA), and it has two variables x (theme) and y (referent), each with their selection restrictions (theme must be +HUMAN_00 and referent +FOOD_00).

```
[ [Headword      bake ]
  [Core grammar  [CLS   [ Aktionsart  ACA ]
                        [Variables  [ Type      x
                                      Role      theme
                                      Macrorol
                                      Phrase
                                      Syntax
                                      Prefer    +HUMAN_00 ]

                                    [ Type      y
                                      Role      referent
                                      Macrorol
                                      Phrase
                                      Syntax
                                      Prefer    +FOOD_00  ] ] ] ] ]
```

Figure 6. AVM representative of the Core Grammar of *bake.*

The Core Grammar of the verb also indicates the types of constructions in which the verb can appear, in the case of *bake*, apart from the FBEN construction, we find the Unexpressed Second Argument Construction (*The kitchen smelled so wonderful while they were baking*); the Instrument Subject Construction (*This oven bakes wonderful bread*); the Material Subject Construction (*This flour bakes a delicious loaf*) and the Benefactive Object Construction (*She baked them a cake*).

Once the parser has resorted to the Core Grammar to check the information related to the verb, there are two possible outputs:

a. The sentence corresponds to a Kernel, as in example 10:

   (10) Louise had baked a cake (Kernel 2)

b. There is a conflict between the information provided by the Core Grammar (number of variables and Aktionsart) and the structure of the sentence, as occurs with our sample sentence:

   (6) Louise had baked a cake for the kids

In the first case, the parser finishes the grammatical checking, while in the second - to account for the additional constituent *for the kids*- it has to resort to the constructional rules stored in the Grammaticon, where it would need to find a schema corresponding to the FBEN construction such as the one we propose in (Figure 7).

```
L1-constr  [ Type          FBEN ]
        CLS        [ Aktionsart      CACC ]
                   Variables   [ Type          x
                               [ Role          agent ]
                               [ Type          y   ]
                               [ Type          w
                                 Role          beneficiary
                                 Phrase        PP
                                 Syntax        argument adjunct
                                 Prep          for
                                 Prefer        +ORGANISM_00 ]
        [ COREL scheme      *(e1:<EVENT>) (f1:w) Beneficiary) ]
```

Figure 7. AVM representative of the For-Benefactive construction.

In the formalization of argumental constructions we find both descriptors and constraints. As the AVM summarizes, the FBEN construction introduces a new variable (w) which functions as an argument adjunct, has the thematic role of beneficiary, presents a realization restriction, namely, it has to be a prepositional phrase introduced by *for and*, finally, it also presents the selection restriction +ORGANISM_00, which is defined in the Ontology as: "an animal, plant, human or any other living thing". This restriction allows the parser to distinguish, for example, between the FBEN construction and other structures in which we may have *V* + *for* and no beneficiary, as is the case in 11) where the for constituent cannot be a candidate for the function Argument Adjunct (AAJ) since it does not have the restriction +ORGANISM_00 and is, in fact, an optional adjunct which is accounted for in the syntactic rules stored in the GDE:

(11) Louise baked a cake *for dessert*.

It is also important to mention the change in aspectual value, from active accomplishment (ACA) to causative accomplishment (CACC), imposed by the construction. The causative nature of the event introduced by the For-Benefactive Construction is clearly depicted in the RRG representation of the logical structure (LS) of our sample sentence[4]:

(7) LS: [[do´(Louise, Ø)] CAUSE [BECOME baked´(cake)]] PURP [BECAME have´(the kids, cake)]

The causativity imposed by the construction also brings about a change in the thematic role of the (x) argument, which, originally, in the lexical representation of the verb *bake* appeared as theme (Figure 6) and is now an agent (Figure 7). This shift validates the assertion made by Periñán-Pascual and Arcas-Túnez (2014:180) that "specific values in constructional schema override those stated in the lexical entry so non-monotonic inheritance takes place in the projection operation between the Lexicon and the L-1 Constructicon".

The parsing process yields a tree as the one shown in Figure 8. This refined tree shows the addition of an L1-Construction node in the LSC to account for the introduction of a third argumental slot (AAJ) into the core of the clause, which is not contemplated in the original LSC in RRG.

[4] Note that this logical structure also corresponds to the Benefactive Object Construction (for example: "Louise baked the kids a cake"), the only difference being that of the status of the Undergoer (U) in the two constructions. Whereas the For Benefactive Construction follows an unmarked undergoer assignment, the ditransitive construction is the output of assigning undergoer to the possessor constituent "the kids". Parsing this type of sentence would imply exactly the same process as any other construction that involves the addition of a participant. Constructional rules will retrieve both the semantic and syntactic information related to that additional constituent.

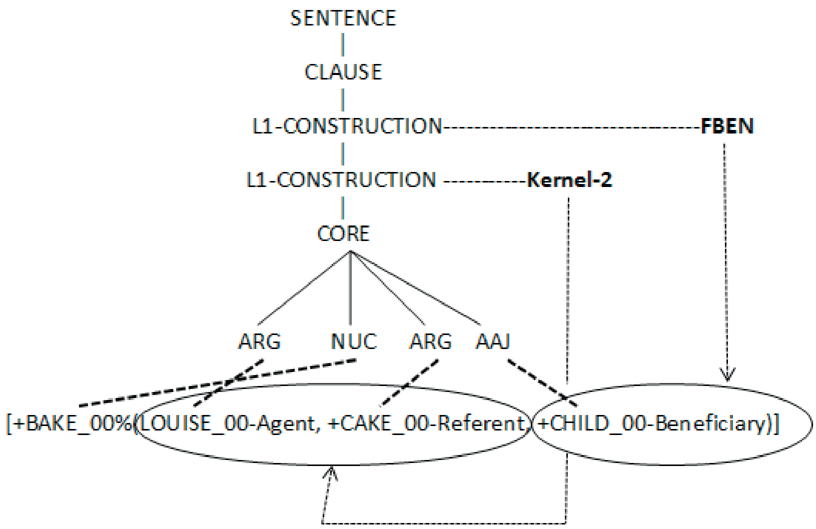

Figure 8. ARTEMIS syntax-semantics linkage.

Although we have used the tree structure representing the syntax-semantics linkage proposed by Periñán-Pascual and Arcas-Túnez (2014:183) for our sample sentence, we believe that to capture the difference which we propose between Kernel construct and L1-construction, it would be desirable to establish a distinction between an L1-construction node and a Kernel construct node, which would always be the one closest to the core.

The parsing routine comes to its end with the extraction of a CLS, as the one shown in 8) for our sample sentence. Starting from the leftmost operator, we learn that the illocutionary force is declarative, the tense is past, there are two L1-constructions, the one we have labelled FBEN and a Kernel 2 (Kernel Construct in our proposal), and the Aktionsart value of the event corresponds to causative accomplishment. Finally, it presents the FunGramKB ontological concepts to which the lexical items of the input sentence are linked, together with their corresponding thematic roles. Example 9 shows the same information in COREL, the FunGramKB representation language:

8) CLS:$<_{IF}^{DECL}<_{TENSE}^{PAST}<_{CONSTR\_L1}^{FBEN}<_{CONSTR\_L1}^{KER2}<_{AKT}^{CACC}<$[+BAKE_00(%LOUISE_00-Agent, +CAKE_00-Referent, +CHILD_00-Beneficiary)] >>>>>>

9) Extended Corel Scheme: +(e1: +BAKE_00 (x1: %LOUISE_00)Theme (x2: +CAKE_00)Referent (f1: (e2: +DO_00 (x1)Agent (e1)Referent (f2: +CHILD_00)Beneficiary))Purpose)

5. CONCLUSION

The aim of the analysis presented has been to offer a detailed methodology of the parsing process in ARTEMIS. In doing so, we have established the difference between Kernel construct and L1-construction. The significance of this distinction is that the former draws from the Core Grammar of the verb, while the latter necessarily involves a modification of the original lexical template. Such an alteration involves a change in the number of arguments and/or a variation in Aktionsart. Thus, the design of ARTEMIS complies with the integrative character of grammatical models which support this NLP prototype, i.e. RRG and the LCM. Both approaches allow a space for the interaction of lexical and constructional information in the construction of the syntactic and the semantic structure of the sentence.

A significant consequence of the methodology underlying the parsing process depicted in this paper has been the necessary reorganization of the L1-Constructicon. Appendix 1 proposes a revised catalogue of argument-predicate constructions within FunGramKB.

We believe that once ARTEMIS is fully implemented and makes feasible the automatic extraction of the CLS underlying any stretch of natural language, it will become a very relevant tool for a variety of NLP tasks.

REFERENCES

Boas, H. and Sag, I. (2012). *Sign-Based Construction Grammar*. Stanford, Cal.: CSLI Publications.

Ferrari, G. (2004). State of the Art in Computational Linguistics. In Van Sterkenburg, P. (ed.) *Linguistics Today: Facing a Greater Challenge*. Amsterdam / Philadelphia: John Benjamins, 163-186. https://doi.org/10.1075/z.126.09fer

Goldberg, A. (1995). *Constructions: A construction grammar approach to argument structure*. Chicago: University of Chicago Press.

Goldberg, A. (2006). *Constructions at Work: the Nature of Generalization in Language*. Oxford: Oxford University Press.

Levin, B. (1993). *English Verb Classes and Alternations. A Preliminary Investigation*. Chicago / London: University of Chicago Press.

Luzondo-Oyón, A. and Ruiz de Mendoza, F. (2015). "Argument structure constructions in a Natural Language Processing environment". *Language Sciences,* 48: 70-89. https://doi.org/10.1016/j.langsci.2015.01.001

Mairal Usón, R. and Ruiz de Mendoza, F. (2009). Levels of description and explanation in meaning construction. In Butler, C. and Martín Arista, J. (eds.) *Deconstructing Constructions*. Amsterdam / Philadelphia: John Benjamins, 153-198.

Mairal Usón, R. and Periñán-Pascual, C. (2009). "The anatomy of the lexicon within the framework of an NLP knowledge base". *RESLA,* 22: 217-244.

Periñán-Pascual, C. (2012). "En defensa del Procesamiento del Lenguaje Natural Fundamentado en la Lingüística Teórica". *ONOMÁZEIN,* 26/2: 13-48.

Periñán-Pascual, C. (2013). Towards a Model of Constructional Meaning for natural Language Understanding. In Nolan, B. and Diedrichsen, E. (eds.) *Linking Constructions into Functional Linguistics: The Role of Constructions in RRG Grammars (Studies in Language Series)*. Amsterdam / Philadelphia: John Benjamins, 205-230. https://doi.org/10.1075/slcs.145.08per

Periñán-Pascual, C. and Arcas-Túnez, F. (2010). 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, C. and Arcas-Túnez, F. (2014). The implementation of the CLS constructor in ARTEMIS. In Nolan, B. and Periñán-Pascual, C. (eds.) *Language Processing and Grammars the role of functionally oriented computational models*. Amsterdam / Philadelphia: John Benjamins, 164-196.

Periñán-Pascual, C. and Mairal Usón, R. (2010). "La gramática de COREL: un lenguaje de representación conceptual The COREL grammar: a conceptual representation language". *ONOMÁZEIN,* 21/1: 11-45.

Ruiz de Mendoza, F. (2013). Meaning construction, meaning interpretation and formal expression in the Lexical Constructional Model. In Nolan, B. and Diedrichsen, E. (eds.) *Linking Constructions into Functional Linguistics: The Role of Constructions in RRG Grammars (Studies in Language Series)*. Amsterdam / Philadelphia: John Benjamins, 231-270. https://doi.org/10.1075/slcs.145.09ib225

Ruiz de Mendoza Ibáñez, F. and Mairal Usón, R. (2008). "Levels of description and constraining factors in meaning construction: an introduction to the Lexical Constructional Model". *Folia Linguistica,* 42/2: 355-400. https://doi.org/10.1515/flin.2008.355

Sag, I, Wasow, T. and Bender, E. (2003). *Syntactic Theory: Formal Introduction*. Stanford: CSLI Publications.

Steels, L. (2015). "Introducing Fluid Construction Grammar". In L. Steels (ed.) *Design Patterns in Fluid Construction Grammar*. Amsterdam: John Benjamins, 3-30.

Van Valin, R. (2005). *Exploring the Syntax-Semantics Interface*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511610578

Van Valin, R. and La Polla, R. (1997). *Syntax*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9781139166799

Wintner, S. (2009). "What science underlies natural language engineering?" *Computational Linguistics,* 35/4: 641-644. https://doi.org/10.1162/coli.2009.35.4.35409

APPENDIX 1: PROPOSAL FOR THE REORGANIZATION OF THE CATALOGUE OF L1-CONSTRUCTIONS

-As Construction

-Benefactive Object Construction

-Causative Construction (accomplishment type)

-Causative Construction (achievement type)

-Causative Motion Structure (atelic)

-Causative Motion Structure (telic)

-Caused Motion Construction

-Cognate Object Construction

-Conative Construction (activity)

-Conative Construction (semelfactive)

-Container Subject Construction

-Creation and Transformational Construction (transitive)

-Ditransitive Construction

-For benefactive construction

-Fulfilling Construction

-Image Impression Construction

-Inchoative Construction

-Instrument Subject Construction

-Intransitive Motion Construction

-Location Object Construction

-Location Subject Construction

-Locative Construction

-Locative Intransitive Construction

-Locative Inversion Construction

-Locatum Subject Construction

-Marked Undergoer Construction

-Material Subject Construction

-Measure Object Insertion (Ceation and Consumption) Construction

-Measure Object Insertion (Motion) Construction

-Middle Construction

-Middle Construction (2)

-Path PP Insertion Construction

-Reaction Object Construction

-Reciprocal Object Construction (transitive)

-Reciprocal Subject Construction

-Reflexive Object Addition Construction

-Resultative Construction (intransitive accomplishments)

-Resultative Construction (intransitive achievements)

-Resultative Construction (non-subcategorized object)

-Resultative Construction (transitive accomplishments)

-Resultative Construction (transitive achievements)

-Split Subject Construction

-Substance / Source Construction

-Substance addition construction

-There- Insertion Construction

-Unexpressed Second Argument Construction

-Unexpressed Third Argument Construction

-Virtual Reflexive Construction

-Way Construction (motion)

-Way Construction (resultative)

APPENDIX 2: LIST OF ABBREVIATIONS

AAJ	Argument-adjunct
ACA	Active Accomplishment
ARTEMIS	Automatically Representing Text Meaning via an Inter-lingua-based system
AVM	Attribute Value Matrix
CACC	Causative Accomplishment
CLS	Conceptual Logical Structure
COREL	Conceptual Representation Language
CONSTR-L1	Level 1 Argumental Construction
DEXTER	Discovering and Extracting Terminology
FBEN	For-Benefactive Construction
FunGramKB	Functional Grammar Knowledge Base
GDE	Grammar Development Environment
LCM	Lexical Constructional Model
LSC	Layered Structure of the Clause
MP	Meaning Postulate
NLP	Natural Language Processing
RRG	Role and Reference Grammar