Towards the Synthesis of Support Verb Constructions: Distribution of Syntactic Actants between the Verb and the Noun

Margarita Alonso Ramos

Universidade da Coruña

Abstract

The syntactic representation of Support Verb Constructions (SVCs) is a challenge for any syntactic theory since, on the one hand, the syntactic actants of a lexeme (and thus also of the support verb) correspond to its semantic actants, and, on the other hand, the support verb is deprived of semantic actants (and, thus, the semantic actants of the predicative noun must be referred to). In this article, we examine the possibilities of the distribution of the syntactic actants (in the Meaning-Text Theory, MTT, of the deep-syntactic actants, DSyntAs) and their correspondence with the semantic actants, SemAs, in the **Oper**-group of Lexical Functions, which model SVCs in MTT. Three options are available: (i) to allow the correspondence between DSynts and SemAs to be variable, (ii) to limit the number of DSyntAs to two, and (iii) to determine the correspondence between DSynts and SemAs depending on the surface valency of the support verb (i.e., the value of the LF in question). We assess these three options, evaluating how intuitive each of them is, how easy each of them can be implemented for synthesis and paraphrasing, etc. Our assessment shows that option (iii) reveals some advantages over the first two options.

1 Introduction

At the end of the eighties I took my first steps in linguistics, guided by hand by Igor Mel'čuk—precisely with a work which focussed on the Spanish support verb DAR. Almost fifteen years later, support verbs continue to be the central topic of our email discussions. I would like to put the blame for that on Igor because he taught me to grapple with a problem until it is solved. The following reflexions are meant to be another effort to bring some light into the problem of support verb modelling, even if I am aware that they offer more problems than solutions.

1.1 *The Object and the Framework of the Study*

This work deals with support verb constructions (SVCs) such as [*to*] *take a walk*, [*to*] *give a kiss*, [*to*] *make a claim*, etc., where the main role of the verb is to provide the syntactic support to the semantic predicate expressed by the noun. In several approaches (see among others, Hausmann, 1989, Fontenelle, 1992, Heid, 1996), support verbs are considered as collocational lexical units, i.e., lexical units that express collocational meanings. The meanings of these verbs can only be represented by very general semantic elements such as 'make' or 'have'.

Collocational lexical units are characterized by the following feature: they are not chosen by the speaker semantically, i.e., due to their meaning, but in accordance with another lexical unit that is semantically chosen. Thus, in a collocation formed by a support verb and a noun that is the direct object of the verb, it is the noun that is chosen for its meaning. For example, wanting to speak about 'walk', the speaker would choose the noun WALK to express this meaning, but he would not choose the verb [*to*] TAKE to express an independent meaning: only when he realizes that he needs a verb to construct the sentence, would he choose this verb to syntactically support the sentence. If he begins with another noun, his choice will be different: WALK combines with TAKE, but JOURNEY with MAKE. The choice will also depend on the language under consideration. In Spanish, the noun PASEO 'walk' combines with DAR 'give' and not with TOMAR 'take', while in French, the equivalent of this noun, PROMENADE, selects FAIRE 'do'.

In the *Meaning-Text Theory* (MTT), collocations are described by *lexical functions* (LFs). An LF provides a lexical unit that expresses a specific collocational meaning contingent on its argument lexical unit. The lexical unit to which the LF is applied is called its *keyword*, and the collocational lexical unit provided by the LF is called its *value*. For details on LFs, see (Mel'čuk, 1996, 1998). In order to illustrate the notion of the LFs, we cite examples of support verb LFs of three different types:[1]

- $\mathbf{Oper_i}$: support verb that takes the keyword as its first complement:[2]
  - $\mathbf{Oper_1}$(*walk*) = [*to*] *take* [ART ~]
  - $\mathbf{Oper_1}$(*defeat*) = [*to*] *inflict* [ART ~ *on* N]
  - $\mathbf{Oper_2}$(*defeat*) = [*to*] *suffer* [~ *at* N's *hands*]
- $\mathbf{Labor_{ij}}$: support verb that takes the keyword as its second complement:

$Labor_{12}$(*interrogation*) = [*to*] *subject* [N *to* ART ~]
$Labor_{12}$(*consideration*) = [*to*] *take* [N *into* ~]
$Labor_{12}$(*loan*) = [*to*] *give* [N *on* ~]
$Labor_{32}$(*loan*) = [*to*] *get*, [*to*] *receive* [N *on* ~]

- **$Func_i$**: support verb that takes the keyword as its grammatical subject:

  $Func_1$(*order*) = *stems*, *comes* [*from* N]
  $Func_2$(*suspicion*) = *falls* [*on* N]
  $Func_2$(*cost*) = *amounts* [*to* N]

LFs play an essential role in the process of sentence synthesis, especially with respect to lexical choice and, more particularly, with respect to the paraphrasing system. Since collocational meanings are expressed under the control of the keyword, their lexicalization takes place at the surface-syntactic level.[3] At the Deep-Syntactic level, collocational lexical units are represented by the LF-symbols. Thus, for collocations with support verbs of the **$Oper_i$**-type, it is this LF that represents the support verb at the Deep-Syntactic level and not the corresponding value, which only appears at the Surface-Syntactic level.

The subscript to the name of an LF indicates which actants of the keyword are realized as its syntactic dependents and how they are realized. Thus, for the noun DEFEAT, which has two actants, X (its 1st actant, 'the winner') and Y (its 2nd, 'the loser'), the subscript '$_1$' of the LF **$Oper_1$** indicates that the lexical realization of X fulfills the function of the grammatical Subject of the value, while in the case of **$Oper_2$** it is the lexical realization of Y which fulfills this function.

In MTT, three types of actants are distinguished (Mel'čuk, 2004a, 2004b):

1. a *Semantic actant* (= SemA) of a lexical unit L that has a predicative meaning 'L' corresponds to an argument of 'L'; SemAs are denoted by Arabic numbers;
2. a *Deep-Syntactic actant* (= DSyntA) of L corresponds to a SemA of L or to a Surface-Syntactic Actant of L; DSyntAs are denoted by Roman numbers;
3. a *Surface-Syntactic Actant* (= SSyntA) corresponds to the Subject, the Object or another complement of L (if L is a finite verb). SSyntAs are denoted by names of the corresponding syntactic-surface relations in a given language.

DSyntAs constitute a sort of interface between SemAs, which are determined only by semantic considerations, and SSyntAs, which are determined only by syntactic considerations (word order, distribution, agreement, etc.). For this reason, DSyntAs are necessarily more general than SSyntAs. Thus, a DSyntA **II** of a verb L stands for a family of syntactic constructions that include the Direct Object, the most important Oblique Object (if L does not have a Direct Object) and the Agentive complement, as well as other syntactic elements. The DSyntAs are determined, so to speak, from both sides, the semantic side and the surface-syntactic side. On the one hand, any expression that syntactically depends on L and manifests a SemA of L is a DSyntA of L. For instance, in *his fear*, the determiner HIS is a DSyntA of FEAR, since the meaning of the phrase is 'John←**1**-fear'. However, HIS is by no means a SSyntA of FEAR: it is its determiner. On the other hand, any expression that is a SSyntA of L is also its DSyntA, even if it does not correspond to one of L's SemAs. For example, the Beneficiary Dative in the case of a verb such as [*to*] SEW is considered to be its Indirect Object. Therefore, it is one of its DSyntAs as well. As a result, in a sentence such as *Maria sew her child a skirt*, SEW has three DSyntAs: *Maria* is **I**, *skirt* is **II**, and *child* is **III**. The DSyntA **III** is introduced to all "creation" verbs by a general syntactic rule of English (Mel'čuk, 2004a, 2004b).

Having introduced the basics on actants, we can now be more precise with respect to the subscript of LFs. The subscript of an LF points to the i-th DSynt-slot in the government pattern (GP) of the keyword that is to be realized as DSyntA **I** of this LF. Thus, the subscript '$_1$' of $\mathbf{Oper_1}(L)$ points to DSynt-slot **I** in the GP of L, the subscript '$_2$' of $\mathbf{Oper_2}(L)$ points to DSynt-slot **II** in the GP of L, etc. In LFs with several subscripts such as $\mathbf{Labor_{ij}}$, the order of subscripts refers to the syntactic role of the DSyntAs of the LF: the subscript '$_i$' of $\mathbf{Labor_{ij}}$ stands for its DSyntA **I** and '$_j$' for its DSyntA **II**. For instance, $\mathbf{Labor_{21}}$ stands for a verb whose DSyntA **I** corresponds to the DSynt-slot **II** of the keyword and whose DSyntA **II** corresponds to the DSynt-slot **I** of the keyword. The syntactic position of the keyword does not need to be specified since it is implicitly given by the definition of **Labor**: it is always its third DSyntA.

The name of the LF also encodes some aspects of the correspondence between SemAs and DSyntAs of SVCs, i.e., a part of its diathesis. Thus, the name of $\mathbf{Func_i}$ signals that its keyword is its DSyntA **I**, while the name of $\mathbf{Oper_i}$ signals that its keyword is its DSyntA **II**. Therefore, in the traditional notation, the information about the diathesis of a support verb is distributed between the name of the LF and its actantial subscripts. For instance, in an LF with only one subscript like $\mathbf{Func_i}$, the subscript does not refer to the DSyntA **I**—as the first subscript of $\mathbf{Labor_{ij}}$ or the only subscript of $\mathbf{Oper_i}$ do—but rather to its DSyntA **II**. In order to facilitate the implementation, a more transparent notation along the lines proposed by Kahane and Polguère (2001) would be desirable. Thus, any support verb, including those represented by **Func**$_i$ and **Labor**$_{ij}$, could be represented in the following way (the subscript '$_C$' denotes the keyword):

$\mathbf{Vsup_{1C}}$(DEFEAT) = [*to*] *inflict* [ART ~ *on* N] ≡ $\mathbf{Oper_1}$
$\mathbf{Vsup_{2C}}$(DEFEAT) = [*to*] *suffer* [~ *at* N'*s hands*] ≡ $\mathbf{Oper_2}$
$\mathbf{Vsup_{C2}}$(SUSPICION) = *falls* [*on* N] ≡ $\mathbf{Func_2}$
$\mathbf{Vsup_{C1}}$(ORDER) = *stems* [*from* N] ≡ $\mathbf{Func_1}$
$\mathbf{Vsup_{12C}}$(LOAN) = [*to*] *give* [N *on* ~] ≡ $\mathbf{Labor_{12}}$
$\mathbf{Vsup_{32C}}$(LOAN) = [*to*] *get* [N *on* ~] ≡ $\mathbf{Labor_{32}}$

In what follows, we do not adopt this radical change, but later (Section 3.3) we make use of the subscript '$_C$' to distinguish between biactantial and triactantial support verbs.

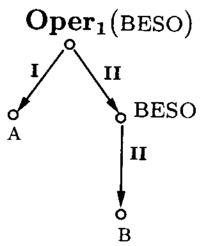
After this brief presentation of our framework, we can specify more clearly our object of study. We concentrate on the process of synthesis of SVCs of the **Oper**$_i$-type in Spanish.[4] More particularly, we pay special attention to the DSyntAs of support verbs by examining their correspondences at the semantic and surface-syntactic levels. As we will show later, the construction of the DSyntR of a SVC involves two types of rules:

1. transition (or correspondence) rules between:
   - the semantic representation (SemR) and the deep-syntactic representation (DSyntR)
   - the DSyntR and the surface-syntactic representation (SSyntR)
2. equivalence rules between synonymous DSyntRs, which can be divided into two further subtypes:
   - lexical paraphrasing rules
   - syntactic paraphrasing rules

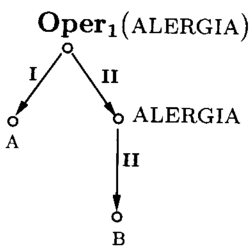
1.2 *The Problem*

In this paper we address two related problems: (i) how to represent the particular diathesis of a support verb; and (ii) how to determine the number of DSyntAs of **Oper**$_i$. As for the first problem, we would like to indicate the "irregular" correspondence between the SemAs of the supported noun and the DSyntA of the support verb. Since in a SVC, the predicate is expressed by the noun, from the semantic viewpoint, the actants are linked to the noun only. However, at the DSynt-level and the SSynt-level, the actants can be linked either to the verb or to the noun. That is, the SemAs of the noun can be realized as SyntAs of the verb or the noun, respectively.

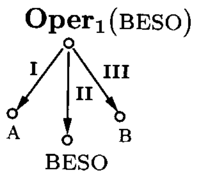
As far as the second problem is concerned, we must draw the reader's attention to the fact that support verbs do not have a fixed number of actants at the DSynt-level. By definition, **Oper**$_{\mathbf{i}}$ has at least two DSyntAs: DSyntA **I** that is the future grammatical Subject of the support verb and DSyntA **II** that will be its first complement. This is the only possible distribution when a monoactantial noun is involved. However, there are many polyactantial nouns that select support verbs. Then, the question to be answered is how the other actants of the supported noun are to be distributed? Or, in other words, what node governs, from the deep-syntactic point of view, the lexical units that function as noun's SemAs different from the SemA **i**: **Oper**$_{\mathbf{i}}$ or its keyword? Thus, the point to decide is whether some SVCs such as *dar un beso a Luis* '[to] give a kiss to Luis' or *sufrir de alergia al polen* '[to] suffer allergy to pollen' should be represented by identical deep-syntactic trees (see (1) and (2) below), so that the fact that the first verb has three SSyntAs, while the second one has only two is manifested only at the surface by different surface-syntactic trees, or whether these actantial differences should be made visible already at the DSynt-level. In the last case (see (3) and (4) below), we would have a tree whose top node is **Oper**$_{\mathbf{1}}$ with three DSyntAs for the construction with DAR and a tree whose top node is **Oper**$_{\mathbf{1}}$ with two DSyntAs for the construction with SUFRIR (the second SemA of ALERGIA being realized as its syntactic dependent).

(1) 

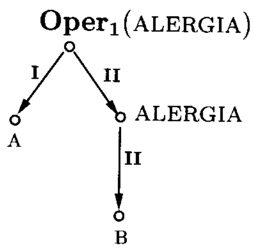
María dio un beso a Luis
'Mary gave a kiss to Luis'.

(2) 

María sufre de alergia al polen
'Mary suffers from allergy to pollen'.

(3) 

María dio un beso a Luis

(4) 

María sufre de alergia al polen

An important fact to be taken into account is that SVCs are dealt with in the paraphrasing system of the *Meaning-Text Model* (MTM). The rules of the paraphrasing system replace, under the control of LFs, lexical nodes in a given DSynt-structure (*DSyntS*) producing another *DSyntS*. For instance, the semantic equivalence between *John welcomed us warmly* and *John gave us a warm welcome* is captured by a rule that maps a full verb [V] onto the corresponding SVC formed by a support verb [$\mathbf{Oper_1}$] and the derived noun of the full verb [$\mathbf{S_0}$(V)].[5] In order to describe this equivalence, we have to adjust the syntactic structure because we pass from a DSyntS with two DSyntAs to a DSyntS with three DSyntAs. However, as we will see later, paraphrasing rules known from the MTT-literature (see, among others, Mel'čuk, 1992) are not sufficiently explicit for this task.

1.3 *The Goal of the Paper*

In the following sections, we examine the possibilities of the distribution of DSyntAs in the SVCs and attempt to find the best description of their DSyntSs. "Best" means a description

- that describes the syntactic relations between the elements participating in the SVC in a most intuitive way,
- that is least expensive in terms of the number of rules to apply,
- that is easiest to implement.

We begin by examining the particularities of the diathesis of support verbs (Section 2). Then, we look at three alternative strategies for the distribution of DSyntAs in a SVC: (i) to leave open the number of DSyntAs of $\mathbf{Oper_i}$ and to entrust the possible adjustments to the syntactic paraphrasing rules (Section 3.1), (ii) to restrict the number of DSyntAs of $\mathbf{Oper_i}$ to two DSyntAs and to entrust the possible adjustments to the correspondence rules between DSynt-level and SSynt-level (Section 3.2), (iii) to determine the DSyntS according to the value of the LF distinguishing between different $\mathbf{Oper_i}$s depending on the number of their DSyntAs (Section 3.3). In Section 4, we first concentrate on transition rules between the SemR and the DSyntR, and present then in Section 5 paraphrasing rules which describe the equivalence between synonymous DSyntRs. In Section 6, finally, we outline the procedure of synthesis for a SVC.

2 The Diathesis of the Support Verb

Before treating the distribution of DSyntAs in SVCs, we would like to emphasize that the SVCs present problems for any syntactic theory. It is necessary to devise a special mechanism to account for the fact that a verb which is treated as empty can have actants. As a general rule, the syntactic actants of a lexeme are in correspondence with its semantic actants. However, since the support verb is empty and cannot have any semantic actant, its syntactic actants must be in correspondence with the semantic actants of the predicative noun. Several authors (among others, Grimshaw and Mester 1988, Rosen 1989, Di Sciullo and Rosen 1991) point out the peculiarity of support (or *light*) verbs: a support verb possesses a skeletal argument structure that contains only empty semantically unspecified positions. In order to show how MTT would treat the "filling" of argumental positions, we have to consider first the diathesis of support verbs.[6]

With respect to the SVC, the following observations call for an explanation:

1. an empty lexeme has syntactic actants,
2. one of the syntactic actants, the noun, is a full predicative lexeme, and
3. the semantic actants of the full predicative lexeme can be realized as syntactic actants of the verb or as syntactic actants of the noun itself.

But before, we must briefly present how the diathesis of a predicative lexeme is represented.

Each lexeme whose meaning is a predicate has a *basic diathesis.* Thus, for example, when we establish the correspondence between the four SemAs of the verb [*to*] PUNISH ('X punishes Y by Z for W') and its four DSyntAs, we are assigning the verb its basic, or lexicographic, diathesis. In the format of the *Explanatory Combinatorial Dictionary* (ECD), this information is given in the *Government Pattern* (GP) of the lexeme in question. If we represent the diathesis by a matrix of two lines where the first line is occupied by the semantic valency and the second line by the deep-syntactic valency, the basic diathesis of [*to*] PUNISH looks as shown in (5c):

(5) a. *X punishes Y with Z for W*

b. *The teacher* [I] *punishes the boy* [Y] *with the repetition of the exercise* [Z] *for arriving late* [W] c.

X	Y	Z	W
I	II	III	IV

The basic diathesis of a verb can be modified by grammatical voice (see Mel'čuk, 1993, 1994, 1997a, 1997b).

In order to establish the diathesis of a support verb, we must treat a SVC as a complex predicate where the nominal constituent provides the semantic valency and the verbal constituent the syntactic valency.[7] It is legitimate to talk about complex predicates here because the verb furnishes the syntactic structure and the noun fills, together with its SemA(s), the syntactic positions of the verb. It is not possible to describe the diathesis of a support verb separately from the noun because the syntactic valency of the verb is closely related to the semantic valency of the noun. Rather, we must draw upon a mechanism that establishes the correspondence between the semantic valency of the noun and the syntactic valency of the support verb.

The **Oper**$_i$-support verb possesses at least two syntactic actants. Therefore, the minimal diathesis of **Oper**$_i$ is defined as follows:

(6) Minimal Diathesis of **Oper**$_i$:

$X(P)$	P
I	II

The keyword predicate (i.e. the nominal predicate) of **Oper**$_i$ is referred to as 'P' and the first SemA of P as '$X(P)$'. The first SemA '$X(P)$' of P is mapped onto the first DSyntA of the support verb and P itself onto the second DSyntA of the verb, i.e., onto the noun that expresses P. The particularity of this diathesis is that the predicate itself corresponds to a DSyntA of the verb and that an argument of this predicate corresponds to a DSyntA depending on a node different from that of the predicate, i.e. on the verb. One could say that a support verb is an adapter of the diathesis of a nominal predicate: a syntactic device that allows the speaker to express the SemAs of a noun by means of syntactic relations characteristic of verbs. Support verbs could be related, in a certain sense, to inflectional categories such as voice or the transitivation. See in this respect (Daladier, 1996): a support verb adapts the diathesis of a noun to that of a verb, so that $X(P)$ (in the case of *inflicting a punishment*) or $Y(P)$ (in the case of *receiving a punishment*) becomes the grammatical Subject of a verb: X *inflicts a punishment* or Y *receives a punishment*. Therefore, the rule of formation of complex predicates can be formulated as follows:

Formation of a Complex Predicate with a Support Verb

In order to express a predicative noun P as the dominant predicate of a sentence, we need a support verb that promotes the syntactic range of its actants on the scale of functional hierarchy:

- the semantic actant i of P is expressed as the deep-syntactic actant I of the support verb (future syntactic Subject);
- P becomes the deep-syntactic actant II of the support verb (its future First Complement).

Until now, we have considered the LF $\mathbf{Oper_i}$ with only two DSyntAs. This is the only possibility in the case of a monoactantial noun. But for polyactantial nouns, the distribution of actants between $\mathbf{Oper_i}$ and the noun is not straightforward: sometimes there is the possibility of attaching the actants either to the verb or to the noun. The usual practice within MTT has been to allow the LF $\mathbf{Oper_i}$ to govern all lexical units filling the DSynt-slots of the noun. Thus, Mel'čuk (1996:61) states that "further DSyntAs of $\mathbf{Oper_i}$, if any, are the phrases described in the government pattern of L [the keyword] as further DSyntAs of L." In fact, it is not possible to foresee what the DSyntAs of $\mathbf{Oper_i}$ of a given noun are because the $\mathbf{Oper_i}$-verb does not have its own basic diathesis. As we already indicated, an LF $\mathbf{Oper_i}$-SVC with any keyword has a minimal diathesis. However, one cannot determine the maximal diathesis of an $\mathbf{Oper_i}$-SVC in connection with a given noun without knowing what the elements of the $\mathbf{Oper_i}$-value at the surface will be. Thus, for instance, in the synthesis of the DSyntS of $\mathbf{Oper_1}$(ALERGIA 'allergy'), it is impossible to say how many DSyntAs will depend on $\mathbf{Oper_1}$ without taking into account what $\mathbf{Oper_1}$(ALERGIA) will be on the surface and without verifying what the syntactic valency of the value is. Given that ALERGIA is a noun with two SemAs, in principle the DSyntS of the $\mathbf{Oper_1}$(ALERGIA)-SVC can be a tree with three DSyntAs or a tree with two DSyntAs (see Figures 8a and 8b below). In this sense, a support verb in a given SVC has a specific diathesis, but $\mathbf{Oper_i}$, as a generalized lexeme, does not.

Therefore, the diathesis of $\mathbf{Oper_i}$ is variable in two ways:

(i) The number of DSyntAs of $\mathbf{Oper_i}$ depends on the number of SemAs of its keyword.

(ii) During the synthesis of the DSyntS of a SVC, nothing controls the distribution of the DSyntAs between the LF and the keyword. Thus, in the case of a keyword C_0 with four SemAs, the possible DSynts of $\mathbf{Oper_i}$ would be: (1) $\mathbf{Oper_i}$ with five DSyntAs; (2) $\mathbf{Oper_i}$ with four DSyntAs (one of the SemAs of the keyword being realized as its syntactic dependent); (3) $\mathbf{Oper_i}$ with 3 DSyntAs (two of the SemAs of the keyword being realized as its syntactic dependents); (4) $\mathbf{Oper_i}$ with two DSyntAs (three of the SemAs of the keyword being realized as its syntactic dependents). This correlation can be represented as shown in Figure 1, p.107.

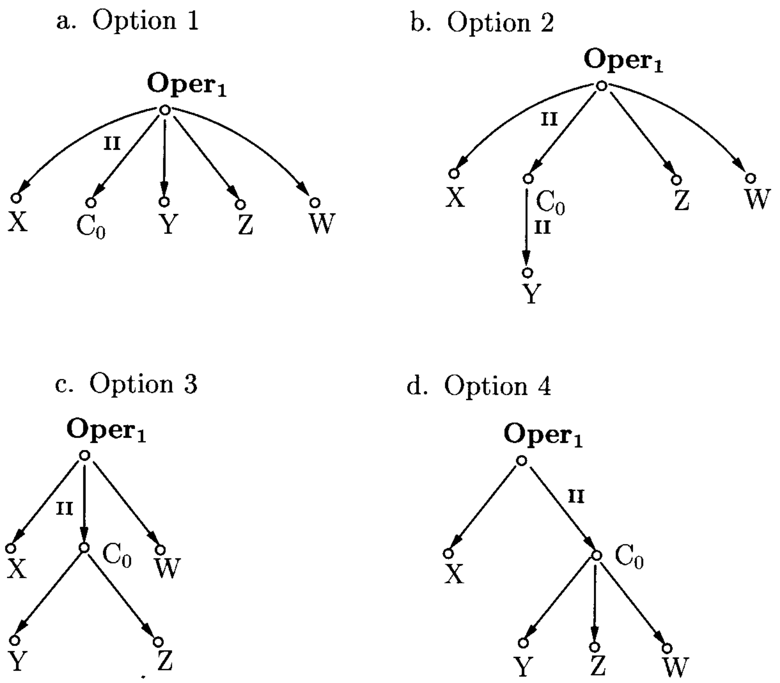


Figure 1: Options for the distribution of DSyntAs of $\mathbf{Oper_1}(C_0)$ in the case of a C_0 with four SemAs

Since each keyword of the LF $\mathbf{Oper_i}$ has a different number of SemAs, the number of DSyntAs of $\mathbf{Oper_i}$ is uncertain. The diathesis of $\mathbf{Oper_i}$, whose keyword has more than one SemA, is, by definition, variable; cf.:

(7) Variable Diathesis of $\mathbf{Oper_i}$

X(P)	P	...	...	...	...
I	II	...	...	...	...

The empty cells of the table represent the potential SemAs and DSyntAs. From the semantic side, the number of SemAs depends on the noun *N* that expresses the predicate *P*. From the syntactic side, the number of DSyntAs depends on both the noun and the support verb in question. For example, the DSyntS of the SVC *sufrir de alergia* [*al polen*] should not be the same as the DSyntS of *tener alergia* [*al polen*], even though the two SVCs are represented by the same LF—$\mathbf{Oper_1}$(ALERGIA). However, it is not possible to determine the diathesis of $\mathbf{Oper_1}$(ALERGIA) without looking at the diathesis of its respective value. As we will propose later, it is only possible to establish the diathesis of a given SVC as a whole. Although a SVC is not a lexical unit, it has a diathesis. This is due to the fact that while a SVC expresses a single predicate, the arguments of this predicate are not only in correspondence with the DSyntAs of the predicative lexical unit, but also with the DSyntAs of the support verb.

Let us now examine the possibilities of the distribution of DSyntAs considering the variable diathesis characteristic of supports verbs.

3 Number of DSyntAs of the LF $\mathbf{Oper_i}$

Drawing on the synthesis of a sentence, we present in this section three approaches to the construction of the DSyntS of a SVC:

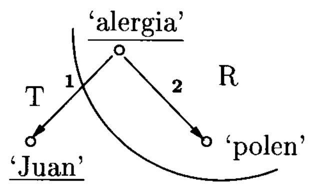
(i) to allow the diathesis of $\mathbf{Oper_i}$ to be variable
(ii) to limit the diathesis of $\mathbf{Oper_i}$ to merely two DSyntAs
(iii) to determine the diathesis of $\mathbf{Oper_i}$ according to the elements of its surface value

We start with the traditional approach in MTT, namely (i).

3.1 *Variable Diathesis of LF $Oper_i$*

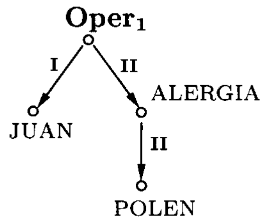
Suppose we want to generate the surface string *Juan tiene<sufre de> alergia al polen* lit. 'Juan has <suffers of> allergy to the pollen' starting from the following SemR:[8]

(8)

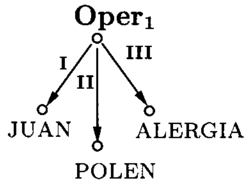


We have the possibility of constructing two DSyntSs (with $\mathbf{Oper_1}$ as top node):

(8) a. **Oper$_1$** with two DSyntAs:



b. **Oper$_1$** with three DSyntAs:



For the passage to the surface-syntactic structure (SSyntS), it is necessary to compute the value of the LF. In the lexicographic entry for ALERGIA, which has the propositional form 'alergia de X a Y', we find, among other LF-information:

Oper$_1$: *tener* [$\sim$ a N = Y]; *sufrir* [*de* $\sim$ $\rightarrow$ *a* N = Y][9]

The verb TENER is triactantial, while SUFRIR is biactantial. As stated above, it is not possible to determine the diathesis of **Oper$_1$**(ALERGIA) without taking into account the diathesis of the verb that functions as its value. But since the correspondence between SemAs and DSyntAs of a SVC is rather particular (its SemAs depend on the noun, while its DSyntAs depend on the verb or on the noun), we need to establish the diathesis for the SVC as a whole. Thus, the diathesis of *tener alergia a Y* can be represented by the following table:

(9) Diathesis of the SVC *tener alergia a Y*

X('alergia')	'alergia'	Y('alergia')
I (V_{sup})	**II** (V_{sup})	**III** (V_{sup})

The diathesis of *sufrir de alergia a Y* is different; cf.:

(10) Diathesis of the SVC *sufrir de alergia a Y*

X('alergia')	'alergia'	Y('alergia')
I (V_{sup})	**II** (V_{sup})	**II** (N)

(9) expresses both SemAs of 'alegria' as DSyntAs of V_{sup}, i.e. of TENER. (10) indicates that V_{sup}, i.e. SUFRIR, has only two DSyntAs. The second SemA of 'alegria' (e.g., POLEN) is expressed as a DSyntA of the noun ALEGRIA.

If we choose the DSynS in (8a), the value of **Oper$_1$** that suggests itself is SUFRIR; if we choose the DSynS in (8b), it is only TENER, which can serve as the top node of the corresponding SSyntS. Nonetheless, the two

DSyntSs can be modified by syntactic paraphrasing rules such that each of them can facilitate the expression of any of the two values in the SSyntS (see Section 5).

Although $\mathbf{Oper_i}$ is considered to be a lexeme of the language, its nature of a deep, or generalized, lexeme prevents it—contrary to all ordinary lexemes—from having a stable number of actants. It is true that many lexemes have more than one government pattern, i.e., modifications of the diathesis that are not expressed syntactically (but rather by morphological means or by a structural word), as for instance, [*to*] PROPOSE. [*To*] PROPOSE possesses the propositional form *X proposes Y to Z.* Its three SemAs can be expressed as three DSyntAs (and SSyntAs): *John* [**I**] *proposes to us* [**III**] *that Mary be the guide* [**II**] or as four DSyntAs (and SSyntAs): *John* [**I**] *proposes us* [**IV**] *Mary* [**II**] *as guide* [**III**]. That is, in contrast to the case of $\mathbf{Oper_i}$, both government patterns of [*to*] PROPOSE can be semantically and syntactically tested: we can verify which are the means of the expression of its SemAs in order to verify the dependencies.

In the case of $\mathbf{Oper_i}$, we do not know the means of the expression of the SemAs of the keyword until we see at the surface what the value of the LF is. We can only verify the means of the expression of the SemAs of the keyword in the context of a given support verb. The government patterns for $\mathbf{Oper_i}$ cannot be restricted.

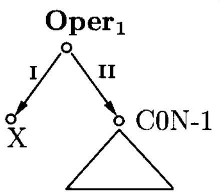
As a conclusion, we can state that LF $\mathbf{Oper_1}$ has a variable number of DSyntAs, but its value has a specific number of DSyntAs. Therefore, in the traditional approach of MTT, it is possible to specify what the diathesis of a support verb is only from the analysis point of view.

3.2 *Only Two DSyntAs for All $Oper_i$*

Since the previous problems are due to the fact that $\mathbf{Oper_i}$ has a variable diathesis, we could give it a status closer to that of other lexemes and to assign it a fixed number of DSyntAs. By definition, $\mathbf{Oper_i}$ has at least two DSyntAs. This number could be declared to be the maximum number of DSyntAs for all SVCs represented by $\mathbf{Oper_i}$. Other SemAs of the noun will be realized at the deep-syntactic level as dependents of the noun, rather than as dependents of the support verb.

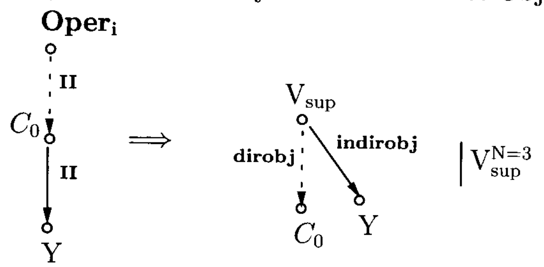
This approach implies a semantization of the DSyntSs. Usually, nodes of a DSyntS must be semantically motivated. That is, they are chosen according to their meaning (Mel'čuk, 1997a:94). However, a node such as $\mathbf{Oper_i}$ is not selected due to its meaning, but rather due to syntactic needs of the language: if the entry node of a given SemR cannot be expressed verbally, another verb is needed as the syntactic top node of the sentence under construction (see Endnote 6). This constraint is undisputed in languages like Spanish, English and French. In spite of the syntactization of the DSyntSs achieved by the introduction of $\mathbf{Oper_i}$, one can try to keep the semantic motivation of actantial relations. Thus, in our sentence, the actantial relation between ALERGIA and POLEN is semantically motivated. However, the relation between between $\mathbf{Oper_1}$ and POLEN is not. The fact that one of the values of $\mathbf{Oper_i}$(ALERGIA), TENER, makes the expression of POLEN as its third SSyntA possible is an issue that rises from the surface syntax, not from the meaning. Therefore, according to this approach, we attempt to keep the actantial relations between governors and dependents as specified in the corresponding SemS.[10] If a predicate 'P' with several arguments is realized in the DSyntS by $\mathbf{Oper_1}$-**II**→L('P'), only one of its SemAs will be realized as a DSyntA of $\mathbf{Oper_i}$; all other SemAs will be realized as DSyntAs of L, i.e., the lexicalization of P. As a result, the triactantial TENER and the biactantial SUFRIR are not distinguished at the deep-syntactic level. Both function as $\mathbf{Oper_1}$, that is, as a support verb necessary to verbalize the meaning of the noun. At this level, we know of this verb only that it posssesses the keyword as its DSyntA **II** and the first SemA of the keyword as its DSyntA **I**. Thus, the deep-syntactic level remains vague and generalized with respect to problems of government. We assign a unique diathesis to all $\mathbf{Oper_i}$s. Therefore, any keyword C_0 with more than one SemA will have an $\mathbf{Oper_i}$ with two DSyntAs only: DSyntA **I** will be the first SemA of the keyword C_0, DSyntA **II** will be the keyword C_0 with all its DSyntAs except the one realized as DSyntA **I** of $\mathbf{Oper_i}$. In a graphical representation, we represent DSyntA **II** as a subtree whose top node is the keyword:

(11) DSyntR for all $\mathbf{Oper_i}$



Let us compare this figure with Figure 1, where we presented all possible syntactic options for a C_0 with four SemAs. Among the two possible DSyntSs mentioned above for $\mathbf{Oper_1}$(ALERGIA), one would always choose the LF with two DSyntAs (see 8a). The rules of correspondence between the DSyntS and the SSyntS will have to take care of the transfer of all remaining actants attached to C_0 as actants of the support verb, on condition that the support verb has the corresponding slots in its government pattern. In order to pass to the surface-syntactic level, a rule of the type as shown in (12) would be applied.

(12) DSyntR II of the keyword $\Longrightarrow$ Indirect Object of the support verb



In this figure, the actantial relation, which is being transferred together with its terminal node, and the resulting actantial relation at the surface-syntactic level are drawn as a solid arc. The context relations are drawn as dashed arcs. The node Y becomes a SSyntA of the support verb on condition that this verb is triactantial (denoted by $V_{sup}^{N=3}$). Thus, this rule would be applied in the case where **Oper**$_1$ is lexicalized by the triactantial TENER, but not in the case where we opt for the biactantial SUFRIR.

Let us examine the advantages and the inconveniences of this solution from a merely linguistic point of view. With the assignment of a unique diathesis to **Oper**$_i$, we give it a status similar to other lexemes. However, at the same time, we falsify the DSyntS by representing at a syntactic level (even though it is the deep-syntactic level, it is above all syntactic) triactantial constructions such as *dar un beso a Luis* '[to] give a kiss to Luis' by a deep-syntactic tree with two actantial branches. The semantization mentioned before hides the syntactic reality. The main objective of a syntactic structure must be to reflect how lexemes are linked syntactically. However, by specifying the lexicalizations of all SemAs of a signified as the syntactic dependents of the corresponding signifier L, we do not show the real syntactic structure. If we assume, as done in MTT, that the DSyntS is more semantic than the SSyntS, this is due to two facts: (i) the nodes of DSyntS must be selected for their meaning, and (ii) the syntactic relations contained in a DSyntS are generalized relations. But, nonetheless, governors and dependents of DSynt-relations must be chosen according to some syntactic criteria.

In the following subsection, we present the third approach in relation to the distribution of the DSyntAs in a SVC.

3.3 *Number of DSyntAs of* $Oper_i$ *Depending on its Value*

The vagueness as produced by the variable diathesis of $\mathbf{Oper_i}$ is avoided if the diathesis of the lexeme that functions as value of $\mathbf{Oper_i}$, i.e. the support verb, is taken into account. The corresponding information can be specified either in the entry for the keyword, or in the entry for the value itself. In the entry for the keyword, different $\mathbf{Oper_i}$s could be distinguished according to their diathesis.[11] Thus, in the entry for ALERGIA, we would find:

$\mathbf{Oper_{1C}}$: *sufrir* [*de* ~]
$\mathbf{Oper_{1C2}}$: *tener* [~ a N = Y]

As a result, the value of the LF which is triactantial on the surface would be encoded in a different way than a value which is biactantial on the surface. The notation '$\mathbf{Oper_{1C2}}$' indicates that both DSynt-slots **I** and **II** of the keyword are filled by phrases that syntactically depend on **Oper**. If TENER is chosen, an $\mathbf{Oper_i}$ with three DSyntAs must be used, while if SUFRIR is chosen, a biactantial $\mathbf{Oper_i}$ must be used. In a certain sense, $\mathbf{Oper_{1C}}$ and $\mathbf{Oper_{1C2}}$ overspecify the denotation of the LF in question: both refer to the same LF, namely to $\mathbf{Oper_1}$ – unlike, e.g., $\mathbf{Oper_{2C}}$. $\mathbf{Oper_{2C}}$ denotes a support verb whose DSyntA **I** corresponds to the DSynt-slot **II** of the keyword. Therefore, it encodes a different information: among other things, $\mathbf{Oper_{2C}}$ is linked to a different semantico-communicative structure. Nonetheless, as already pointed out above, $\mathbf{Oper_{1C}}$ and $\mathbf{Oper_{1C2}}$ are not entirely redundant.

The notation '$\mathbf{Oper_{1C2}}$' is in agreement with the conventions of the MTT. As mentioned in the Introduction, in LFs with several subscripts like $\mathbf{Conv_{ijk}}$, the order of subscripts refers to the syntactic role of the DSyntAs of the LF. For instance, the notation $\mathbf{Conv_{3214}}$([*to*] BUY) = [*to*] *sell* indicates that the DSyntA **III** of [*to*] BUY corresponds to the DSyntA **I** of [*to*] SELL; the DSyntA **I** of [*to*] BUY is realized as the DSyntA **III** of [*to*] SELL, and so on (*X buys Y from Z with W* ≈ *Z sells Y to X for W*). So, for $\mathbf{Conv_{ijk}}$, the subscript '$_i$' refers to the DSyntA **I**, '$_j$' to the DSyntA **II** and '$_k$' to the DSyntA **III**. The subscripts of $\mathbf{Oper_{1C2}}$ are interpreted along the same lines: '$_1$' stands for its DSyntA **I**, '$_C$' for its DSyntA **II** and '$_2$' for its DSyntA **III**. The alternative encoding of a triactantial support verb as $\mathbf{Oper_{12}}$ would raise the problem of interpreting the second subscript as the DSyntA **II** of the LF rather than as the DSyntA **III**. In what follows, we use the notation '$\mathbf{Oper_{1C2}}$' to refer to a triactantial support verb, and '$\mathbf{Oper_{1C}}$' to refer to a biactantial verb.

This convention allows for a more deterministic synthesis of a SVC and makes obsolete adjustment rules that repair dead ends (see Subsection 6.1).

If the synthesis of a SVC is made taking into account the specific diathesis of **Oper**$_i$ traditionally determined at the SSynt-level, a correlation holds not only between the SemAs of the keyword C_0 and the possible number of DSyntAs of **Oper**$_i$ (as in the traditional MTT-approach), but also between the SSyntAs of a given value of **Oper**$_i$ and **Oper**$_i$ itself.

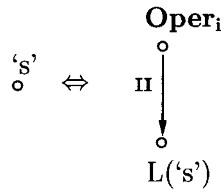
In the next section, we formulate the transition rules concerning SVCs from the semantic level to the DSynt-level, while keeping in mind the three different approaches.

4 Transition Rules Concerning SVCs

We begin with the discussion of the transition rules between the SemR of a SVC and its DSyntS, and examine then the transition between the DSyntS and the SSyntS.

Kahane and Mel'čuk (1999) distinguish between nodal rules and edge rules. According to them, nodal rules are also lexical rules. In other words, at the left-hand side of a nodal rule, a semantic node is specified, and at the right-hand side a lexical node. However, for SVCs, simple nodal rules are not appropriate. Rather, we must formulate *fission nodal* rules. A fission nodal rule maps a semantic node onto two lexical nodes which are linked by a deep-syntactic relation. For **Oper**$_i$-SVCs, a fission nodal rule establishes the correspondence between a semantic node labeled by a meaning '*s*' and a DSynt-subtree where the governor is the LF **Oper**$_i$ and the dependent is the nominal lexeme L with the meaning '*s*'.

(13) Nodal fission rule



1) 's' has no verbal realization
2) 's' is not the Theme

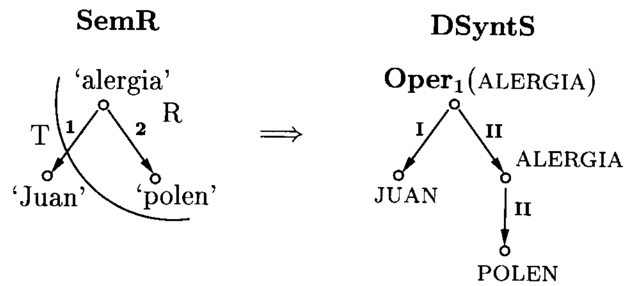
Since this rule is independent of the realization of the SemAs of '*s*', it is applicable in all three of the above approaches. However, its conditions deserve some comments.

In the traditional MTT-approach, if the meaning '*s*' has a possible verbal realization at the DSynt-level, the corresponding verb is chosen as the syntactic governor. It is only via a paraphrasing rule that a DSyntS with a full verbal node as syntactic governor can be replaced by the equivalent SVC—as, e.g., ACOGER 'to welcome' by *dar una acogida* '[to] give a welcome'. Therefore, the first condition is needed to make explicit that $\mathbf{Oper}_i$ is introduced as syntactic governor only if the meaning 's' does not have a verbal realization.[12] However, it is also possible to foresee a direct transition of '*s*' to an $\mathbf{Oper}_i$-construction, i.e,. of 'acoger' to the SVC $\mathbf{Oper}_1$(ACOGIDA)—without passing by the directly associated verb $\mathbf{V}_0$(ACOGIDA). In this case, the first condition does not hold. It seems advisable to allow for both ways of producing a valid DSyntS: one with a full verb and another with a SVC. Using paraphrasing rules, it is then possible to derive the full verb from the SVC as well as the SVC from the full verb.

The second condition, '*s*' ≠ Theme, controls the choice between $\mathbf{Oper}_i$ and $\mathbf{Func}_i$. The choice of '*s*' as entry node depends on its thematic position: in the standard case, '*s*' is the communicatively dominant node of the Rheme. If '*s*' makes part of the Theme and has no verbal realization, the syntactic governor should be **Func**. Let us examine the following example:

(14) Transition SemR ⇒ DSyntS for
$\mathbf{Oper}_1$(ALERGIA) and $\mathbf{Func}_2$(ALERGIA)

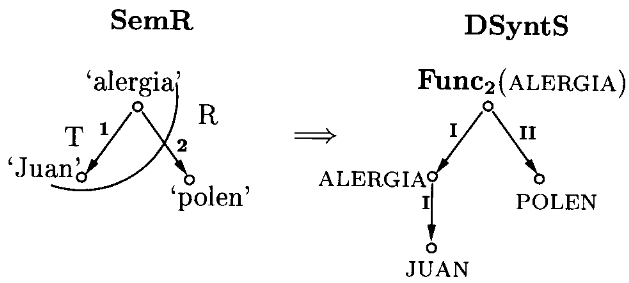
a.



Example:

Juan sufre de alergia al polen
'Juan suffers from allergy to pollen'.

b.



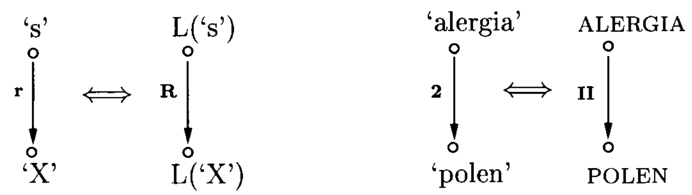
Example:

La alergia de Juan procede del polen
'The allergy of Juan comes from polen'.

Depending on the deep-syntactic realization of the semantic relations, we distinguish two types of arborization rules: (i) standard rules and (ii) mismatch rules.

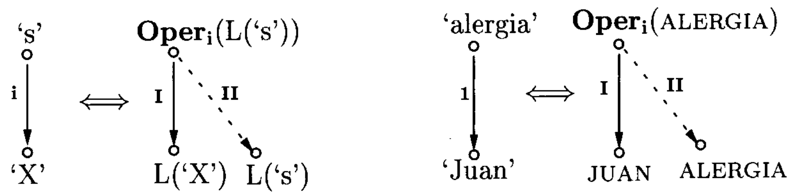
Standard Rules. In a *standard arborization rule*, the governor and the dependent linked by a DSynt-relation R express the meaning of the governor, respectively the dependent linked by a Sem-relation. The instance of R corresponds directly to the instance of r: for r = **1**, R = **I**; for r = **2**, R = **II**; etc. See the following figure (the nodes are considered to be part of the context in all rules of arborization):

(15) a. Standard arborizarization rule R_1 b. Instantiation of R_1



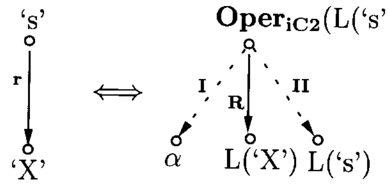
Mismatch Rules. In a *mismatch arborization rule*, only the dependent DSynt-node expresses the meaning of the dependent Sem-node. The syntactic governor does not correspond to the semantic governor. This rule is applicable to actantial relations of all governors produced by the fission nodal rules. Different conditions must be established for the different actantial relations. For each relation, we formulate a separate arborization rule. The following rule concerns the DSyntA I of the LF. It establishes the correspondence between the SemA **i** and the DSyntA I for all $\mathbf{Oper_i}$ ($\mathbf{Oper_{1/2/3/4}}$):

(16) a. Mismatch arborization rule R_2 b. Instantiation of R_2



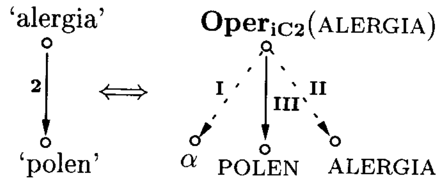
For the other DSyntAs of $\mathbf{Oper_i}$, approach-specific rules are defined. Thus, the following rule handles DSyntAs in accordance with the third apparoach (as in (12), dashed arcs indicate the required context of the resulting actantial relations):

(17) a. Mismatch arborization rule $R_{2.2}$



1) $\mathbf{r} \neq \mathbf{1}$; $\mathbf{r} \neq \mathbf{i}$
2) $\mathbf{R} \equiv \mathbf{r}+1$
3) $\mathbf{Oper_i}$ is not biactantial

b. Instantiation of $R_{2.2}$



Obviously, this rule does not apply in the case $\mathbf{Oper_i}$ with two DSyntAs.

If the diatheses of $\mathbf{Oper_{1C2}}$ and $\mathbf{Oper_{1C}}$ are to be distinguished, it is necessary to specify how the SemAs of the keyword are mapped onto the DSyntAs of **Oper**. Thus, for other actantial branches leaving the $\mathbf{Oper_{1C2}}$-node, the above rule shows the mismatch between the Sem-relation and the DSynt-relation: if $\mathbf{r} = \mathbf{2}$, then $\mathbf{R} = \mathbf{III}$, if $\mathbf{r} = \mathbf{3}$, then $\mathbf{R} = \mathbf{IV}$, and so on.

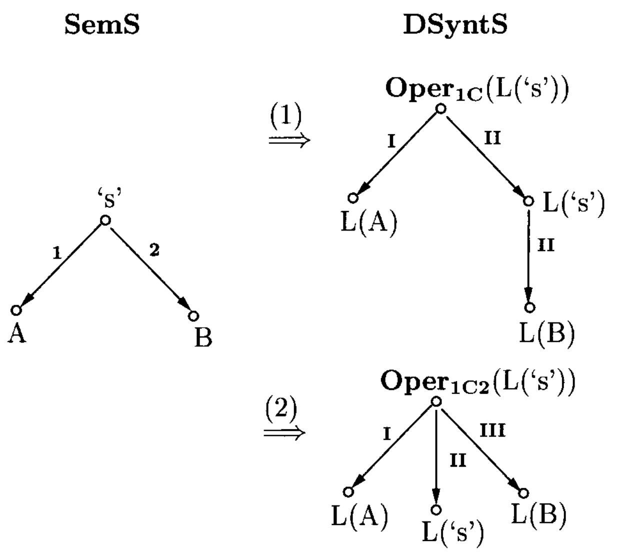
The rules in (16) and (17) have an identical left-hand side. However, the right-hand side context in (17) ensures that this rule is not applied to 'alergia'-**2**→'polen' as long as no **I**-relation has been introduced to produce $\mathbf{Oper_2}$-**I**→POLEN.

As far as $\mathbf{Oper_{1C}}$ is concerned, the rule applied to the second SemA of the keyword would be a standard arborization rule as we saw in (15).

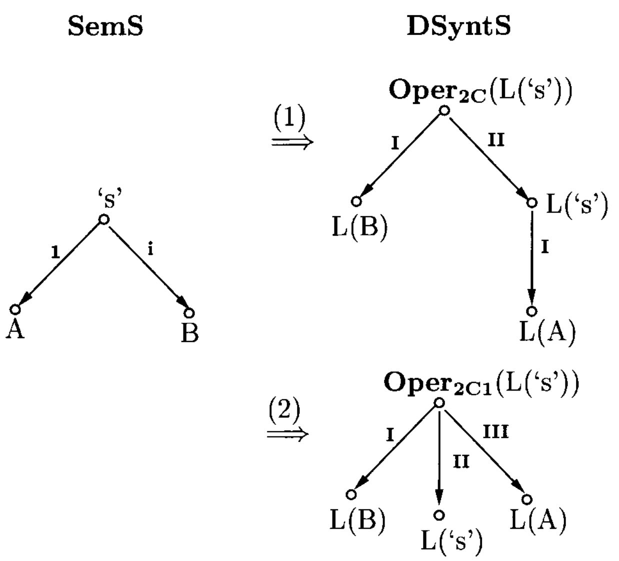
In the traditional approach of MTT, the second SemA of ALERGIA would be treated randomly either as the DSyntA **III** of $\mathbf{Oper_1}$ or the DSyntA **II** of ALERGIA. In the first case, the rule would be nearly identical to that shown in (17), with the difference that there would be no condition on the syntactic governor: any **Oper** can, in principle, have a DSyntA **III**.

Now we can regroup the rules just elaborated to show the correspondence between complete semantic and deep-syntactic structures. From a single SemS, two different DSyntSs can be obtained if two **Oper**-LFs are available: the biactantial **Oper**-LF (referred to as '$\mathbf{Oper_{1C}}$') and the triactantial **Oper**-LF (referred to as '$\mathbf{Oper_{1C2}}$'). The next figure (18) represents the situation encoded in (8).

(18) Mapping a semantic configuration onto $\mathbf{Oper_{1C}}$ or $\mathbf{Oper_{1C2}}$



(19) Mapping a semantic configuration onto $\mathbf{Oper_{2C}}$ and $\mathbf{Oper_{2C1}}$



In (19), we represent the correspondence of a semantic configuration with an **Oper**$_i$ (**i** $\neq$ **1**). The only difference between (18) and (19) is the correlation between the Sem-branch **i** and the DSynt-branch I. Other rules are the same as for **Oper**$_{1C}$ and **Oper**$_{1C2}$. Cf. textual examples in Spanish for both of these subcases:

B está bajo el control de A 'B is under A's control'.

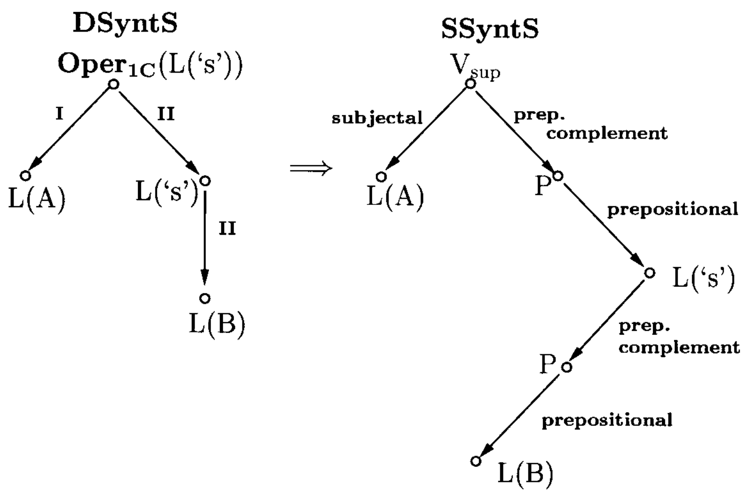
B recibe un golpe de A 'B receives a blow from A'.

If paraphrasing is conceived as the choice between different options during the transition from level n to level $n+1$, one can consider the transitions in the previous figure as paraphrasing. In this sense, several SemS⇒DSyntS transition rules with the same left-hand side (and different right-hand sides), constitute a paraphrasing rule (for paraphrasing transition rules, see Iordanskaja *et al.*, 1996:289-290).

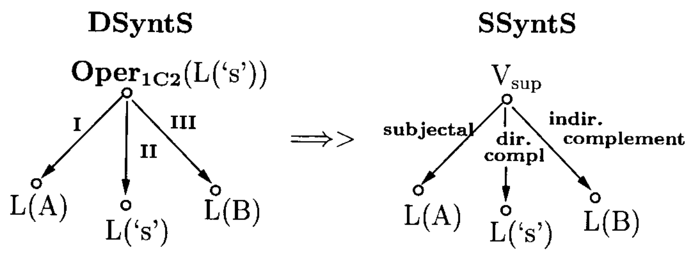
The passage from the DSynt-level to the SSynt-level will not be presented in detail here. In our work, this passage is realized in terms of standard DSynt-rules: computing of the value of the LF, mapping DSynt-relations onto SSynt-relations or onto SSynt-nodes by arborization rules, etc. All information for this passage is taken from the government pattern of the value of the LF. Thus, for example, if the LF-value is SUFRIR '[to] suffer', the preposition which introduces the keyword will be DE, while the preposition which introduces the prepositional complement of ALERGIA will be A. Consider, for illustration, the mapping of an **Oper**$_{1C}$-construction onto its SSynt-realization in (20a), and the mapping of an **Oper**$_{1C2}$-construction onto its SSynt-realization in (20b).

(20) Mapping **Oper**$_{1C}$ and **Oper**$_{1C2}$ onto a SSynt-Structure

a.



b.



5 Paraphrasing Rules Concerning SVCs

In what follows, we present the paraphrasing system as defined in MTT. This system derives from a basic DSyntS all possible synonymous DSyntSs via paraphrasing rules. The basic DSyntS is the "canonical", i.e., the most direct, deep-syntactic image of the corresponding SemS. Two types of rules are employed: lexical rules, which are formulated in terms of LFs, and syntactic rules, which perform the structural changes induced by the lexical rules.

5.1 *Lexical Paraphrasing Rules*

The lexical paraphrasing rules are language-universal. They encode semantic (quasi-)equivalences expressed in terms of LFs. Each rule encodes a possible substitution of a given DSynt-subtree by another DSynt-subtree *salva significatione*. That is, the source subtree and the destination subtree are by definition semantically (quasi-)equivalent. Lexical paraphrasing rules are thus conceived as lexical substitution operations, not as lexical choice operations.

Two types of lexical paraphrasing rules can be distinguished: (i) rules that replace an individual lexeme by an LF, i.e., rules of the type $C_0 \equiv \mathrm{LF}(C_0)$, and (ii) rules that replace a given LF by another LF of the same keyword, i.e., rules of the type $\mathrm{LF}_1(C_0) \equiv \mathrm{LF}_2(C_0)$. With respect to SVCs, the rules of the first type are always rules of fission because they replace a node by two nodes that are linked by an actantial branch. Rules of the second type apply to a fissioned node substituting a node by another node. The structural changes provoked by this substitution are handled by syntactic paraphrasing rules, which are presented in the following section.

For illustration of lexical paraphrasing rules, we cite one of the rules given in (Mel'čuk, 1992:37-43). As stated above, the symbol 'C_0' designates the starting lexeme, i.e., the keyword of the corresponding LF. The LF $\mathbf{S_0}$ is a nominalization of this lexeme.

Lexical paraphrasing rule:

$$C_0 \equiv \mathbf{Oper}_1(\mathbf{S}_0(C_0))\text{-}\mathbf{II}\rightarrow\mathbf{S}_0(C_0)$$

Cf. application of this rule:

(21) *El profesor castigó* $[C_0]$ *al niño* 'The teacher punished the boy'.

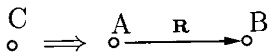
$\equiv$

El profesor puso $[\mathbf{Oper}_1(\mathbf{S}_0(C_0))]$ *un castigo* $[\mathbf{S}_0(C_0)]$ *al niño*
'The teacher imposed a punishment on the boy'.

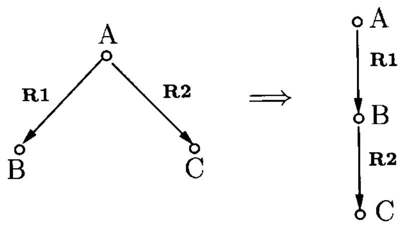
5.2 *Syntactic Paraphrasing Rules*

The syntactic paraphrasing rules realize the changes in the DSyntS induced by a lexical rule. Three changes are possible; see (Mel'čuk, 1992:45):

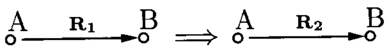
1. *Fission* or *Splitting* of a node in two:



2. *Transposition* of a DSynt-branch to another governor



3. *Renaming* of a DSynt-branch

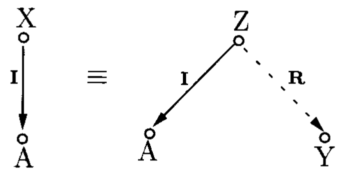
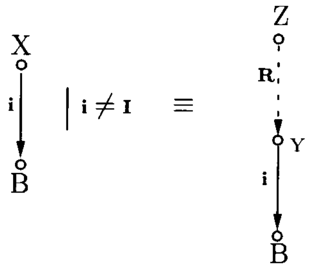


Each lexical paraphrasing rule implies the application of one or several syntactic rules for readjusting the structure to the lexical changes produced. Obviously, for the lexical fission rules, the syntactic operation of fission is obligatory. As for the distribution of the DSyntAs between the different nodes that constitute a SVC, the syntactic paraphrasing rules as presented in (Mel'čuk, 1992:45-52) are ambiguous. In the following section, we discuss this distribution.

5.3 *Distribution of DSyntAs in the Syntactic Fission Rule*

Let us begin with the meta-rule formulated by Mel'čuk (1992:45-46) for the control of the syntactic operations imposed by lexical paraphrasing.

Meta-rule:

> If a syntactic rule of fission is applied, the dependent nodes of the fissioned node X that are not mentioned in the rule are distributed among the two new nodes Z-**R**→Y in the following way:
>
> – The actant **I** of X remains as dependent node of the top node of the new tree:
>
> 
>
>
> – Other actants of X become actants of Y:
>
> 
>

Note that this meta-rule is valid for $\mathbf{Oper}_\mathrm{i}$ with two DSyntAs only.

Next, we show how syntactic paraphrasing rules operate to serve the aforementioned lexical paraphrasing rule, i.e., $C_{0[V]} \equiv \mathbf{Oper}_1\text{-}\mathbf{II}{\rightarrow}\mathbf{S}_0(C_0)$. One fission and two transfer operations that are controlled by the meta-rule are necessary. Consider the corresponding rules and their application to the DSyntS of the first sentence in (21), which we repeat here, together with the resulting sentence for the convenience of the reader:

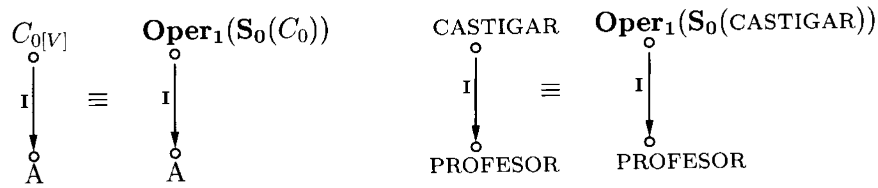
(22) *El profesor castigó* [C_0] *al niño*
'The teacher punished the boy'.

≡

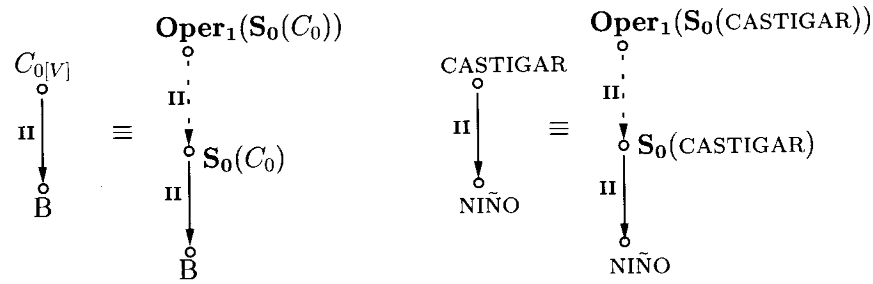
El profesor puso [$\mathbf{Oper}_1(\mathbf{S}_0(C_0))$] *un castigo* [$\mathbf{S}_0(C_0)$] *al niño*
'The teacher imposed a punishment on the boy'.

1. Fission of a node into a branch with **Oper₁**:

2. Transfer of DSyntA **I**:



3. Transfer of DSyntA **II**:

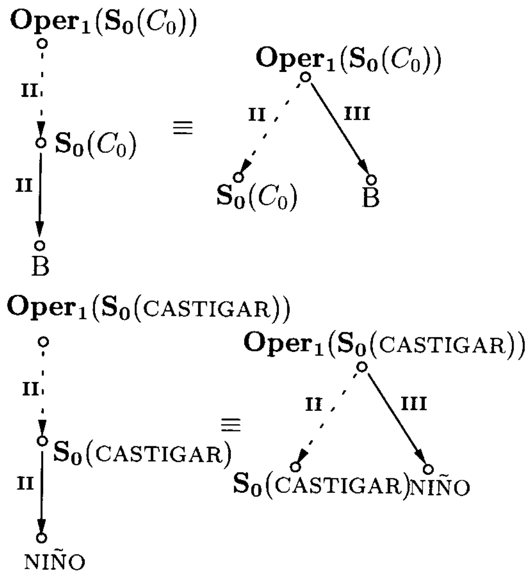


According to the meta-rule, the DSyntA **II** of CASTIGAR '[to] punish' becomes a DSyntA of the noun CASTIGO 'punishment'. However, in the SSyntS, EL NIÑO 'the boy' is treated as an Indirect Object of the support verb—which is marked by its doubling by the dative clitic: *El profesor le puso un castigo al niño.* Therefore, the operations triggered by the meta-rule so far are not sufficient, and an additional transfer rule is necessary to transfer DSyntA **II** of CASTIGO to DSyntA **III** of **Oper** (see Mel'čuk, 1992, Rule 12); cf. Rule 2 in 4., next page.

Note that Rule 2 does not apply in the case of a biactantial Oper since all necessary syntactic operations are already specified in the meta-rule.

After this short examination of the paraphrasing rules, it seems advisable to make a distinction between different **Opers** and, thus, to consider the different government patterns of **Oper**-values. The syntactic operations which are to be performed differ depending on whether the value of the LF is biactantial or triactantial: only a triactantial value requires the application of Rule 2. Therefore, we propose to distinguish between two lexical paraphrasing rules: one which aims for an $\mathbf{Oper}_{1C2}$ and transfers the DSyntA **II** of the full verb as DSyntA **III** of the LF using Rule 2, and another which aims for an $\mathbf{Oper}_{1C}$ and which does not trigger Rule 2.

4. Transfer of DSyntA **II**, Rule 2:



6 Synthesis Procedure

In Section 4, we showed the correspondence between SemS and SyntS of SVCs from a static point of view. In the following, we evaluate first the three possibilities of the distribution of DSyntAs discussed in Section 3 from the application point of view. Then, we exemplify the passage from the SemR to the DSyntS for the example *Juan tiene <sufre de> alergia al polen* step by step.

6.1 *Evaluation of Possibility 1: Variable Diathesis*

The variable diathesis of $\mathbf{Oper}_i$ reflects the paraphrastic potential of the MTM, but as far as the implementation is concerned, this variability creates several problems. As Iordanskaja *et al.* (1996:283) point out, in the context of automatic text synthesis, the goal is not to produce all paraphrases for a given input, but to obtain only one output: "the best one in the given context." This means that the result of the processing of a given SemR must be a single sentence. In order to obtain a sentence with the meaning encoded in the SemR shown in (8), one has to choose from the outset one of the two possible DSyntSs for the SVC.

A sentence generator such as MATE (Bohnet *et al.*, 2000) is able to handle the mapping between a DSyntS with a biactantial **Oper** and a SSyntS with a triactantial support verb and *vice versa* by accounting for flexible rule conditions. Thus, the specification of a condition that the surface-syntactic top node must be a triactantial verb will exclude a verb like SUFRIR '[to] suffer' as a governor of three SSyntAs. Consider a sample DSynt-rule encoded in the MATE-format:

```
        leftside:  Xds-II→Yds
       rightside:  Xss-indobj→Yss-prep→Zss
                   Zss.lex := Yds.lex
                   Yss.lex := a
         context:  Zds-II→Xds
                   Xss
          conditions:  Zds.lex = Oper1
                       Lexicon::(Xss.lex).valency = triactantial
                       Zds↔Xss
 correspondences:  Yds⇔Zss
```

A few words on the syntax of the rule are in order. Xds, Yds, and Zds are DSynt-node variables. Xss, Yss, and Zss are SSynt-node variables. Both Dsynt-nodes and SSynt-nodes are structured object nodes. A structured object node consists of a set of feature-value pairs. One of the features is 'lex'. Its value is the name of the node in question, i.e., the lexeme that serves as the label of the node. ':=' stands for a feature-value assignment. '=' used in the condition-slot stands for an equality check. 'Lexicon::' denotes an access to the lexical resource named "Lexicon"; 'Lexicon::(α.lex).f' stands thus for 'retrieve from the entry for the lexeme bound to α.lex in Lexicon the value of the feature f'.

The 'leftside'-slot of the above rule specifies the fragment of a DSyntS that is to be mapped onto the fragment of a SSyntS specified in the first line of the 'rightside'-slot. The feature 'lex' of the SSynt-variable Zss is assigned the value carried by the feature 'lex' of the DSynt-variable (in our example, this is POLEN). The feature 'lex' of the SSynt-varable Yss is assigned the lexeme A. The 'context'-slot requires the presence of the branch Zds-**II**→Xds in the DSyntS to which this rule is applied and of the node Xss in the SSyntS being derived from this DSyntS. The 'condition'-slot contains the restrictions that the value of the feature 'lex' of the variable Zds be '**Oper$_1$**', the lexeme that is the value of the feature 'lex' of the variable

Xss be triactantial, and the DSynt-variable Zds be already mapped (by one of the previous rules) onto the SSynt-variable Xss. The 'correspondences'-slot establishes a correspondence between the DSynt-node bound to Yds and the SSynt-node bound to Zss.

In our example, Xss can be instantiated only by TENER, not by SUFRIR because only the former possesses 'triactantial' as value of the feature valency.

Unfortunately, the above rule presupposes a lexical entry for the support verb TENER. This assumption causes several problems. For instance, since, as stated before, the government of a support verb changes depending on the noun, it is not clear how many entries should be defined for a support verb such as TENER. Therefore, instead of specifying that the lexeme to which the SSynt-variable Xss is bound (TENER) must be triactantial, it is more appropriate to specify that the node Zds can only be realized as a triactantial **Oper**. Thus, we propose to replace the conditions from above by the following condition:

conditions: Zds.lex = Oper_{1C2}

This condition avoids producing an incorrect SSyntS with SUFRIR as the governor of an Indirect Object.

The construction of a SSyntS from a given DSyntS can be blocked either by the government of the value of an LF (as just stated), or by the government of its keyword. For instance, a supported noun in a SVC can govern two different prepositions in order to express its second DSyntA-slot. Thus, the noun MIEDO 'fear' governs the preposition A and DE, as in:[13]

(23) a. *Juan le tiene miedo a María* lit. 'Juan has fear to María'.
b. *Juan tiene miedo de María* lit. 'Juan has fear of María'.

Only in (23a), *María* is a SSyntA of the support verb. If *María* is a SSyntA of *miedo*, it is introduced by DE. The rule formalism as used in MATE also allows for the specification of the correspondence between a triactantial $\textbf{Oper}_1$(MIEDO) and a biactantial verb (example 23b). Consider the corresponding DSynt-rule:

leftside:	Xds-**III**→Yds
rightside:	Xss-**prep.com**→Yss-**prep**→ Zss
	Yss.lex := *de*
	Zss.lex := Yds.lex
context:	Xds-**II**→Zds
	Xss
conditions:	Xds.lex = Oper_1
	Zds ↔Xss
correspondences:	Yds⇔Zss

In this case, we have to specify that the only preposition governing the node Zss (*María*) is DE. But again, we wonder whether it would not be easier to avoid a rule like this one and to establish the correspondence between MIEDO-**II**→MARÍA and MIEDO-**prep.com**→DE-**prep**→MARÍA, presupposing the context TENER-**dirobj**→MIEDO.

As became clear, keeping the diathesis of a SVC-variable (or vague) presupposes the use of several *ad hoc* DSynt-SSynt rules—even when we want to produce only one sentence. However, if transition rules are interpreted as paraphrasing rules, several SSyntSs can be derived without a recourse to such *ad hoc* rules. In this case, one would start from a single SemS in order to construct two DSyntSs (one with **Oper** with three DSyntAs and another one with two DSyntAs). Each of these DSyntSs would then give rise to a specific SSyntS. That is, the interpretation of transition rules as paraphrasing rules still ensures that all possible DSyntSs are covered, but makes the passage towards the corresponding SSyntSs more straightforward.

6.2 *Evaluation of Possibility 2: Diathesis with Only Two DSyntAs*

From the application point of view, the diathesis of $\mathbf{Oper}_i$ with only two DSyntAs presents certain advantages. Thus, the transition between the SemS and the DSyntS is controlled by a single rule that distributes the DSyntAs between the LF and its keyword. This rule has been presented in the literature as a meta-rule of paraphrasing that controls the syntactic operations associated with the corresponding lexical paraphrasing rules (see page 122 above). It can also be encoded as a MATE-rule that controls the distribution of the DSyntAs; cf.:

leftside:	Xsem-**2/3/4**→Ysem
rightside:	Xds-**II/III/IV**→Yds
context:	Zds-**II**→Xds
conditions:	Zds.lex = Oper_i
correspondences:	Xsem⇔Xds
	Ysem⇔Yds

According to this rule, all but the first of the SemAs of the keyword are realized as DSyntAs of the noun rather than of the support verb. As a result, all DSyntSs that correspond to a SVC are uniform, and there is no multiplication of syntactic structures produced by the variable diathesis of $\mathbf{Oper}_i$. Furthermore, there are no DSyntSs that cannot be mapped to a SSyntS, and therefore, also no need for readjustment rules, such as, e.g., the one that transforms a triactantial structure into a biactantial one. However, the standardization of the DSyntS requires in certain cases more complex rules for the transition between the DSyntS and the SSyntS. For instance, for all triactantial support verbs, a rule would be necessary that transposes the DSyntAs **II** and **III** of the keyword to the support verb.

To summarize, we can state that the approach that pursues Possibility 1 (Subsection 6.3) makes all passages between the levels of an MTM more complex, while the approach that pursues Possibility 2 makes only passage from the DSyntS-level towards the SSyntS-level more complex.

6.3 *Possibility 3: Diathesis Differentiated Depending on SSyntS*

According to the third approach, the synthesis of a SVC has to take the government of the value of $\mathbf{Oper}_i$ into account. Depending on the number of SSyntAs of the support verb in question, different $\mathbf{Oper}_i$ will be distinguished. In this case, the rules in the MATE-format include as condition the valency of the governor node. Consider, for illustration:

leftside:	Xsem-**2**→Ysem
rightside:	Xds-**III**→Yds
	Yds.lex := Sem::Ysem.lex
context:	Xds-**II**→Zds
conditions:	Xds.lex = Oper_{1C2}
correspondences:	Ysem⇔Yds

'Sem' stands here for a semantico-lecical resource in which semantemes are assigned their signifiers.

Applying this rule to 'alergia'-**2**→'polen', MATE attaches the lexical node POLEN (bound to the variable Yds) as the DSyntA **III** of the governor node only when the governor is $\mathbf{Oper}_{1C2}$ (see the conditions-slot).

A different DSynt-rule establishes the correspondence between the deep-syntactic $\mathbf{Oper}_{1C2}$-**III**→POLEN and the surface-syntactic TENER→A→POLEN. Cf.:

leftside:	Xds-**III**→Yds
rightside:	Xss-**indobj**→Yss
	Yss-**prep**→Zss
	Xss.lex := Xds.lex
	Yss.lex := *a*
	Zss.lex := Yds.lex
context:	Xds-**II**→Zds
conditions:	Xds.lex = Oper_{1C2}
correspondences:	Yds⇔Zss

The result is TENER-**indobj**→A-**prep**→POLEN.

The value of the feature 'lex' of the DSynt-node Xds makes reference to a triactantial lexeme: in our example, only TENER will be encoded as $\mathbf{Oper}_{1C2}$(ALERGIA), while SUFRIR will be encoded as $\mathbf{Oper}_{1C}$(ALERGIA).

In this approach, the number of rules is smaller since no rules are needed to establish the correspondence between a biactantial **Oper**-node and triactantial **Oper**-values and *vice versa*. Furthermore, due to the use of the "looking ahead"-strategy, the synthesis of a SVC does not arrive at a dead-end, where no realization is possible. The strategy of looking ahead is equivalent to the introduction of the values of LFs at the DSynt-level—as has been proposed by Iordanskaja *et al.* (1996:284). Iordanskaja *et al.* opt for the introduction of the values of LFs during synthesis as early as possible, but they keep track of the origin of each lexeme obtained via the application of a specific LF as value of this LF. In our case, it is not necessary to know the lexeme itself; it suffices to know its diathesis. Thus, if a given $\mathbf{Oper}_i$ has several values, all having the same number of SSyntAs, there is no need to choose one of them at the DSynt-level. However, if an **Oper** has values with a different number of SSyntAs (i.e., with different diatheses), this must, of course, be specified in the entry for the keyword.

Our goal is to keep the DSyntSs of the **Oper**-SVCs of the same noun as similar as possible. Iordanskaja *et al.*'s proposal to introduce LF-values at the DSynt-level opposes this goal since being introduced at the DSynt-level, a specific **Oper**-value makes the corresponding DSynt-tree distinct from the trees of the other values of the same **Oper**. Therefore we propose to keep the name of the LF at the DSynt-level, and to postpone the introduction of concrete LF-values to the SSynt-level. An additional argument for keeping the name of the LF at the DSynt-level is the use of LF-names by paraphrasing rules, which operate at the DSynt-level.

6.4 *Passage from the Sem-Level to the DSynt-Level for a SVC*

Having examined the advantages and shortcomings of the three approaches to the distribution of the DSyntAs in a SVC, we use the last of these approaches to illustrate the passage from the Sem-level to the DSynt-level. The purpose of this section is to illustrate step by step the process of mapping the SemS in (8) onto the DSyntSs in (8a) and (8b).

Step 1: Application of the fission nodal rule. The only semantic node that can be chosen as starting node for processing the SemS (the so-called *entry node*) is the dominant node of the Rheme, 'alergia'. Due to the general constraint that for languages such as Spanish, the top node of a sentence must be a finite verb, a fission nodal rule must be applied.

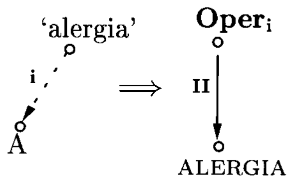
This rule is a composition of a lexicalization nodal rule and an arborization rule that inserts a new DSynt-node and a new DSynt-branch. The newly inserted DSynt-node does not correspond to any of the Sem-nodes at the source side. Encoded in the MATE-format, the fission nodal rule in question looks as follows:

leftside:	Xsem
rightside:	Xds-**II**→Yds
	Xds.lex := Oper_i
	Yds.lex := Sem::Xsem.lex
context:	Xsem-**i**→Ysem
conditions:	Lex::(Sem:: Xsem.lex).Oper_i
	Ysem.theme

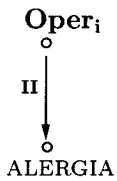
Being applied to the SemS in (8), this rule produces the DSyntS in (24b):

(24) Mapping the semantic entry node onto an $\textbf{Oper}_i$-SVC (as usual, the dashed arc in the left-hand side of the rule stands for the context)

a. Rule



b. Resulting structure



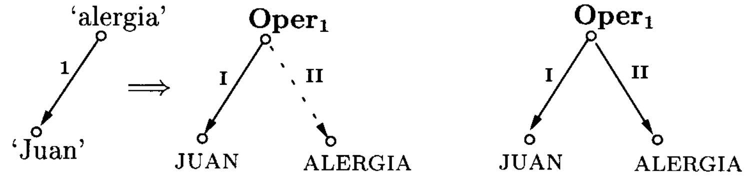
Step 2: Mapping SemA i onto DSyntA I To map the Sem-relation **i** onto the DSyntA I of the governor node, the following rule in the MATE-format must be applied.

leftside:	Xsem-**i**→Ysem
rightside:	Xds-**I**→Yds
	Yds.lex := Sem::Ysem.lex
context:	Xds-**II**→Zds
	Xds.lex = Oper_i
correspondences:	Ysem ⇔Yds

This rule presupposes that the DSynt-relation II with $\textbf{Oper}_i$ as head has already been introduced in the resulting structure.

(25) Mapping the semantic branch **1** onto the deep-syntactic branch **I**

a. Rule b. Resulting structure



Cf. both the substructure produced by this rule and the DSynt-structure under construction in (25).

Step 3: Mapping SemA 2. Two rules are available to map SemA **2** onto the DSynt-level. Both require that the semantic entry node has already been mapped onto an **Oper**-construction. In addition, the first requires a triactantial **Oper**, while the second requires a biactantial one. The availability of an **Oper**-LF with the required diathesis for a given keyword is checked by accessing the lexical entry of the keyword. In the entry for ALEGRIA, we find two different **Oper**-LFs: $\mathbf{Oper_{1C2}}$ = TENER and $\mathbf{Oper_{1C}}$ = SUFRIR.

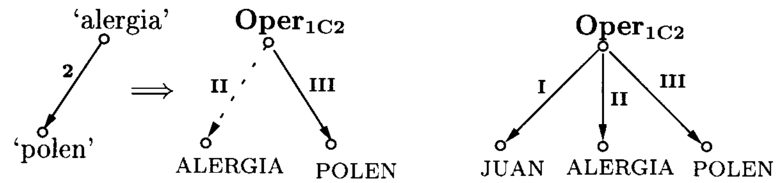
The first rule has already been presented in the previous section; we reproduce it here for the convenience of the reader:

leftside: Xsem-**2**⟶Ysem
rightside: Xds-**III**⟶Yds
Xds.diath := Oper_{1C2}
context: Xds -**II**⟶Zds
conditions: Lex::Zds.Oper_{1C2}
correspondences: Ysem ⇔Yds

Consider, again, the substructure produced by this rule and the structure constructed so far:

(26) Mapping of SemA **2** onto DSyntA **III**

a. Rule b. Resulting structure



The second rule looks as follows in the MATE-format:

leftside: Xsem-**2**→Ysem
rightside: Xds-**II**→Yds
Xds.diath := Oper_{1C}
context: Xds-**II**→Zds
conditions: Lex::Zds.Oper_{1C}
Xsem ↔Zds
correspondences: Ysem ⇔Yds

The application of this rule results in the following structures:

(27) Mapping of SemA **2** onto DSyntA **II**

a. Rule b. Resulting structure

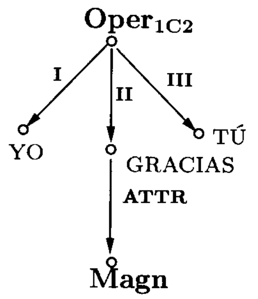
At the moment of the application of the nodal fission rule, we do not want to specify the diathesis of $\textbf{Oper}_i$ since this rule can also be applied to generate something like *Estás acatarrado? – No, tengo alergia* 'Have you a cold? – No, I have an allergy'. It is only when the semantic relation **2** is considered that we must verify the diathesis of **Oper**. If the lexical entry for the keyword includes a triactantial support verb, the semantic relation **2** can be realized as a DSyntA **III** of **Oper** and in this case, the application of the rule adds the feature "triactantial" (denoted by $\textbf{Oper}_{1C2}$) to the governor node (Figure in 26). In contrast, if the semantic relation **2** is realized as DSyntA of the keyword, the governor node will be assigned the default **Oper**-diathesis (denoted by $\textbf{Oper}_1$ or its equivalent $\textbf{Oper}_{1C}$).

7 Conclusion

Throughout this work we have emphasized the importance of an explicit encoding of the $\textbf{Oper}_1$-diathesis. Our proposal to distinguish between biactantial and triactantial Oper does not attempt to offer new LFs, but, rather, more specified subtypes of an LF. Thus, we have argued that an LF such as $\textbf{Oper}_1$ has two subtypes: a biactantial one (chosen by default) and a triactantial one. As demonstrated, this distinction allows a more straight-forward passage from a SemS to a SSyntS of a SVC in that it does not require rules that manipulate the actantiality of governor nodes at the DSyntS-level as well as at the SSynt-level. Although if considered only the $\mathbf{Oper}_1$-LF, we believe that other LFs show an analogous picture. Therefore, a detailed study of the diathesis of other LFs is needed.

Acknowledgments

The work described in this paper has been supported by the Spanish Ministry of Education and Science (HUM2005-08052-C02-02). The first sketch of the paper has been read and commented upon by the participants at the workshop in Saint-Just (Summer 1999): Lidija Iordanskaja, Sylvain Kahane, Igor Mel'čuk, Jasmina Milićević, Alain Polguère and Leo Wanner. Their suggestions significantly contributed to the improvement of this work. I express my deepest gratitude to all of them. The final text underwent the scrutiny of Leo Wanner and Igor Mel'čuk. Thank you, Leo, for your patience and your generous help, and, Igor, for your birthday, a little present:



Notes

[1]Even though the three types of support verbs present a similar problem, we will limit ourselves to support verbs that take the keyword as their first complement. Henceforth, except indications to the contrary, the term 'support verb' refers only to a SVC represented by the LF $\mathbf{Oper}_i$. In the following examples, the symbol 'ART' indicates that a determiner is necessary, the symbol '~' stands for the keyword.

[2]The first complement of a lexical unit L is either its direct object (if L can have one), its indirect object (if L cannot have a direct object), or the "strongest" prepositional object (in the case of the absence of the former two object types); see (Mel'čuk, 1996:61).

[3]Apart from the semantic level, all levels of linguistic representation are subdivided into a deep sublevel (*D-*) and a surface sublevel (*S-*). The deep sublevels are oriented towards meaning: their objective is to express all semantic distinctions that are relevant at the level in question. The surface sublevels are geared towards text: their objective

is to express all formal distinctions relevant at the level in question. For the distinction between the Deep-Syntactic level and the Surface-Syntactic level, see (Mel'čuk, 1988:58-69, Mel'čuk, 1997c:23-28).

[4]But note that the same problems appear in other languages as well.

[5]In terms of LFs, this rule is formulated as follows: $\mathrm{V} \Leftrightarrow \mathrm{Oper}_1(\mathrm{S}_0(\mathrm{V}))\text{-}\mathbf{II}{\rightarrow}\mathrm{S}_0(\mathrm{V})$.

[6]The term 'diathesis' as used in MTT does not correspond to what one usually finds in the traditional literature, where it is taken as the equivalent of grammatical voice (see Lazard, 1994, 1997). In MTT, the diathesis of a lexeme L is the correspondence between the SemAs and the DSyntAs of L (Mel'čuk, 1994:135).

[7]The term *complex predicate* as used in the literature covers a great variety of phenomena, starting with the denominal verbs in English and going to the applicative construction of Yoruba (see Alsina *et al.*, 1997). As Gunkel (1998) pointed out, this heterogeneity of phenomena makes the concept of the complex predicate very vague. A greater differentiation is needed. I use this term to refer to the situation in SVCs, where a verbal lexical unit provides syntactic positions, while a nominal lexical unit "feeds" these positions semantically. This interpretation is not completely in line with the interpretation in, e.g., (Rosen, 1989; Alsina *et al.*, 1997) and (Butt & Geuder, 2001). For these authors, in a complex predicate, two or more semantic nuclei contribute to the argument structure. I rather agree with Mohanan (1997), who underlines the following fact: in Hindi, complex predicates have a simple argument structure, although the verb and the noun constitute a phrase. In the present paper, I also defend the thesis that in a SVC, the arguments are semantically linked to a single predicate, although they are syntactically distributed between the verb and the noun.

[8]Being approximate, the SemR (8) includes nonetheless the semantic structure (SemS) and the semantic-communicative structure (SemCommS) of the corresponding sentence. SemS represents the propositional meaning, while SemCommS captures the speaker's communicative intention by making the distinction between Theme (denoted by 'T') and Rheme (denoted by 'R'). SemCommS also marks the communicatively dominant nodes of Theme and Rheme: 'alegria' in the Rheme-area and 'Juan' in the Theme-area. SemCommS plays a decisive role in the syntactic arborization of the semantic structure. Thus, the construction of the syntactic tree begins with the determination of the entry node that gives rise to the syntactic head of the tree. This determination is intimately bound to the SemCommS (Mel'čuk, 2001).

[9]Following Mel'čuk (1996:101), we use an arrow to indicate that a N depends syntactically only on the keyword (rather than on the verb SUFRIR). This notation is, in a certain sense, redundant: the information that the noun ALEGRIA selects the preposition A to introduce its DSyntA **II** can be extracted from the government pattern of this noun. However, without the arrow, the presentation of the two values of $\mathbf{Oper_1}$(ALEGRIA), TENER and SUFRIR, would be identical—despite the fact that only TENER possesses three actants. Another option would be to indicate the third DSyntA for TENER only: *tener* [~ a N]; *sufrir* [*de* ~].

[10]Let us note that we refer to actantional relations in syntax only. For instance, the semantic actantial relation between 'red' and 'apple' must necessarily be inverted in syntax, but in this case, there is no actantial relation involved—rather a modificative relation: APPLE-**ATTR**→RED.

[11]This is the strategy followed in the *Lexique actif du français*; see (Polguère, 2000).

[12]In general, in such a case, the introduction of a support verb LF (**Oper**, **Func** or **Labor**) or of a copular verb is obligatory. See syntacticization rules in (Mel'čuk,

2001:41): "If the entry node 's' is chosen in the initial SemS has a non-verbal lexical expression L('s'), then instead of L('s'), the model must put into DSyntS at its top node one of the following DSynt-configurations: BE-**II**→L('s'), $Oper_i$-**II**→L('s'), $Func_i$-**I**→L('s'),$Labor_{ij}$-**III**→L('s')." The rules for establishing the entry node were formulated for the first time by Polguère (1990).

[13]In fact, the government of MIEDO is more complex. MIEDO selects the preposition DE only in SVCs. Cf.?*El miedo de oscuridad es enfermizo* 'The fear of the darkness is pathological'; rather: *El miedo a oscuridad es enfermizo* 'The fear to the darkness is pathological'. But *Juan tiene miedo de la oscuridad* lit.'Juan has fear of the darkness' (i.e. in a SVC), DE is perfectly acceptable.

Bibliography

Alsina, A., J. Bresnan & P. Sells. 1997. *Complex Predicates.* Stanford: CSLI Publications.

Bohnet, B., A. Langjahr & L. Wanner. 2000. "A Development Environment for an MTT-Based Sentence Generator". *Proceedings of the First International Conference on Natural Language Generation.* Mitzpe Ramon, Israel.

Butt, M. & W. Geuder. 2001. "On the (Semi)lexical Status of Light Verbs". *Semi-lexical Categories. The Function of Content Words and the Content of Function Words* ed. by N. Corver & H. van Riemsdijk. 323–370. Berlin & New York: Mouton de Gruyter.

Daladier, A. 1996. "Le rôle des verbes supports dans un système de conjugaison nominale et l'existence d'une voix nominale en français". *Langages.* 121:35–53.

Di Sciullo, A. & S.T. Rosen. 1991. "Constructions à prédicats légers et quasi-légers". *Revue québécoise de linguistique.* 20:13–35.

Fontenelle, T. 1992. "Using a Bilingual Computerized Dictionary to Retrieve Support Verbs and Combinatorial Information". *Acta Linguistica Hungarica.* 41.1–4:109–121.

Grimshaw, J. & A. Mester. 1988. "Light Verbs and θ-Marking". *Linguistic Inquiry.* 19.2:205–232.

Gunkel, L. 1998. "Review of Alsina et al. (eds.), Complex Predicates, Stanford: CSLI". *Theoretical Linguistics.* 24.2–3:265–271.

Hausmann, F.-J. 1989. "Le dictionnaire des collocations". *Wörterbücher, Dictionaries, Dictionnaires, Volume 1* ed. by F.-J. Hausmann, O. Reichamann, H. E. Wiegand & L. Zgusta. 1010–1019. Berlin & New York: Walter de Gruyter.

Heid, U. 1996. "Using Lexical Functions for the Extraction of Collocations from Dictionaries and Corpora". *Lexical Functions in Lexicography and Natural Language Processing* ed. by L. Wanner. 115–146. Amsterdam & Philadelphia: Benjamins Academic Publishers.

Iordanskaja, L., M. Kim & A. Polguère. 1996. "Some Procedural Problems in the Implementation of Lexical Functions for Text Generation". *Lexical Functions in Lexicography and Natural Language Processing* ed. by L. Wanner. 279–297. Amsterdam & Philadelphia: Benjamins Academic Publishers.

Kahane, S. & I. Mel'čuk. 1999. "Synthése des phrases à extraction en français contemporain". *T.A.L.*. 40.2:25–85.

Kahane, S. & A. Polguère. 2001. "Formal Foundation of Lexical Functions". *Proceedings of the Workshop COLLOCATION: Computational Extraction, Analysis and Exploitation. Held at the 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics.* 8–15. Toulouse.

Lazard, G. 1994. *L'actance.* Paris: Presses Universitaires de France.

Lazard, G. 1997. "Actance, diathèse : questions de définition". *Bulletin de la Société de Linguistique de Paris.* 93:115–136.

Mel'čuk, I. 1988. *Dependency Syntax: Theory and Practice.* Albany: State University of New York Press.

Mel'čuk, I. 1992. "Paraphrase et lexique : la théorie Sens-Texte et le Dictionnaire explicatif et combinatoire". *Dictionnaire explicatif et combinatoire du français contemporain. Volume III.* ed. by I.A. Melčuk. 9–59. Montréal: Presses de l'Université de Montréal.

Mel'čuk, I. 1993. *Cours de morphologie générale. Volume 1.* Montréal/Paris: Presses de l'Université de Montréal/Èditions du C.N.R.S.

Mel'čuk, I. 1994. *Cours de morphologie générale. Volume 2.* Montréal/Paris: Presses de l'Université de Montréal/Èditions du C.N.R.S.

Mel'čuk, I. 1996. "Lexical Functions: A Tool for the Description of Lexical Relations in the Lexicon". *Lexical Functions in Lexicography and Natural Language Processing* ed. by L. Wanner. 37–102. Amsterdam & Philadelphia: Benjamins Academic Publishers.

Mel'čuk, I. 1997a. "Cas grammaticaux, construction verbale de base et voix en massaï: vers une meilleure analyse des concepts". *Bulletin de la Société de Linguistique de Paris.* 92.1:49–113.

Mel'čuk, I. 1997b. *Cours de morphologie générale, volume 4.* Montréal/Paris: Presses de l'Université de Montréal/Èditions du C.N.R.S.

Mel'čuk, I. 1997c. *Vers une linguistique Sens-Texte. Leçon inaugurale.* Paris: College de France.

Mel'čuk, I. 1998. "Collocations and Lexical Functions". *Phraseology. Theory, Analysis and Applications* ed. by A. P. Cowie. 23–53. Oxford: Oxford University Press.

Mel'čuk, I. 2001. *Communicative Organization of Sentences in Natural Language.* Amsterdam & Philadelphia: Benjamins Academic Publihshers.

Mel'čuk, Igor A. 2004a. "Actants in Semantics and Syntax I: Actants in Semantics". *Linguistics* 42(1):1–66.

Mel'čuk, Igor A. 2004b. "Actants in Semantics and Syntax II: Actants in Syntax". *Linguistics* 42(2):247–291.

Mohanan, T. 1997. "Multidimensionality of Representation: NV Complex Predicates in Hindi". *Complex Predicates* ed. by A. Alsina, J. Bresnan & P. Sells. 431–471. Stanford: CSLI Publications.

Polguère, Alain. 1990. *Structuration et mise en jeu procédurale d'un modèle linguistique déclaratif dans un cardre de génération de texte.* PhD thesis: Département de linguistique, Université de Montréal.

Polguère, A. 2000. "Towards a Theoretically-Motivated General Public Dictionary of Semantic Derivations and Collocations for French". *Proceedings of the EURALEX International Congress '2000.* 517–527. Stuttgart.

Rosen, S.T. 1989. *Argument Structure and Complex Predicates.* PhD thesis: Brandeis University.