HOW TO DESIGN LEXICAL AND CONSTRUCTIONAL TEMPLATES

A STEP BY STEP GUIDE

Ricardo Mairal and Francisco Ruiz de Mendoza

1. The motivation: Why do we need lexical templates?

1.1. Lexical templates grow out as an attempt to (i) provide richer semantic description than those provided by logical structures as posited in Role and Reference Grammar

sing:	**do'** (x, [**sing'** (x)])
show:	[**do'** (x, ∅)] CAUSE [BECOME [**see'** (y, z)]]

These two examples illustrate the inconsistency that RRG shows when one tries to represent the meaning of a predicate. Why do we use *sing* and a more elaborate decomposition in the case of *show*? Does RRG provide any criteria to choose where the chain of semantic decomposition actually ends? The answer is no. So, we need to stipulate a mechanism that actually accounts for more elaborate semantic descriptions and provides a procedure indicating the type of primitive that we are supposed to use.

1.2. If we want to provide richer semantic representations, the question that arises is why don't we adopt frames as in Frame Semantics? Although the methodology is very similar (both approaches propound core and peripheral elements as part of the semantic description of a predicate), frames lack a unified formalism that is methodologically close to constructional templates. This is an important point since then unification, which is the cardinal operation in the model, becomes a more straightforward free redundancy process. Recall that lexical and constructional template share the view that both are based on aktionsart distinctions, although the approach and the classes differ.

1.3. Lexical template are meant to be typologically valid representations such that we can represent a predicate like, say, *sleep* or *cry* in a wide range of different languages by using a metalanguage other than English. This is a real challenge for a theory that aspires to typological adequacy. As things stand, if we want to define *llorar, recibir*, o *aprender*, are we supposed to use English (as in (1)) or Spanish (as in (2)):

(1)

a. llorar:	**do'** (x, [**cry'** (x)])
b. recibir	BECOME **have'** (x,y)
c. aprender	BECOME **know'** (x, y)

(2)

a. llorar:	**do’** (x, [**llorar’** (x)])
b. recibir	BECOME **tener’** (x,y)
c. aprender	BECOME **saber’** (x, y)

One of the challenges is to develop a universal semantic metalanguage which makes sense in every language of the world. In fact, lexical templates are based on this universal semantic metalanguage.

1.4. We might ask ourselves if there is no preceding work within RRG which antedates these ideas. In this regard, Van Valin and Wilkins (1993) and Van Valin and LaPolla (1997: chapter 3) provide some tentative representations along the following lines:

The predicate *remember* (Van Valin and Wilkins, 1993: 511):

BECOME think.again (x) about something.be.in.mind.from.before (y)

Speech act verbs and the entry for *promise* (Van Valin and La Polla, 1997:117):

a. **do’** (x, [**express**(α).**to**(β).**in.language**.(γ)’ (x,y)])
b. **do’** (x, [**express**(α).**to(**β**).in.language**.(γ)’ (x, y)]) CAUSE [BECOME **obligated’** (x, w)
α = w
β = y

In subsequent work Mairal and his associates bengan to develop more articulated and richer semantic representations:

The lexical class of *contact-by-impact* verbs (Mairal, 2003)
[[**do´** (w, [**use.tool.**(α)**.in.**(β)**.manner.for.**(δ)´ (w, x)] CAUSE [**do´** (x, [**move.toward´** (x, y) & INGR **be.in.contact.with´** (y, x)], α = x.

The problem in the preceding examples resides in the fact that *express, obligated, think, move* are regarded as primitives. There seems to be no reason or explicit criteria for such a decision. It is as though *express,* for example, had been plucked out of the air without the use of any heuristic procedure for designating a set of primitives or, for that matter, an inventory of semantic fields, not to mention a description of their internal organization.In this sense it is doubtful that **obligated’**, which is posited as the primitive predicate for the lexical entry for *promise*, can even be remotely called a primitive. Furthermore, although *think* and *move* are putative universals, nothing is said of how these predicates have been arrived at or where they have come from. Although couched in more elaborate semantic decompositions, Mairal's (2003) lexical templates are still not systematic enough in their use of activity and state primitives. Primitives such as *manner*, *tool*, and *use* appear in these representations, but no explanation is given of how they have been obtained.

In essence, there are sufficient criteria to formulate a new system of lexical represnations based on a universal semantic metalanguage.

2. **The metalanguage in a lexical template**:

As any other natural language, a universal semantic metalanguage (e.g. a language that makes sense crosslinguistically) must be based on a syntax and a semantics. The semantic module contains the set of words (in our case primes or primitives) that we will use for our representations. We know your reaction; how can I identify these primes? Is there any heuristic procedure? The answer is yes, as will be shown in 3.1.

Moreover, once we have a catalogue of primes we will need something that allows us to combine them to represent more specific hyponyms. For example, if we hypothesize that *know* is a primitive how can we represent its hyponyms, e.g. *learn, understand* etc? We will need something – in our case a number of lexical functions – that will account for the idiosyncratic properties of hyponyms. At this stage, we can also provide you with an inventory of lexical functions as those in the Appendix. Recall that these functions have been extracted from Mel'cuk's *Explanatory and Combinatorial Lexicology* (ECL) framework Mel'cuk, 1989; Mel'cuk et al., 1995; Mel'cuk and Wanner, 1996; Alonso Ramos,2006)[1].

There is a basic corollary which has become a must. The final inventory must be **systematic, finite, and with some sort of internal organization**. It is simply not feasible to create an endless *ad hoc* list of semantic primitives to be used every time the need arises. Furthermore, any such inventory must be created on the basis of methodological principles that justify all choices.

2.1. **The semantic part**

- This is the more complex part of a template since not only do we have to decide the type of primitives and the set of semantic parameters that define a given predicate but also how to combine the two together into a user friendly notational device.

- Decide the type of primitive involved. This is a crucial part and your decision depends on the lexical domain to which the predicate you want to define belongs to. Here is a chart that might help you decide the type of primitive involved in each domain:

[1] We refer the reader to the project's website for an update account of the most recent work: http://www.fas.umontreal.ca/ling/olst/

Lexical domain	Nuclear term
EXISTENCE	be/happen
CHANGE	become
POSSESSION	have
SPEECH	say
EMOTION	feel
ACTION	do, make
COGNITION	know, think
MOVEMENT	move (go/come)
PHYSICAL PERCEPTION	see / hear / taste / smell / touch
MANIPULATION	use

This means that all predicates that belong to COGNITION will be defined in terms of two primitives: KNOW and THINK. If we move on to the domain of POSSESSION and SPEECH, we will use HAVE and SAY. There are two interesting domains: Exitence nd Change that we will have to look at more carefully.

There are two questions that are relevant: The first concerns the type of typological empirical evidence that we have to show that these primitives exist in every language. The second has to do with the exhaustiveness of the inventory; why don't we extend this classification to nominal, adjectival primitives?

In relation to the first question, the inventory of primitives coincides with those of Wierzbicka's Natural Semantic Metalanguage Approach, which has been shown to be valid over a hundred languages. As for the second, that is still a task for a future project.

- Then, so far we only have the primitive and now we need to define the hyponyms, which means the identification of the semantic parameters that define a given predicate. For example, if we are asked to represent those predicates of SPEECH that belong in the subdomain 'to cause sb to believe sth, using words or not', we only know that the primitive we will use is *say*.

- A further step consists in determining the distinguishing parameters. In doing so, we suggest that you should look up the meaning of words in a dictionary. This will help you identify these parameters. Look at the following lexical subdomain and try to identify the set of distinguishing semantic parameters:

(3)

<u>To cause sb to believe sth, using words or not</u>

convince to cause sb to believe sth, using words or not.

persuade to convince sb to do/believe sth by reasoning with them.

coax to try to persuade sb to do sth in a pleasant way.

wheedle to try to persuade sb to do sth by saying pleasant things you do not mean.

proselytise to try to persuade sb to believe the way you do [religion] [politics].
cajole to persuade sb to do sth they do not want to do by saying pleasant things which are not true.
inveigle to persuade sb to do sth by trickery.
dissuade to persuade sb not to do/believe sth.
deter to dissuade sb from doing sth by showing them that the result would be unpleasant.

- There are recurrent patterns that pervade the lexicon: intensification, culmination of a process, the beginning, continuation or the cease of a process, the time axis (past, present, future), the place etc. All of these notions have a corresponding **lexical function** (see appendix 2).

2.2. The syntactic module

What is a lexical function?

A lexical function (LF) is written as:

(4)
$\mathbf{f}(x) = y$,

where f represents the function, *x*, the argument, and *y,* the value expressed by the function when applied to a given argument. The meaning associated with an LF is abstract and general and can produce a relatively high number of values. For example, **Magn** is a function that expresses intensification, It can be applied to different lexical units and produces a high set of values:

(5)	**Magn**	(smoker)	=	heavy;
(6)	**Magn**	(bachelor)	=	confirmed
(7)	**Magn**	(error)	=	craso
(8)	**Magn**	(llorar)	=	llorar como una Magdalena

Here are some more examples of the lexical functions **Able** ('such that someone or something can be /do/ have'), **Result** (the state or endpoint of a process) and **Mult** ('a set of')[2]:

(9)	**Able** $_1$	(envidia)	=	envidioso
(10)	**Able** $_2$	(envidia)	=	envidiable
(11)	**Result**	(aprender)	=	saber
(12)	**Mult**	(perro)	=	jauría

(Mel'cuk *et al*, 1995: 126-127; Alonso Ramos, 2005).

[2] For a more detailed description of the whole set of lexical functions, we refer the reader to the work done by Margarita Alonso Ramos, from where these examples have been extracted.

However, unlike Mel'cuk's lexical functions, our inventory of operators is used to organize the lexicon vertically instead of horizontally (see 4 and 5 below).

- Let us consider the following sample and try to identify how *know* combines with lexical functions to express the meaning of these predicates:

(13)

confuse to cause sb not to understand sth.

puzzle to confuse sb, causing them to think for a long time.

confound to confuse sb causing them to feel surprise and doubt.

perplex to confuse sb, causing them to feel worry.

It is clear that, although they all share the same semantic scenario (e.g. *to cause sb not to understand sth,*) the three hyponyms (*puzzle, confound* and *perplex*) encode different distinguishing properties that allow us to differentiate one from the rest; in the case of *perplex*, there is a feeling of worry, something which is not the case with *confound* and *puzzle*. A similar argument could be used to define *confound*, which encodes a feeling of surprise and doubt, which contrasts with *puzzle*, which focalises the temporal dimension of continuity**.** So, a possible formalization of these notions could be posited along the following lexical functions:

(14)

confuse to cause sb not to understand sth.

do (x, 0) CAUSE [BECOME NOT **know**' (y, z)]

(15)

puzzle to confuse sb, causing them to think for a long time.

do (x, 0) CAUSE [BECOME NOT **know**' (y, z)]

CAUS INVOLV CONT(think)$_{12}$ = AN ACTVITY IS INVOLVED SUCH THAT 1 THINKS ABOUT 2 FOR A LONG TIME

(16)

confound to confuse sb causing them to feel surprise and doubt.

do (x, 0) CAUSE [BECOME NOT **know**' (y, z)]

INVOLVCAUSSYMPT$_{12}$ (FEELING_TYPE: surprise, doubt)

(17)

perplex to confuse sb, causing them to feel worry.

do (x, 0) CAUSE [BECOME NOT **know**' (y, z)]

INVOLVCAUSSYMPT$_{12}$ (FEELING_TYPE: worry = AN ACTVITY IS INVOLVED SUCH THAT
1 EXPERIENCES A SYMPTOM OF WORRY.

- Scope of lexical functions. After analyzing an important part of the verbal lexicon we have observed that lexical functions refer to the following parameters:
  - Manner.
  - Instrument.
  - Temporal dimension:
  - The length of a process in terms of its inception, duration or culmination.
  - The nature of the second argument:

(18)

Expect: [FACT$_{12}$ (true)$_2$ / FACT $_{12}$ (become)$_2$] **think'** (x, y)

The lexical entry in (18) designates a state logical structure which is subject to two possible interpretations; (i) the effector's evaluation that the proposition will be true in the future; (ii) the effector's mental disposition that some action will occur in the future. These two interpretations have important consequences for the type of morphosyntactic structure of the complements and in fact this can be codified in terms of a number of lexical rules of the following type:

(19)

FACT (true) [y] =	Clausal Subordination
FACT (become) [y] =	Core cosubordination
PURP do [y] =	Core cosubordination
INVOLV$_{12\ (\text{ACTION})}$ [y] =	Core coordination
INVOLV$_{12\ (\text{see})\ 12}$ [y] =	Core cosubordination
ALL $_2$ [y] =	Clausal subordination

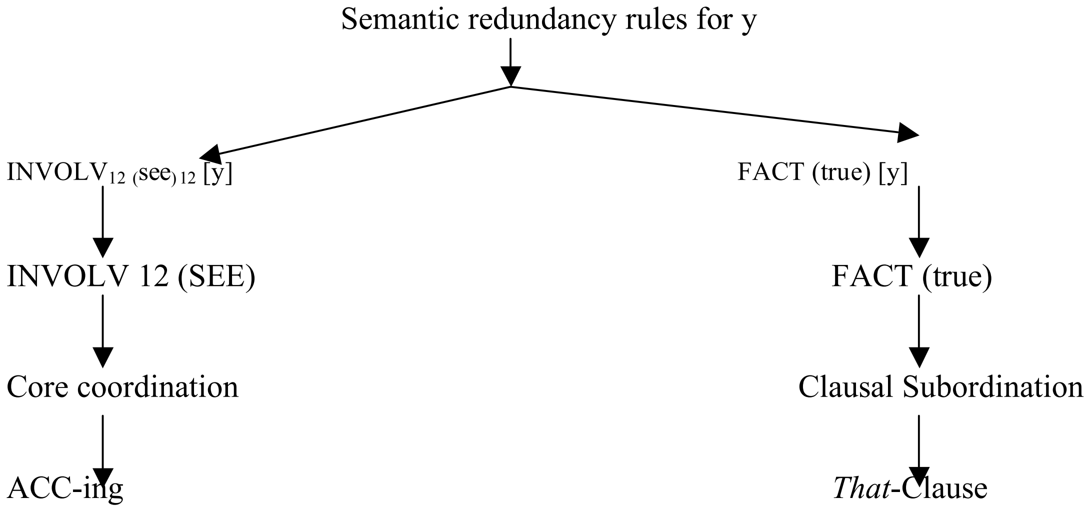
These lexical rules have been shown to have great predictive power in terms of the syntactic clause linkage. Consider the following predicate:

(19)

imagine: [FACT $_{12}$ INVOLV$_{12}$(see)$_{12}$ LOC$_{in}$ (abstract body_part: mind)] **think'** (x, y)

The lexical entry in (19) consists of a state predicate defined by the primitive *think* and two arguments such that the first argument thinks about a mental percept (FACT). The semantic component specifies an activity that involves *see*ing (semantic primitive) it in his mind. This predicate entertains two possible interpretations of the second argument, i.e. as a mental proposition and as a direct percept. This can be seen in the following diagram:

(20) Semantic redundancy rules for y



2.3. **The logical structure**

- This is the easiest part since your choice is very much determined by the type of Aktionsart as discussed in Van Valin (2005). We are aware that there are predicates that are certainly controversial and subject to different interpretations, but these are the least.

- In order to identify the basic Aktionsart, Van Valin (2005: 2.1.1.) has formulated seven tests that you can always follow[3].

3. The formalism

At this stage, we have an inventory of primes and a catalogue of lexical functions that have cross-linguistic validity. Now, a further issue that arises concerns the development of a formalized system for lexical representation based on this new metalanguage. In

[3] For a description and comment on these tests, we also refer you to Mairal and Cortés (2006).

connection with this, we believe that logical structures can be maintained and be enriched by adding a new semantic component as shown in the following schema:

[semantic representation] + logical structure = lexical entry

The semantic module is encoded by means of lexical functions that are essentially paradigmatic and captures those pragmatic and semantic parameters that are idiosyncratic to the meaning of a word and also serve to distinguish one word off from others within the same lexical hierarchy. The second module follows the orthodox practice of RRG as described in section. This module includes a description of the *Aktionsart* properties that are characteristic of a given predicate together with the set of variables that have a syntactic impact. In order to mark the difference between the properties of the two modules we will be using two types of variables: *internal* and *external* variables. Internal variables –which are marked with numerical subscripts- refer to the semantic parameters of a predicate, while the latter –which are represented by Roman characters- capture syntactically relevant meaning aspects[4]. An important feature is that all units involved in the lexical representation of the predicate must have a universal status, that is, all the units must be part of the metalanguage

4. **Case studies**:

- Before commenting on various cases, let us just offer a comparative view of the predicate regret as encoded in RRG and in our approach:

Regret

a. regret' (x, y) RRG

b. [SYMPT (sadness) INVOLV$_{12}$ (want)DEGRAD (do)$_2$ LOC$_{in}^{temp\leftarrow}$/(become)2 LOC$_{in}^{temp\leftarrow}$] **feel'** (x, y)

Regret is a commentative predicate which means that it asserts an emotional reaction and takes the complement as background (cf. Noonan, 1985). In such a case, the LS proposed in (a) ignores this semantic interpretation and uses **regret** as a primitive, a decision which is highly questionable. In contrast (b) includes a semantic representation which describes the emotional reaction that the effector experiences. The effector feels/experiences an emotion, a physical symptom (SYMPT) of sadness about *y*, an event. Furthermore, there is a subactivity INVOLV$_{12}$ implied by the predicate such that the effector wants the second argument (an event) not to have happened (*become*) in the past. Moreover, this representation is typologically adequate since it only uses elements that are extracted from the metalanguage. In this regard, **feel** seems to be a better candidate than **regret** as a primitive.

[4] The formalism we use is an adaptation to our framework of the one in Van Valin and LaPolla (1997: 117), who mark external variables with Roman characters and internal variables with Greek letters.

- **DOMAIN OF COGNITION**

Grasp: [MagnObstr & Culm$_{12[[\text{ALL}]]}$] **know'** (x, y)

The entry for *grasp* has two parts: (i) the semantic component in brackets; (ii) the representation of the logical structure. In this case, this predicate is represented by a state logical structure which takes **know'** as a primitive and has two arguments (x, y). This logical structure is in turn modified by a lexical function (or operator) MagnObstr, which specifies the large degree of difficulty involved in carrying out the action. The other lexical function, Culm, captures the end-point of knowing something (which is understanding). Note that grasp is a hyponym of understand and consequently inherits all the semantic properties of the superordinate. ALL is another lexical function -of the kind postulated by Mel'cuk- that falls within the scope of the internal variables. In our proposal, this lexical function refers to the propositional content of the object of apprehension.

Realize: [Instr (see)$_{12}$Loc$_{\text{in}}$ (body_part: mind) & Culm$_{12[\text{ALL}]}$] **know'** (x, y)
where x = 1 and y = 2

The lexical entry for *realize* has a state logical structure which is inherited from the superordinate term (*understand*, Culm$_{12[\text{ALL}]}$) and a semantic description of the idiosyncratic properties of this predicate. This semantic part, INSTR (see)$_{12}$LOC$_{\text{in}}$ (BODY_PART: mind) [understand], is interpreted as follows: the cognizer (arg1) comes to know or understand a mental percept (arg2). This is done by *seeing* (semantic primitive) it in his/her *mind* conceptualized as a location (LOC$_{\text{in}}$). The mind is represented as an abstract body_part, which means it is in a partitive relationship to *body*. Note that this predicate inherits the properties of its immediate superordinate, a feature that we have marked with the symbol &.

Interpret: [Instr (Caus (have))$_{12\ (\ (\text{other})\ \text{ALL})\ 3}$ & Culm$_{12[\text{ALL}]}$] **know'** (x, y)

In the case of *interpret* we also have a state in the logical structure plus a semantic description, Instr (Caus (have))$_{12\ (\ (\text{other})\ \text{ALL})\ 3}$ (*understand*, Culm$_{12[\text{ALL}]}$)]. The cognizer (arg1) understands a mental percept (arg2) causing it (CAUS) to *have* (semantic primitive) an*other* (semantic primitive) meaning.

Reminisce: [Involv(say)$_{12}$Sympt$_1$ (feeling_type: pleasure)] **think'** (x, y)

The predicate *reminisce* inherits the properties of *remember* and thus encodes a state logical structure, which is in turn modified by a semantic representation with the following format: Involv(say)$_{12}$Sympt$_1$ (feeling_type: pleasure) [remember]. *Reminisce* has the same basic meaning as *remember,* but involves (Involv) a cognizer (arg1) saying (*say*) things about it (arg2). This subactivity produces physical symptoms or feelings (SYMPT) in the cognizer/experiencer, which come from feeling_type: pleasure.

Consider: [$\text{LOC}_{\text{in}}^{\text{TEMP}\leftrightarrow}{}_{12}$ CONT] **think'** (x, y)

The representation for *consider* includes a state logical structure and a semantic representation $\text{LOC}_{\text{in}}^{\text{TEMP}\leftrightarrow}{}_{12}$ CONT [think], which is interpreted as follows: a cognizer (arg1) thinks about a mental percept (arg2) located in the present *(*$\text{LOC}_{\text{in}}^{\text{TEMP}\leftrightarrow}$) for a long time (CONT).

Study: [PurpMagn(know)$_{12}$ & $\text{Loc}_{\text{in}}^{\text{temp}\leftrightarrow}{}_{12}$ Cont] **think'** (x, y)

As in the previous examples, the state logical structure for *study* is enriched by a semantic representation PURPMAGN(know)$_{12}$ [consider1]. In *study,* a cognizer (arg1) thinks about a mental percept (arg2) located in the present. PURP codifies the purpose of the action, which is for the cognizer to *know* (semantic primitive) more (MAGN) about the mental percept (arg2).

Plan: [PlusIntent$_{12}$Purp(do)$_{12}$ $\text{Loc}_{\text{in}}^{\text{temp}\rightarrow}$] **think'** (x, y)

Apart from the state logical structure, the predicate in (37) includes a semantic description in the following terms: PLUSINTENT$_{12}$PURP(do)$_{12}$ $\text{LOC}_{\text{in}}^{\text{TEMP}\rightarrow}$ [think]. In *plan* the cognizer (arg1) thinks about a mental percept (arg2) with intentionality (PLUSINTENT). The purpose (PURP) of his action is to *do* (semantic primitive) it in the future ($\text{LOC}_{\text{in}}^{\text{TEMP}\rightarrow}$).

Persuade: [[FACT$_{12}$ (true)$_2$ / FACT$_{12}$ (become)] INSTR (say)$_{123}$] [**do'** (x, Ø)] cause [become **think'** (y, z)]

The lexical entry for *persuade* includes a causative accomplishment logical structure which is in turn modified by a semantic component. Firstly, this component specifies the semantic nature of the second argument which is: (i) the expression of the effector's attitude or judgement regardint the truth of the proposition (this reading is in fact inherited from its superordinate *believe*); (ii) a mental disposition regarding a possible action. Besides, there is an instrumental (INSTR) parameter that indicates that the effector carries out this action by *say*ing (semantic primitive) something to y. As shown elsewhere, the two interpretations are very revealing of the type of morphosyntactic structure and syntactic clause linkage of the predicate and the complement.

5. CONSTRUCTIONAL TEMPLATES

Constructional templates are inspired in the work of construction-based approaches like Goldberg (1995, 2005). A grammar consists of an inventory of constructions, which are in turn defined as form meaning pairings. Goldberg (1995:3-4) identifies the following inventory:

Type of construction	**Semantic representation**
Ditransitive *Pat sent Hill a fax*	X CAUSES Y TO HAVE Z
Caused motion *Pat sneezed the napkin off the table*	X CAUSES Y TO MOVE TO Z
Resultative *She kissed him unconscious*	X CAUSES Y TO BECOME Z
Intransitive *The fly buzzed into the room*	X MOVES Y
Conative *Sam kicked at Bill*	X DIRECTS ACTION AT Y

- Constructional templates, such as those discussed in Goldberg, are in turn linked to lexical entries, each of which is elaborated on the basis of Fillmore's *Frame Semantics*.
- A frame, like a lexical template, describes situation types (e.g. buying and selling, eating, etc.) and includes a very rich semantic description of a lexical item by integrating what they call core and non- core elements, which coincide to a large extent with our distinction between external and internal variables. The purpose of constructing rich semantic descriptions by integrating central and peripheral elements within a semantic scenario is shared by both frames and templates.
- The difference is that lexical templates include a much more formalized representation, something which brings them closer to the notational device used for constructional templates. In a way we could say that lexical and constructional templates have the same metalanguage, in such a way that unification becomes a more straightforward process, which is not the case with the notation used in frame semantics. For the sake of illustration, consider our proposed format of the caused-motion construction:

do' (x, [**pred'** (x, y]) CAUSE [BECOME NOT **be-in'** (y,z)]

pred' (x, y) CAUSE [BECOME NOT **be-in'** (y,z)]

Note that what is characteristic of this construction is that there is an induced phenomenon which causes a change of location. The second part is a recurrent pattern (e.g. BECOME NOT **be-in'** (y, z)]) in every representation of the constructional template, while the first part varies between an activity and a state template:

do' (x, [**pred'** (x, y)])

[**pred'** (x, y)]

In order to constrain this part, we need to account for the semantic properties of the construction, which, like in the case of lexical templates, are formalized by a number of lexical functions. At this stage, for example we can venture that an expression like *Peter loved Mary back into life*, which is a state predicate, would be encoded by the primitive **feel’**.

A different case is the resultative construction, which introduces another logical structure which focalizes the result obtained in a process (be this an activity or an accomplishment). Then, a simplified format of the semantic representation of this construction would be the following:

$$[\text{LS1}]\ \text{CAUSE}\ [\text{LS2}].$$

The first template is usually an activity or a causative accomplishment while the second lexical template is saturated by the operator BECOME and a nucleus which is usually identified with either an adjectival or a prepositional phrase. The final template for the resultative phrase indicates the existence of two predicates (e.g. the main predicate and the result predicate).[5]

$$\begin{tikzcd}
& {[\text{LS1}]\ \text{CAUSE}\ [\text{LS2}].} \arrow[dl] \arrow[dr] & \\
\mathbf{do'}\ (x, [\mathbf{pred'}\ (x, y)]) & & {[\text{BECOME}\ \mathbf{pred'}\ (y)]}
\end{tikzcd}$$

An interesting question is whether it is possible to constrain the types of activity that can be part of this construction. This is an issue which we have dealt with by formulating a number of external constraints.

[5] In fact, these verbs act as separate nuclei and function as a complex predicate, thus resulting in a nuclear juncture, an aspect of the theory that is beyond the scope of this paper.

APPENDIX 1:
List of semantic primes in Wierzbicka's NSM

Grammatical category	Wierzbicka's Semantic Primitives
nouns	I, YOU, SOMEONE, PEOPLE, SOMETHING/THING, BODY
determiners	THIS, THE SAME, OTHER
quantifiers	ONE, TWO, SOME, ALL, MANY/MUCH
evaluators	GOOD, BAD
descriptors	BIG, SMALL, (LONG)
intensifier	VERY
mental predicates	THINK, KNOW, WANT, FEEL, SEE, HEAR
speech	SAY, WORD, TRUE
actions, events and movement	DO, HAPPEN, MOVE
existence and possession	THERE IS, HAVE
life and death	LIVE, DIE
time	WHEN/TIME, NOW, BEFORE, AFTER, A LONG TIME, A SHORT TIME, FOR SOME TIME, MOMENT
space	WHERE/PLACE, HERE, ABOVE, BELOW; FAR, NEAR; SIDE, INSIDE; TOUCHING
"logical" concepts	NOT, MAYBE, CAN, BECAUSE, IF
augmentor:	MORE
taxonomy, partonomy	KIND OF, PART OF;
similarity	LIKE

Appendix 2: LIST OF LEXICAL FUNCTIONS

Lexical Function	**Definition**
ECL Lexical Functions (with their application adapted to paradigmatic structure)	
ANTI	Antonym. This LF also combines with other LFs to negate them.
BON	Good (expression of praise)
CAUS	Cause
CONT	Continuity/duration
CULM	The highest point of []
DEGRAD	To get worse
FACT	Be realized
INCEP	The beginning of []
INSTR	Instrument
INVOLV	Subactivities implied by the predicate
LOC$_{ad}$	Spatial location with directionality “to”
LOC$_{in}$	Spatial location with directionality “in”
LOC^{temp}	Temporal location which can have arrows marking past ($\overleftarrow{\quad}$), present (↔) or future ().
MAGN	intense(ly), very [intensifier], to a very high degree
MINUS	less of []
OBSTR	to function with difficulty
PERM	permit
PLUS	more of
SYMPT	physical symptoms
Additional lexical functions used in this article	
EFF	effort
POSS	possibility
PROB	probability
PURP	purpose