Kahane, S. et A. Polguère (2001) Formal foundation of lexical functions. Actes du colloque *COLLOCATION: Computational Extraction, Analysis and Exploitation*, Toulouse, pp. 8-15.

Formal foundation of lexical functions

Sylvain Kahane
LaTTiCe/TALaNa, UFRL, case 7003
Université Paris 7 - Denis Diderot,
75251 Paris Cedex 05, France
sk@ccr.jussieu.fr

Alain Polguère
OLST — Département de linguistique et de traduction
Université de Montréal, CP 6128, Succ. Centre-Ville
Montréal (Québec) H3C 3J7, Canada
Alain.Polguere@umontreal.ca

Abstract

Lexical functions were introduced in Meaning-Text theory to account for two interrelated families of lexical relations: semantic derivations and collocations. However, the language traditionally used in Meaning-Text lexicography for encoding lexical functions is not fully specified from a formal point of view. In this paper, we propose a formal foundation for lexical functions based on two complementary encoding devices which we believe are computationally tractable and fill the gaps of the traditional formal language of lexical functions.

1 Introduction

A proper treatment of collocations is required by most high-level NLP applications, such as machine translation and text generation. For instance, it is required in order to automatically translate French *gros fumeur*, lit. ‘big smoker’, into its English counterpart *heavy smoker*. In this given case, an MT system has to first identify that *gros fumeur* is a *COLLOCATION*. Following Meaning-Text terminology, we call *collocation* a linguistic expression made up of at least two components:

1. the *BASE* of the collocation: a full lexical unit (see *fumeur*) which is “freely” chosen by the speaker;
2. the *COLLOCATE*: a lexical unit (see *gros*) or a multilexical expression which is chosen in a (partially) arbitrary way to express a given meaning and/or a grammatical structure contingent upon the choice of the base.

Once the MT system has recognized that *gros* is a collocate of *fumeur* expressing intensification, it has to “know” that the intensifier of *smoker*, the translation for *fumeur*, is *heavy*, and not *big*, *large*, *thick* or *fat* (other common translations for *gros*). It could seem at first glance more straightforward to directly store in a bilingual index that *heavy smoker* is the proper translation for *gros fumeur*. However, such approach will lead system developers with no choice but building huge bilingual indexes of collocations for each language pair they want to handle. It is more rational to develop rich monolingual dictionaries describing collocations controlled by each lexical unit and to limit the scope of bilingual lexicons to establishing correspondences between lexical entries. Moreover, unlike bilingual collocation indexes, monolingual dictionaries of the above-mentioned type are fully reusable in the context of other NLP applications.

Collocations are numerous and various in nature. Some lexical units are the base for hundreds of collocations, expressing very different meanings, with a variety of syntactic structures. Such is the case of most nouns of feeling such as Fr. COLÈRE ‘anger’, whose lexical description is extensively used in the present study: *colère aveugle*/*noire*/…, lit. ‘blind/black/… anger’, *colère sourde*/*froide*, lit. ‘deaf/cold anger’, *fou*/*ivre de colère*, lit. ‘mad/drunk of anger’, *rouge*/*blanc de colère*, lit. ‘red/white of anger’, etc., to mention just a few examples. This shows that a powerful formal language is needed in order to encode base-collocate relations in reusable computational dictionaries.

In the context of early works on MT systems, Žolkovskij and Mel’čuk (1965) introduced the concept of *LEXICAL FUNCTION* (LF) to model base-collocate relations between lexical units. A formal language for describing these relations by means of LFs has been developed and used extensively in lexicographic descriptions found in *EXPLANATORY COMBINATORIAL DICTIONARIES* (ECDs) — see (Mel’čuk and Zholkovsky, 1984) for Russian and

(Mel'čuk et al., 1984, 1988, 1992, 1999) for French. However, the formal bases of the *ECD ENCODING* of LFs have never been made totally explicit, leaving researchers with a formal device that seems loose from both a computational and conceptual point of view. We believe it is essential to address this issue as the concept of lexical function itself has proved to be particularly suited for applications in computational lexicography (see Fontenelle, 1997; Polguère, 2000a) and NLP (see Iordanskaja et al., 1996; Polguère, 2000b).

In this paper we introduce two new formal encodings of LFs, each serving well-specified purposes.

The first alternative encoding we propose is computationally tractable and makes explicit the inner value and role of LFs in natural language, thus making it easier for Meaning-Text outsiders to understand and manipulate them. Because it makes explicit all formal properties of lexical relations, we hereafter refer to it as the *EXPLICIT ENCODING*.

The second alternative encoding is defined on the first one and is closer to the ECD encoding. It is made up of a closed set of simple LFs from which higher level LFs are obtained by (algebraic) combinations of simple LFs. We hereafter refer to this encoding as the *ALGEBRAIC ENCODING*.

Another encoding, based on "controlled" natural language, the *LAF ENCODING*, has been developed by Mel'čuk and Polguère (Polguère 2000a) for a general public dictionary for French (the *Lexique Actif du Français* = LAF).

This paper is organized as follows: Section 2 introduces the notion of semantic derivation and its link to collocations; Section 3 gives a theoretical perspective on LFs; Section 4 introduces the notion of LF encoding, focussing on the ECD encoding; Sections 5 and 6 are devoted to the presentation of our explicit and algebraic encodings, respectively; Section 7 relates these encodings to the ECD and LAF encodings; Section 8 is a brief conclusion.

2 Semantic derivations

Base-collocate relations introduced in Section 1 are syntagmatic relations between lexical units. In addition to these, LFs can be used to model paradigmatic relations, termed *SEMANTIC DERIVATIONS* in Meaning-Text lexicology.

Typical semantic derivations are (i) (quasi)synonymy/antonymy, (ii) verbal, nominal, adjectival or adverbial derivations, and (iii) name of a participant or circonstant — e.g. CRIME is linked by semantic derivations with AUTHOR [*of a crime*] or CRIMINAL, VICTIM, INSTRUMENT [*of a crime*], etc. Such relations are called **semantic** derivations as no morphological link needs to exist between lexical units involved, contrary to standard (morphological) derivation.

Collocations and semantic derivations are conceptually linked. For instance, if one wants to express *rain* with an intensification, one can opt for a collocate of *rain* such as *torrential* or a semantic derivation of *rain* such as *downpour*. The lexical relations between *rain* and *torrential* and *rain* and *downpour* are related by the fact that *torrential rain* and *downpour* are paraphrases. This shows that both types of lexical relation could and should be encoded by means of the same conceptual device, namely LFs.

3 A closer look at the notion of LF

In order to understand the rationale behind the term *lexical function* (put forward in Žolkovskij and Mel'čuk, 1965), it is necessary to first notice that base-collocate relations are oriented. For instance, *heavy* is a collocate (acting as intensifier) of the base *bombardment* in *heavy bombardment*, and not the other way round. Semantic derivations are also oriented. For instance, MURDERER is the standard name of the **first actant** of MURDER_{V}. Conversely, MURDER_{V} designates the **action** performed by a MURDERER.

Because they are oriented, these lexical relations can be modeled by means of **functions**, accounting for their inherent orientation, hence the name *lexical function*.

The notation $\mathbf{f}(L_1)=L_2$ means that a lexical relation **f** holds from L_1 to L_2. We call L_1 the *KEYWORD* and L_2 the *VALUE* of **f**. Using this functional notation, it is therefore possible to encode the two above-mentioned relations holding between MURDER_{V} and MURDERER as:

1st actant(*murder*$_{\text{V}}$) = *murderer*
action(*murderer*) = *murder*$_{\text{V}}$.

Because several lexical units can be linked to a lexical unit L_1 by **f**, **f** is not exactly a mathematical function.[1]

Each LF **f** corresponds to a linguistically homogenous set of lexical relations. In other words, if **f**(L_1)=L_2 and **f**(L'_1)=L'_2, then L_2 provides roughly the same linguistic features to L_1 as does L'_2 to L'_1, that is, the same ratio of semantic content and the same modification of syntactic behavior. Consequently, an LF can be viewed as some sort of "generalized" lexical unit (see Wanner, 1996:23) whose signifier can only be known once **f** is combined with (applied to) the keyword. In other words, contrary to true lexical units, LFs are not associated with specific realizations. Their realizations depend on their contexts of application, that is on the keywords.

In order to be able to postulate LFs, i.e. generalizations upon lexical relations, the linguistic "content" associated with each particular LF has to remain vague. To illustrate this point, we will take the standard LF of intensification, called **Magn**, which is somehow an idealization of "pure" intensification. It is never expressed as such for at least two reasons.

First, the intensification of the meaning of a lexical unit L is in fact always the intensification of a component of this meaning. For instance, while *deadly* in *deadly combat* applies to the number of casualties, *fierce* in *fierce combat* applies to the actual intensity of the combat. The case of the LF of "realization" **Real$_i$** is even more striking. The meaning expressed by **Real$_1$**(*recommendation*) in *to follow a recommendation* is obviously distinct from the meaning expressed by **Real$_1$**(*car*) in *to drive a car*.

The second reason why **Magn** is never expressed as such is that values returned by **Magn**(L) themselves correspond to full lexical units that have their own specific meaning. Therefore, even if we should consider that *intense* and *fierce* are both equivalent **Magn** of *combat*, it is still possible to identify semantic differences between the two collocations *intense combat* and *fierce combat* on the basis of the definitional meaning of the two corresponding collocates.

[1] For this reason, *value* denotes the **set** {L_2} of lexical units linked to L_1 by **f** in standard Meaning-Text terminology.

Whether for **Magn**, **Real$_1$** or any other LF, it is theoretically possible to opt for more *GRANULARITY* in the encoding and postulate more than just one LF for a given set of lexical links. However, if the concept of LF has to retain its descriptive and generalization power, there is no doubt that each unit of description has to be rather coarse. For example, lexical choices in MT or in text generation cannot always be perfect and it is better to consider too many values (noise) than to miss a valid value (silence). Complex strategies can be envisaged in order to make a choice among several possible values, exploring for instance the actual lexical meaning of each value (Mel'čuk and Wanner, 2001).

4 Encoding LFs

4.1 The notion of encoding

An encoding of LFs is a correspondence between the set of LFs and a formal language such that any natural operation on LFs will be associated with an operation in this formal language. In other words, if an LF **h** is understood to be the result of some form of "combination" of two LFs **f** and **g**, then the encoding of **h** in the formal language should be the result of the application of a formal operation to the encodings of **f** and **g**.

A lexical function denotes a set of pairs of lexical units, linked by the corresponding lexical relation. Therefore, one can see the encoding of LFs as a correspondence between the set $\mathbf{L^2}$ of all possible pairs of lexical units and a formal language. For instance, the pair **(***combat*, *fierce***)** will correspond to **Magn**, **(***car*, *to drive***)** will correspond to **Real$_1$**, and **(***car*, *fierce***)** will not correspond to any element of the formal language.

An encoding E defines a partition of $\mathbf{L^2}$. A given encoding E_1 is said to be more *GRANULAR* than another encoding E_2 if E_1 defines a finer partition of $\mathbf{L^2}$ than E_2, that is, if E_2 collapses together some LFs which are considered separately by E_1.

4.2 ECD encoding of LFs

The modeling of LF relations offered by Meaning-Text theory provides computational linguistics with a conceptual foundation that we believe should be kept almost as it is. What we propose

to deeply revise is **how** LFs should be formally accounted for: what we termed in Section 1 the *ECD ENCODING*. This encoding is made up of a set of about sixty "primitive" LF relations and of rules for combining them (for a presentation, see Mel'čuk, 1996). While Meaning-Text literature usually describes the basic lexicon of the ECD encoding, it is interesting to note that no detailed account has been made of all the rules governing the combination of the units of this lexicon.

Formulas used in this encoding, although linear and apparently homogeneous in nature (**Magn**, **Oper$_1$**, **CausOper$_1$**, etc.), account for two distinct properties of the lexical relation they encode: a *SEMANTIC CONTENT* and a *SYNTACTIC FRAME* of behavior. Thus, for instance, **IncepOper$_1$**(*disease*) = *to contract* [ART ~] encodes the following information:

1. semantic content: *to contract a disease* means ‘to start [see **Incep**] to experience [see **Oper$_1$**] a disease’;
2. syntactic frame: *to contract* is a verb that takes the noun *disease* as complement and the first actant (see the subscript 1 of **Oper$_1$**) of this noun as grammatical subject in order to express the above semantic content.

5 Towards an explicit encoding of LFs

The explicit encoding we propose describes each LF relation holding between two lexical units by means of two distinct formulas: the encoding of the LF's semantic content and the encoding of its associated syntactic frame. We will examine successively (Sections 5.1 and 5.2) each of these two representational components.

5.1 LFs' semantic content

The semantic content of an LF is a configuration of predicate-argument relations holding between *PRIMITIVE LF MEANINGS*. These primitive meanings correspond to some LFs already identified by Meaning-Text theory. They are primitive in that they will not be accounted for by means of other LF meanings. Primitive meanings are named using the standard symbols found in ECDs (although these symbols refer to the whole LF in the ECD encoding) followed by the argument structure between square-brackets. We list below all primitive meaning that will be used in this paper, using the following template of presentation:

`<Formal encoding>`: ‘<Semantic gloss>’.

`Incep[`*Arg*`]`: ‘*Arg* begins’
`Caus[`*Arg$_1$*`,` *Arg$_2$*`]`: ‘*Arg$_1$* causes *Arg$_2$*’
`Magn[`*Arg*`]`: ‘*Arg* is intense’
`AntiMagn[`*Arg*`]`: ‘*Arg* is little’
`Plus[`*Arg*`]`: ‘*Arg* increases’
`Minus[`*Arg*`]`: ‘*Arg* decreases’
`Fact[`*Arg*`]`: ‘*Arg* functions’
`Real[`*Arg$_1$*`,` *Arg$_2$*`]`: ‘*Arg$_1$* realizes *Arg$_2$*’
`Manif[`*Arg$_1$*`, (`*Arg$_2$*`)]`: ‘*Arg$_1$* manifests itself (in *Arg$_2$*)’
`Sympt[`*Arg$_1$*`,` *Arg$_2$*`]`: ‘*Arg$_1$* takes place, revealed by *Arg$_2$*’
`Non[`*Arg*`]`: ‘*Arg* does not hold’

In addition to primitive LF meanings, some special notations are used to refer to specific meanings:

- #: meaning of the keyword;
- 1, 2, 3 …: first, second, third … semantic actants of the keyword;
- Ω: other (unspecified) semantic participant.

Some formulas may have to include non standard components. These cannot be formalized and are simply introduced between semantic quotes (‘…’). Examples below illustrate the use of special keywords and symbols for arguments. We use actual LF relations controlled by Fr. COLÈRE, whose predicate-argument structure is ‘colère de X envers Y à cause de Z’ (‘X's anger towards Y due to Z’):[2]

- `Caus[2/3,#]`
  [= ‘Y/Z causes anger’]
  E.g. *Y/Z **met** X en colère* (lit. ‘Y/Z puts X in anger’)
- `Sympt[#,‘poings de X’]`
  [= ‘X feels anger, which is revealed by X's fists’]
  E.g. *X **serre les poings** de colère* (lit. ‘X squeezes the-fists of anger’)

Primitive meanings can be combined to form more complex meanings through predicates recursively taking arguments:

- `Caus[1,Minus[Manif[#]]]`
  [= ‘X causes a decrease in the manifestation of his anger’]
  E.g. *X **étouffe** sa colère* (lit. ‘X suffocates his anger’)

Components can also be combined by using the infix operator ^ expressing specification/characterization:[3]

---

[2] COLÈRE (‘anger’) is described in great detail using ECD encoding in (Mel'čuk et al., 1984).

• #^Magn
[= ‘anger which is intense’]
E.g. ***rage***

Curly brackets {…} indicate a meaning functioning as context; i.e. it is not part of the LF's actual semantic content.

Finally, parentheses (…) are used to specify the scope of operators when needed:

• {1}^(#^Magn)
[= ‘[X] such that his anger is intense’]
E.g. [*X*] ***fou de*** *colère* (lit. ‘mad of anger’)

5.2 LFs' syntactic frame

The syntactic frame of a given LF **f** provides two types of information on possible values of **f**(L):

1. their part of speech —there are only four parts of speech: V(erb), N(oun), A(djective) and Adv(erb);
2. their diathesis —that is, the list of their syntactic dependents in increasing order of oblicity.

For instance, the formula V[2,1] denotes a verbal value for **f**(L) taking the second (semantic) actant of the keyword L as subject and its first actant as first complement. It is not necessary to encode more precisely the exact syntactic realizations of the syntactic arguments (for instance *direct object* rather than *indirect*), but it is essential to encode the oblicity in order to account for the communicative organization of the resulting structure. Consider the communicative contrast between the three values obtained below:

$\begin{pmatrix} \# \\ V[\#,1] \end{pmatrix}$ (*colère*) = *habiter* ‘to live in’ [N=X]

E.g. *Une grande colère habitait Jean.*

$\begin{pmatrix} \# \\ V[1,\#] \end{pmatrix}$ (*colère*) = *éprouver* ‘to feel’ [ART ~]

E.g. *Jean éprouvait une grande colère.*

$\begin{pmatrix} \# \\ V[2,\#] \end{pmatrix}$ (*colère*) = *encourir* ‘to incur’ [ART ~]

E.g. *Pierre encourait sa* [=*Jean*] *colère.*

The above examples show the use of complete formulas of the explicit encoding. They are matrices made up of semantic content (first row) and syntactic frame (second row) subformulas. For instance, the first matrix expresses that *habiter* is an "empty" verb that takes *colère* as grammatical subject and the first actant of *colère* as complement in order to express that a feeling of anger is experienced by someone.

For A and Adv values, which are meant to function as syntactic modifiers, the governor is indicated as first element in the argument list, followed by ^. For instance, A[1^] denotes an adjectival value that functions as modifier of the first actant of the keyword:

$\begin{pmatrix} \{1\}\hat{}\# \\ A[1\hat{}] \end{pmatrix}$ (*colère*) = *fâché* ‘angry’.

This example shows that *fâché* is an adjectival constituent that can function as modifier of the first actant of *colère* (*la colère de X* ‘X's anger’ → *X fâché* ‘angry X’) in order to characterize this actant as being involved in a "situation of anger."

The contrast between *PARADIGMATIC* and *SYNTAGMATIC* LFs (roughly, semantic derivations *vs.* collocations) is directly available in the explicit encoding: if, as is the case in the above formula, # does not appear in the syntactic frame component of an LF formula, the value of **f**(L) is a semantic derivation; otherwise, the value is a collocate. Contrast the following formula with the preceding one:

$\begin{pmatrix} \{1\}\hat{}(\#\hat{}Magn) \\ A[1\hat{},\#] \end{pmatrix}$ (*colère*) = *fou* [*de* ~].

While A[1^] represents an adjective, A[1^,#] represents a word (e.g. a preposition) or a phrase which once combined with the keyword will form an adjectival phrase.[4]

6 Algebraic encoding of LFs

The algebraic encoding, directly inspired by the ECD encoding, is based on linear unidimensional formulas that synthesize both the informa-

[3] The use of the ^ operator is motivated by the fact that we want to take into account communicative relations holding between meanings (and not just predicate-argument relations). For instance, *to lose* is a predicate taking *battle* as argument in both *to lose a battle* and *a lost battle*; but in the first case the communicative head is *to lose* and in the second case, it is *battle*. Note that we do not include in our formulas the argument of Magn[], which is always identical to its communicative governor (i.e. what appears at the left-hand side of the ^ operator).

tion on the semantic content and syntactic frame of LFs. It uses a finite number of *SIMPLE LFS*, all other LFs being expressed by some form of concatenation of these simple LFs. We will first give the definition of some simple LFs in terms of matrices of the explicit encoding. Later, we will show that complex LFs, and their corresponding formulas in algebraic encoding, can be obtained by means of operations performed on simple LFs.

Simple LFs

We list below ten simple LFs, together with their corresponding matrices in explicit encoding. The syntactic frame of these matrices is underspecified as actual values for **f**(L) can possess more syntactic actants than those introduced in the standard matrix associated with **f**.[5]

$$\mathbf{Func_0} := \begin{pmatrix} \# \\ V[\#] \end{pmatrix} \quad \mathbf{Func_i} := \begin{pmatrix} \# \\ V[\#,i] \end{pmatrix}$$

$$\mathbf{Oper_i} := \begin{pmatrix} \# \\ V[i,\#] \end{pmatrix} \quad \mathbf{Caus} := \begin{pmatrix} Caus[\Omega,\#] \\ V[\Omega,\#] \end{pmatrix}$$

$$\mathbf{Manif} := \begin{pmatrix} Manif[\#,\Omega] \\ V[\#,\Omega] \end{pmatrix}$$

$$\mathbf{Non} := \begin{pmatrix} Non[\#] \\ pos(\#)[\#] \end{pmatrix} \quad \mathbf{Incep} := \begin{pmatrix} Incep[\#] \\ pos(\#)[\#] \end{pmatrix}$$

The expression pos(#) denotes the part of speech of the keyword: **Non** and **Incep** do not impose any part of speech for their value; they behave as some kind of syntactic "chameleons."

$$\mathbf{Magn} := \begin{pmatrix} \{\#\}\hat{}Magn \\ A[\#\hat{}] \end{pmatrix} \text{ or } \begin{pmatrix} \{\#\}\hat{}Magn \\ Adv[\#\hat{}] \end{pmatrix}$$

$$\mathbf{A_i} := \begin{pmatrix} \{i\}\hat{}\# \\ A[i\hat{},\#] \end{pmatrix} \quad \mathbf{Adv_i} := \begin{pmatrix} \{i\}\hat{}\# \\ Adv[i\hat{},\#] \end{pmatrix}$$

Composition of LFs

The first "natural" operation on LFs that we may think of is the *COMPOSITION* of LFs, that is, the application of an LF **g** to the application of another LF **f**: **g**∘**f**(L) = **g**(**f**(L)).

As has already been mentioned in Meaning-Text literature, this operation bears very little interest in terms of lexicographic description: if **f**(L_1) = L_2 and **g**(L_2) = L_3, this does not imply that an LF relation holds between L_1 and L_3.

Take for instance the case of the adjective *sour* (*a sour apple*/*dish*/*liquid*/…). The most neutral value for **Oper**$_1$(*sour*) is *to be* (*This apple is sour*). But what collocational value can be returned for **Incep**(*to be*)? There does not seem to be any other choice than the very general *to start*: *This is salty* → *This starts to be salty.* (We ignore here the non collocational, fused value *to become*.) Clearly, it would be farfetched to pretend that an LF relation holds between *sour* and *to start*; they cannot even combine, contrary to a base and its collocate. On the other hand, there is definitely a special relation holding between *sour* and *to turn* (*It turned sour*), which is not the result of a composition of LFs, but the result of another operation: the product.

Product of LFs

The *PRODUCT* is the most productive mode of combination of LFs. Consider two syntagmatic LFs **f** and **g**. Their product **h** is a syntagmatic LF such that

1. **h**(L) is a collocate of L;
2. **h**(L) is a paraphrase for **g**∘**f**(L)⊕**f**(L), where the ⊕ symbol denotes the *LINGUISTIC UNION* (i.e. standard linguistic combination) of the linguistic elements it connects.

The product **h** of **g** and **f** is noted **g.f** in the algebraic encoding. It is formally defined as follows.

Let c(**f**) be the content of **f**, d(**f**) be the diathesis of **f**, and pos(**f**) be the part of speech of **f**. The *PRODUCT* **g.f** of **g** and **f** is rewritten in the explicit encoding as:

$$\mathbf{g.f} := \begin{pmatrix} c(\mathbf{g}):\#\rightarrow c(\mathbf{f}) \\ pos(\mathbf{g})[d(\mathbf{g}):\#\rightarrow d(\mathbf{f})] \end{pmatrix}$$

In other words, c(**g.f**) is equal to c(**g**) where # is replaced with c(**f**), and d(**g.f**) is equal to d(**g**) where # is replaced with d(**f**). Thus **g.f**(L) has both the same meaning and the same syntactic diathesis than the multilexical expression **g**∘**f**(L)⊕**f**(L).

For instance, the products of **Incep** and **Caus** with an LF **f** are defined by the following formulas of the explicit encoding:

[4] Note that *en colère* 'angry' will not be treated as a collocation of *colère*, but as a semantic derivation. It behaves exactly as an adjective and not as an adjectival phrase (for instance it can be modified by *très* 'very': *très en colère* vs. **très fou de colère*).

[5] Due to this underspecification, the algebraic encoding (as well as the ECD encoding) is less granular than the explicit encoding.

$$\mathbf{Incep.f} := \begin{pmatrix} \mathrm{Incep}[\mathrm{c}(\mathbf{f})] \\ \mathrm{pos}(\mathbf{f})[\mathrm{d}(\mathbf{f})] \end{pmatrix};$$

$$\mathbf{Caus.f} := \begin{pmatrix} \mathrm{Caus}[\Omega, \mathrm{c}(\mathbf{f})] \\ \mathrm{V}[\Omega, \mathrm{d}(\mathbf{f})] \end{pmatrix}.$$

Our *sour*/*to turn* problem can now be solved. It is a product of LFs, namely $\mathbf{Incep.Oper_1}$(*sour*) = *to turn* [~], as the following paraphrase relation holds: *to turn* [=$\mathbf{Incep.Oper_1}$(*sour*)] *sour* ≡ *to start* [=$\mathbf{Incep} \circ \mathbf{Oper_1}$(*sour*)] *to be* [=$\mathbf{Oper_1}$(*sour*)] *sour*.

The collocate *to turn* [*sour*] is properly accounted for by the following explicit encoding formula, which defines $\mathbf{Incep.Oper_1}$:

$$\begin{pmatrix} \mathrm{Incep}[\#] \\ \mathrm{V}[1, \#] \end{pmatrix}.$$

An interesting property of the LF product is that it is a potentially unbounded mode of combination, which is associative; that is: $\mathbf{h.(g.f)} = \mathbf{(h.g).f}$. For instance:

$$\mathbf{Caus.Non.Manif} = \begin{pmatrix} \mathrm{Caus}[\Omega, \mathrm{Non}[\mathrm{Manif}[\#]]] \\ \mathrm{V}[\Omega, \#] \end{pmatrix}$$

Fusion and paradigmatic LFs

In order to account for the link that can exist between some syntagmatic and paradigmatic lexical relations, we introduce the operation of *FUSION*.[6] It associates to each syntagmatic LF **f** a corresponding paradigmatic LF $/\!/\mathbf{f}$ such that $/\!/\mathbf{f}$(L) is a paraphrase of L⊕**f**(L).

In most cases, the effect of the fusion operator // is to remove # from the syntactic frame of **f**.[7] For instance:

$$/\!/\mathbf{Oper_i} = \begin{pmatrix} \# \\ \mathrm{V}[\mathrm{i}] \end{pmatrix} \; ; /\!/\mathbf{A_i} = \begin{pmatrix} \{\mathrm{i}\}\hat{}\# \\ \mathrm{A}[\mathrm{i}\hat{}] \end{pmatrix}$$

Product involving a paradigmatic LF

Consider a syntagmatic LF **f** and a paradigmatic LF **g**. Their product **h** is a syntagmatic LF such that

1. **h**(L) is a collocate of L;
2. **h**(L) is a paraphrase for **g**∘**f**(L).[8]

The product of **g** and **f**, where **g** is a paradigmatic LF, is still written **g.f** in the algebraic encoding even though the "generic" definition in terms of explicit encoding does not apply here. Another definition is required for this specific type of product, namely:

$$\mathbf{g.f} := \begin{pmatrix} \mathrm{c}(\mathbf{g}):\#\rightarrow\mathrm{c}(\mathbf{f}), \mathrm{i}\rightarrow\mathrm{i}(\mathbf{f}) \\ \mathrm{pos}(\mathbf{g})[\mathrm{d}(\mathbf{g}):\mathrm{i}\rightarrow\mathrm{i}(\mathbf{f})] \end{pmatrix}$$

where **i** is any i[th] actant of # and **i(f)** any corresponding i[th] actant of **f**. For instance:

$$/\!/\mathbf{A_i.f} = \begin{pmatrix} \{\mathrm{i}(\mathbf{f})\}\hat{}\mathrm{c}(\mathbf{f}) \\ \mathrm{A}[\mathrm{i}(\mathbf{f})\hat{}] \end{pmatrix}$$

$$/\!/\mathbf{A_2.Manif} = \begin{pmatrix} \{\Omega\}\hat{}\mathrm{Manif}[\#, \Omega] \\ \mathrm{A}[\Omega\hat{}, \#] \end{pmatrix}$$

E.g. (*Un geste*/*regard*/…) *rempli* [*de colère*]
(lit. ‘a gesture/look/…filled with anger’)

7 Relation between encodings

The algebraic encoding has been made as close as possible to the ECD encoding. In most cases they are identical except for the fact that the ECD encoding does not explicitly encode operations on LF; for instance the ECD formula $\mathbf{A_2Manif}$ corresponds to the algebraic formula $/\!/\mathbf{A_2.Manif}$.[9] Nevertheless, there are several cases where the two encodings greatly diverge, which we think of as evidence of formal problems posed by the ECD encoding, problems that can be solved easily in the algebraic encoding.

As opposed to the explicit encoding, we think that the algebraic and ECD encodings are suited for lexicographers. Algebraic formulas are not

[6] The fusion operator does not exist as such in the ECD encoding, which only uses the // symbol as a "mark" on a value indicating that it is "fused": **Magn**(*rain*) = *//downpour*. We believe that this notation, while convenient when fused and non fused values have to be listed together, hides the fact that a fused value does not bear the same relation with the keyword as a non fused value.

[7] This does not apply to modifying LFs such as **Magn**, whose case cannot be dealt with here due to space constraints.

[8] Because **f**(L) is a collocate of L and **g**∘**f**(L) a semantic derivative of **f**(L), **g**∘**f**(L) could itself be a collocate of L. But it could be a semantic derivative of L as well. This shows that even in such cases where composition and product are very similar, they are not equivalent operations.

[9] Note also that two distinct algebraic formulas can correspond to the same LF. For instance:
$/\!/\mathbf{A_1.Func_2} = /\!/\mathbf{A_2.Oper_1} = \mathbf{A_2}$.

dissimilar to expressions in natural language. A formula such as **Caus.Non.Manif**(L), for instance, can be very directly translated into pseudo-English as *to cause the non-manifestation of L* and, therefore, can be used as pseudo-paraphrase for the value itself in English sentences. For this reason, the algebraic encoding is a metalanguage with which the lexicographer and the linguist can "think." However, we propose this metalanguage to be defined on top of the explicit encoding, and not the other way round, in order to provide it with steadfast formal foundations. The change from the algebraic to the explicit encoding is trivial: it consists in replacing the simple LFs by their explicit definition and to compute the operations.

Furthermore, we believe that the translation of algebraic formulas into expressions in "controlled" English, French or other natural languages could be performed automatically. For the purpose of our experimentation with COLÈRE we manually produced these translations, some of which are listed below. Note that the translation procedure takes as parameter the general semantic value (what we term the *semantic label*) of the keyword. Thus, the sample translations that follow are valid for nouns of feeling only and the reader may replace # with *feeling* in reading the proposed translations:

Oper$_1$ ≡ [*X*] *experiences* #; **Oper$_2$** ≡ [*Y*] *is the target of* #; **Oper$_3$** ≡ [*Z*] *is the reason for* #; **Func$_0$** ≡ [#] *takes place*; **Func$_1$** ≡ [#] *is in X*; **Func$_2$** ≡ [#] *is targeting Y*; *//***A$_1$.f** ≡ *who* **f**; *//***A$_2$.f** ≡ *whom* **f**.

For lack of space, we will not elaborate further on the problem of bridging the gap between different modes of encoding. Suffice it to say that this problem has to be carefully addressed, especially in contexts where one wishes to popularize the concept of lexical function (for language learning, layman dictionaries, etc.).

8 Conclusions

Our main purpose in developing an explicit and an algebraic encoding for LF relations was to make available formalisms that would be computationally tractable. Such formalisms should be suitable for applications such as MT and text generation, as well as for the maintenance and the development of lexical databases.

Our explicit encoding is suitable for these tasks as it meets the following three requirements: it is entirely defined in terms of its syntax and semantics; it allows us to express all information that seems relevant to the processing of lexical databases; it allows for the definition of formal operations such as product and fusion and can be connected to other encodings (algebraic, ECD, LAF) more suitable for human reasoning.

References

Fontenelle, T. 1997. *Turning a Bilingual Dictionary into a Lexical-Semantic Database*. Niemeyer, Tübingen.

Iordanskaja, L., M. Kim and A. Polguère. 1996. Some Procedural Problems in the Implementation of Lexical Functions for Text Generation. In (Wanner, 1996): 279-297.

Mel'čuk, I. A. 1996. Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In (Wanner, 1996): 37-102.

Mel'čuk, I. A. et al. 1984, 1988, 1992, 1999. *Dictionnaire explicatif et combinatoire du français contemporain: Recherches lexico-sémantiques I*, *II*, *III*, *IV*. Presses de l'Université de Montréal, Montreal.

Mel'čuk, I. A. and L. Wanner. 2001. Toward a lexicographic Approach to Lexical Transfer in Machine Translation (Illustrated by the German-Russian language Pair). To appear in *Machine Translation*.

Mel'čuk, I. A. and A. K. Zholkovsky. 1984. *Explanatory Combinatorial Dictionary of Modern Russian*. Wiener Slawistischer Almanach, Vienna.

Polguère, A. 2000a. Towards a theoretically-motivated general public dictionary of semantic derivations and collocations for French. *Proceedings of EURALEX'2000*, Stuttgart: 517-527.

Polguère, A. 2000b. A "natural" lexicalization model for language generation. *Proceedings of the Fourth Symposium on Natural Language Processing* (SNLP'2000), Chiangmai: 37-50.

Wanner, L. (ed.). 1996. *Lexical Functions in Lexicography and Natural Language Processing*. Benjamins, Amsterdam/Philadelphia.

Žolkovskij, A. K. and I. A. Mel'čuk. 1965. O vozmožnom metode i instrumentax semantičeskogo sinteza. *Naučno-texničeskaja informacija*, 5: 23-28.