The Unit Graphs Framework: A graph-based Knowledge Representation Formalism designed for the Meaning-Text Theory

Maxime Lefrançois, Fabien Gandon

Wimmics, Inria, I3S, CNRS, UNSA
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis, France
Maxime.Lefrancois@inria.fr, Fabien.Gandon@inria.fr

Abstract

In this paper we are interested in the choice of a graph-based knowledge representation formalism that would allow for the representation, manipulation, query, and reasoning over linguistic knowledge of the Explanatory Combinatorial Dictionary of the Meaning-Text Theory (MTT). We show that neither the semantic web formalisms nor the Conceptual Graphs Formalism are suitable for this task, and we justify the introduction of the new Unit Graphs framework. We then detail the core of this formalism which is a hierarchy of unit types driven by their actantial structure. Finally we define the new deep semantic representation level for the MTT, where the specialization of actantial structures of deep semantic unit types may correspond to a specialization of conveyed meanings.

Keywords

Linguistic Knowledge Representation, Meaning-Text Theory, Unit Graphs, Explanatory and Combinatorial Dictionary.

1 Introduction

In this paper we are interested in the choice of a graph-based Knowledge Representation (KR) formalism that would allow for the represention, manipulation, query, and reasoning over linguistic knowledge of the Explanatory Combinatorial Dictionary (ECD), which is the lexicon at the core of the Meaning-Text Theory (MTT) (c.f. for instance Mel'čuk, 2006). We envision two application scenarios of such a formalization:

- In a ECD lexicographic edition oriented project, we could enable the semi-automation of some of the lexicographers tasks. For instance, we could check that a set of constraints is satisfied, or we could suggest preliminary drafts of articles (e.g., lexical function key-value pairs, lexicographic definition sketches, government pattern).

- We could propose a syntax, which is a formal language based on knowledge engineering standards. Like WordNet today, the linguistic knowledge written with that syntax could be published to the web of linked data[1]. This would support their use as a highly structured lexical resource by consumers of the linked data cloud.

Most past or current projects that aimed at implementing the ECD did so in a lexicographic perspective. One important example is the RELIEF project (Lux-Pogodalla and Polguère, 2011), which aims at representing a lexical system graph named RLF (Polguère, 2009) where lexical units are interlinked by paradigmatic and syntagmatic links of lexical functions (Mel'čuk, 1996). In the RELIEF project, the description of Lexical Functions is based on a formalization proposed by Kahane and Polguère (2001). Moreover, lexicographic definitions start to be partially formalized using the markup type that has been developed in the Definiens project (Barque and Polguère, 2008; Barque et al., 2010), which aims at formalizing lexicographic definitions with genus and specific differences for the TLFi[2].

One exception is the proprietary linguistic processor ETAP-3 that implements a variety of ECD for Natural Language Processing (Apresian et al., 2003; Boguslavsky et al., 2004). Linguistic knowledge are asserted, and linguistic and grammatical rules are directly formalized in first order logic.

Adding to these formalization works, our goal is to propose a formalization from a knowledge engineering perspective, compatible with standard KR formalisms. The term *formalization* here means not only *make non-ambiguous*, but also *make operational*, i.e., *such that it supports logical operations* (e.g., knowledge manipulation, query, reasoning). We thus adopt a knowledge engineering approach applied to the domain of the MTT.

In this paper we first justify the introduction of the new Unit Graphs formalism (§2), we then detail the conjunctive unit types hierarchy (§3) at the core of this framework, and we finally draw some important implications for the MTT (§4).

2 Choice of a Knowledge Representation Formalism

At first sight, two existing KR formalisms seem interesting for the MTT. Semantic web formalisms (RDF/S, OWL, SPARQL), because the linked data is built on them, and Conceptual Graphs (CGs) formalism (Sowa, 1984; Chein and Mugnier, 2008), as we are to lead logic reasoning on graphs. Both formalisms are based on directed labelled graph structures, and some research has been done towards using them to represent dependency structures and knowledge of the lexicon (OWL in (Lefrançois and Gandon, 2011a; Boguslavsky, 2011), CGs at the conceptual level in (Bohnet and Wanner, 2010)).

Let us first recall that for a specific Lexical Unit L, Mel'čuk (2004, p.5) distinguishes considering L in language (i.e., in the lexicon), or in speech (i.e., in an utterance). KR formalisms also do this distinction using types. Objects of the represented domain are named instances (or objects, or individuals), and are typed (or classified).

[1] The web of data is a W3C initiative, highly active today, `http://linkeddata.org`

[2] Trésor de la Langue Française informatisé, `http://atilf.atilf.fr`

2.1 Semantic Web Formalisms

There is a world wide deployment of the semantic web formalisms, and the RDF[3] syntax is the standard for structured data exchange over the web of linked data. The expressivity of RDF would be sufficient to represent the knowledge of the ECD. Yet, the semantics of RDF, in the logical sense, is limited to that of oriented labelled multi-graphs, and we wish also to enable the manipulation and reasoning over linguistic knowledge of the ECD. We thus need to introduce more semantics with RDFS[4] or OWL[5], while keeping the expressivity as low as possible to keep good computational properties. Yet RDFS and OWL only support binary relations, which is not the case of most valency-based predicates. One would need to use reification of n-ary relations[6], but then no semantics is attributed to such relations.

The ULiS project (Lefrançois and Gandon, 2011a,b) nevertheless proposed an architecture to enable such semantics: each lexical unit supports the projection of its lexicographic definition over itself. Lefrançois (2013) proved that this solution leads to an overwhelming computational complexity, i.e., the undecidable first order logic.

One alternative to represent lexicographic definitions of lexical units would be to use two reciprocal CONSTRUCT SPARQL[7] rules. But we then face the problem of rule languages and their compatibility with OWL (c.f., Krisnadhi et al., 2011), that led to no consensus nor standard today. These different problems led us to consider another formalism to represent knowledge of the ECD. We nevertheless want to be able to export these knowledge in RDF to exchange them over the web of linked data.

2.2 The Conceptual Graphs Formalism

The Conceptual Graphs (CGs) formalism (Sowa, 1984; Chein and Mugnier, 2008) has many similarities with the MTT. In their basic version, CGs represent typed instances interconnected by typed n-ary relations. Actually, the main goal of Sowa was natural language processing, and he originally inspired from the same works than MTT founders: Tesnière (1959). Sowa (1989) early suggested to introduce type definition of concepts and relations that do look similar to lexical units definitions in the ECD, and later on, Leclère (1998) also worked on the possibility to reason with type and concept definitions. One more asset of CGs is the fact that there are transformations between CGs and RDF/S (c.f., Corby et al., 2000; Baget et al., 2010). One could use these transformations to rewrite CGs to RDF for publication over the web of linked data. Moreover, one could adapt the architecture described in the ULiS (Lefrançois and Gandon, 2011a,b) project to CGs. Yet it is also not natural to represent the knowledge of the ECD using the CGs. Here are two reasons for that:

- A semantic unit may be represented as a concept type as it is instantiated in utterance semantic representations. On the other hand, if the associated lexical unit is predicative and has Semantic Actant Slots (SemASlots), then the semantic unit may dually be represented as a n-ary relation, so that its instances link other semantic units. The CGs don't offer a natural representation of this duality. In fact, in CGs, one must alternate concepts

[3] RDF - Resource Description Framework, `http://w3.org/RDF/`

[4] RDFS - RDF Schema, `http://www.w3.org/TR/rdf-schema/`

[5] OWL - Web Ontology Language, `http://www.w3.org/TR/owl2-overview/`

[6] N-ary relations on the Semantic Web, `http://www.w3.org/TR/swbp-n-aryRelations`

[7] SPARQL, `http://www.w3.org/TR/sparql11-overview/`

and relations, and a semantic representation of an utterance such as the one in figure 1 can't be directly represented by a CG.
- SemASlots of a lexical unit may differ from those of the lexical unit from which its sense derives[8]. Yet in the CGs, the inheritance mechanism of relation types, that models the fact that *a relation type is more specific than another*, is constrained so that two relations with different arities must be incomparable. One thus cannot use this natural inheritance mechanism to model the specialization of senses.

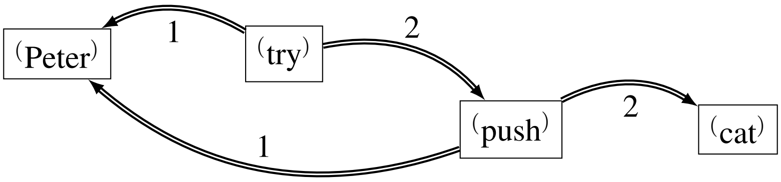


Figure 1: Illustration of the duality concept/relation of semantic units in the MTT, semantic representation of utterance *Peter tries to push the cat*.

2.3 The new Unit Graphs Formalism

To sum up, neither the semantic web formalisms nor the CGs formalism allow for a natural representation of a hierarchy of unit types that may have actant slots, which is the basic knowledge of the ECD. As the CGs formalism is the closest from the MTT, we decide to use it as a starting point for designing a new graph-based formalism adapted to the representation of the knowledge of the ECD. We will also define transformations to the RDF syntax for sharing knowledge and publishing over the web of data. As we are to represent linguistic units of different nature (e.g., semantic units, lexical units, grammatical units, words), we choose to use the term *unit* in a generic manner and name the result of this adaptation *Unit Graphs (UGs) framework*.

3 The Unit Types Hierarchy

In this section we study how we shall revisit the CGs formalism so as to make it adapted to represent a hierarchy of unit types that may have actant slots. First of all, in the Unit Graphs (UGs) mathematical framework, the objects of the represented domain are named *units*, and are typed. Parallel with existing KR formalisms and Mel'čuk (2004, p.5), we thus establish a distinction between:

- Unit types (e.g., semantic unit type, lexical unit type), described in the ECD;
- Units (e.g., semantic unit, lexical unit), represented in the Unit Graphs (UGs).

Unit types will specify through actant slots how their instances (i.e., units) shall be linked to other units in a UG. Unit types and their actantial structure are described in a structure called hierarchy and denoted $\mathcal{T}$.

[8] For instance, semantic unit ⁽rain⁾ is more specific than ⁽fall⁾ but the meaning of *what falls* and *where it falls from* is fixed to ⁽water drops⁾ and ⁽sky/cloud⁾ (Mel'čuk, 2004, p.14).

Definition 1. A hierarchy of unit types is a tuple $\mathcal{T} \overset{\text{def}}{=} (T_D, \boldsymbol{S}_{\mathcal{T}}, \boldsymbol{\gamma}, \boldsymbol{\gamma_1}, \boldsymbol{\gamma_0}, C_A, \{\varsigma_t\}_{t\in\boldsymbol{T}}, \perp^{\sqcap}_{A})$ that enables to construct a coherent pre-ordered set of unit types with an actantial structure, i.e., actant slots that may be obligatory, optional or prohibited. Actant slots are signed, their signatures characterise the type of units that fill these slots.

3.1 Primitive Unit Types and Actant Slots

First, $\mathcal{T}$ contains a set of declared *Primitive Unit Types (PUTs)* denoted T_D. This set contains linguistic PUTs of different nature (e.g., semantic, lexical, grammatical). So that Actant Slots (ASlots) are named, $\mathcal{T}$ contains a set of binary relation symbols called *Actant Symbols (ASymbols)*, and denoted $\boldsymbol{S}_{\mathcal{T}}$. $\boldsymbol{S}_{\mathcal{T}}$ contains numbers for the semantic unit types, and other classical symbols for the other levels under consideration (e.g, roman numerals **I** to **VI** for the MTT's Deep Syntactic level).

Then, no matter whether it is semantic, lexical or grammatical, a PUT $t \in \boldsymbol{T}$ has a set (that may be empty) of *Actant Slots (ASlots)* whose symbols are chosen in the set of ASymbols. Some ASlots may be obligatory, other optional (Mel'čuk, 2004, p.24), and we postulate that some may be prohibited too. For instance the Lexical Unit Type (LexUT) TO EAT has at least one obligatory semantic ASlot which is for the animal that eats, and an optional semantic ASlot which is for the container the animal eats in. If one specializes the meaning of TO EAT to define a new LexUT, we identify three basic cases that may happen:

- An optional ASlot may become obligatory.
- An optional ASlot may become prohibited, e.g., the container for TO GRAZE;
- A new ASlot may be introduced;

In order to represent these different types of ASlots and so that their presence in the hierarchy of Unit Types is coherent, we introduce three bijective mappings over the set of ASymbols:

- $\boldsymbol{\gamma}$ assigns to every $s \in \boldsymbol{S}_{\mathcal{T}}$ its radix[9] unit type $\boldsymbol{\gamma}(s)$ that introduces an Actant Slot (ASlot) of symbol s. We denote $\boldsymbol{\Gamma}$ the range of $\boldsymbol{\gamma}$, i.e., the set of *radices*[10].
- $\boldsymbol{\gamma_1}$ assigns to every $s \in \boldsymbol{S}_{\mathcal{T}}$ its obligat[11] unit type $\boldsymbol{\gamma_1}(s)$ that makes the ASlot of symbol s obligatory. We denote $\boldsymbol{\Gamma_1}$ the range of $\boldsymbol{\gamma_1}$, i.e., the set of *obligant*[12].
- $\boldsymbol{\gamma_0}$ assigns to every $s \in \boldsymbol{S}_{\mathcal{T}}$ its prohibet[13] unit type $\boldsymbol{\gamma_0}(s)$ that makes the ASlot of symbol s prohibited. We denote $\boldsymbol{\Gamma_0}$ the range of $\boldsymbol{\gamma_0}$, i.e., the set of *prohibent*[14].

The set of *Primitive Unit Types (PUTs)* is denoted $\boldsymbol{T}$ and defined as the disjoint union of the set of declared PUTs T_D, radices $\boldsymbol{\Gamma}$, obligant $\boldsymbol{\Gamma_1}$, prohibent $\boldsymbol{\Gamma_0}$, plus the *prime universal PUT* $\top$ and the *prime absurd PUT* $\perp$ (eq. 1).

$$\boldsymbol{T} \overset{\text{def}}{=} T_D \uplus \boldsymbol{\Gamma} \uplus \boldsymbol{\Gamma_1} \uplus \boldsymbol{\Gamma_0} \uplus \{\perp\} \uplus \{\top\} \qquad (1)$$

We then introduce an inheritance mechanism for the PUTs, in the form of a specialization pre-order[15] $\lesssim$ over the set $\boldsymbol{T}$. $t_1 \lesssim t_2$ models the fact that the PUT t_1 is more specific than

[9] radix is a latin word that means ⸢root⸣.

[10] radices is the plural for radix.

[11] obligat is the conjugated form of the latin verb obligo, 3p sing. pres., ⸢it makes mandatory⸣.

[12] obligant is the conjugated form of the latin verb obligo, 3p sing. plur., ⸢they make mandatory⸣.

[13] prohibet is the conjugated form of the latin verb prohibeo, 3p sing. pres., ⸢it makes prohibited⸣.

[14] prohibent is the conjugated form of the latin verb prohibeo, 3p sing. plur., ⸢they make prohibited⸣.

[15] A pre-order is a reflexive and transitive binary relation.

the PUT t_2. $\lesssim$ is defined as the smallest pre-order such that: i) it includes the set $C_A \subseteq \boldsymbol{T}^2$ of asserted PUTs comparisons (eq. 2), ii) $\top$ (resp. $\bot$) is maximal (resp. minimal) (eq. 3), and iii) for all ASymbol the radix is greater than the obligat and the prohibet (eq. 4).

$$(t_2, t_1) \in C_A \Rightarrow t_1 \lesssim t_2 \tag{2}$$

$$\forall t \in \boldsymbol{T}, \bot \lesssim t \lesssim \top \tag{3}$$

$$\forall s \in \boldsymbol{S}_{\mathcal{T}}, \boldsymbol{\gamma_1}(s) \lesssim \boldsymbol{\gamma}(s) \text{ and } \boldsymbol{\gamma_0}(s) \lesssim \boldsymbol{\gamma}(s) \tag{4}$$

As every ASlot has a symbol, the set of ASlots of a PUT $t \in \boldsymbol{T}$ is defined as the set of their symbols $\boldsymbol{\alpha}(t) \subseteq \boldsymbol{S}_{\mathcal{T}}$. Formally, the set $\boldsymbol{\alpha}(t)$ is the set of ASymbols whose radix is more general or equivalent to t (eq. 5), and thus every PUT that is more specific than the radix of an ASymbol $s \in \boldsymbol{S}_{\mathcal{T}}$ inherits an ASlot with symbol s. Similarly, the set of obligatory (resp. prohibited) ASlots of a PUT t is denoted $\boldsymbol{\alpha_1}(t)$ (resp. $\boldsymbol{\alpha_0}(t)$) and is defined as the set of ASymbols whose obligat (resp. prohibet) is more general or equivalent to t (eq. 6-7). The set of optional ASlots of a PUT t is denoted $\boldsymbol{\alpha_?}(t)$ and is the set of ASlots that are neither obligatory nor prohibited (eq. 8). The number of ASlots of a Primitive Unit Type (PUT) is denoted its *valency*.

$$\boldsymbol{\alpha}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S}_{\mathcal{T}} \mid t \lesssim \boldsymbol{\gamma}(s)\} \tag{5}$$

$$\boldsymbol{\alpha_1}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S}_{\mathcal{T}} \mid t \lesssim \boldsymbol{\gamma_1}(s)\} \tag{6}$$

$$\boldsymbol{\alpha_0}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S}_{\mathcal{T}} \mid t \lesssim \boldsymbol{\gamma_0}(s)\} \tag{7}$$

$$\boldsymbol{\alpha_?}(t) \stackrel{\text{def}}{=} \boldsymbol{\alpha}(t) - \boldsymbol{\alpha_1}(t) - \boldsymbol{\alpha_0}(t) \tag{8}$$

In the lexicographic definitions, the type of the potential fillers of a SemASlot is sometimes written before the name of the variable. In the unit types hierarchy, *signatures* of PUTs give means to represent this information explicitly. More generally, not any unit may fill a specific ASlot of a PUT. For instance, only semantic units may fill ASlots of a semantic unit, and only units of type ⁽animal⁾ may fill ASlot 1 of Semantic Unit Type (SemUT) ⁽to eat⁾.

Formally, the set of signatures of PUTs is denoted $\{\varsigma_t\}_{t \in \boldsymbol{T}}$ and is a set of functions from $\boldsymbol{\alpha}$ to $\boldsymbol{T}^{\cap}$. For all PUT t, ς_t is a function that associates to every aslot s of t a set of PUTs $\varsigma_t(s)$ that characterise the type of the units that may fill this slot. For instance the signature of ⁽to eat⁾ for its ASlot 1 is noted $\varsigma_{⁽\text{to eat}⁾}(1) = \{⁽\text{animal}⁾\}$. Signatures of a PUT t_1 may only be more specific than those of each of its ancestors t_2: if $t_1 \lesssim t_2$ and s is a common ASlot of t_1 and t_2, the signature of t_1 for s must be more specific or equivalent than that of t_2. For instance, the signature of ⁽savour⁾ for 1, i.e., {⁽person⁾}, is more specific than that of ⁽to eat⁾ which is {⁽animal⁾}.

The actantial structure of a PUT t is thus defined as the sets of its obligatory, prohibited and optional ASlots, and their signatures. It is inherited and possibly specialized by every descendent of t.

3.2 Hierarchy of Unit Types

A unit type may consist of several conjoint PUTs. In particular, it may be a lexical PUT and multiple grammatical PUTs, like $\{def, plur,$ ANIMAL$\}$ (⁽the animals⁾). To represent this, we introduce the set $\boldsymbol{T}^{\cap}$ of possible *Conjunctive Unit Types (CUTs)* over $\boldsymbol{T}$ as the powerset[16] of $\boldsymbol{T}$, i.e., $\boldsymbol{T}^{\cap} \overset{\text{def}}{=} 2^{\boldsymbol{T}}$. The definition of the actancial structure of PUTs is naturally extended to CUTs as follows:

$$\boldsymbol{\alpha}^{\cap}(t^{\cap}) \overset{\text{def}}{=} \textstyle\bigcup_{t \in t^{\cap}} \boldsymbol{\alpha}(t) \tag{9}$$

$$\boldsymbol{\alpha}_{\mathbf{1}}^{\cap}(t^{\cap}) \overset{\text{def}}{=} \textstyle\bigcup_{t \in t^{\cap}} \boldsymbol{\alpha}_{\mathbf{1}}(t) \tag{10}$$

$$\boldsymbol{\alpha}_{\mathbf{0}}^{\cap}(t^{\cap}) \overset{\text{def}}{=} \textstyle\bigcup_{t \in t^{\cap}} \boldsymbol{\alpha}_{\mathbf{0}}(t) \tag{11}$$

$$\boldsymbol{\alpha}_{\mathbf{?}}^{\cap}(t^{\cap}) \overset{\text{def}}{=} \boldsymbol{\alpha}^{\cap}(t^{\cap}) - \boldsymbol{\alpha}_{\mathbf{1}}^{\cap}(t^{\cap}) - \boldsymbol{\alpha}_{\mathbf{0}}^{\cap}(t^{\cap}) \tag{12}$$

$$\varsigma_{t^{\cap}}^{\cap}(s) \overset{\text{def}}{=} \textstyle\bigcup_{t \in t^{\cap} | s \in \boldsymbol{\alpha}(t)} \varsigma_t(s) \tag{13}$$

Some PUTs are incompatible. For instance, no unit may be of both grammatical unit types def and $indef$. To represent this, $\mathcal{T}$ contains a set of declared absurd CUTs, denoted $\bot_A^{\sqcap}$, with $\bot_A^{\sqcap} \subseteq \boldsymbol{T}^{\cap}$.

Finally, the pre-order $\lesssim$ over $\boldsymbol{T}$ is extended to a pre-order $\overset{\cap}{\lesssim}$ over $\boldsymbol{T}^{\cap}$ (c.f., Lefrançois and Gandon, 2013). $\overset{\cap}{\lesssim}$ is computed as the smallest pre-order such that: i) it contains the natural extension of a pre-order over a set to a pre-order over its powerset (eq. 14), ii) $\top$ and $\varnothing$ are both maximal elements (eq. 15), ii) every CUT declared absurd is minimal (eq. 16), iii) the conjunction of $\boldsymbol{\gamma}_{\mathbf{1}}(s)$ and $\boldsymbol{\gamma}_{\mathbf{0}}(s)$ is minimal for all $s \in \boldsymbol{S}_{\mathcal{T}}$ (eq. 17), and iv) if a CUT has a signature that is minimal, then it is minimal (eq. 18). The bottom of the pre-ordered set $\boldsymbol{T}^{\cap}$is the set of absurd CUTs, i.e., the unit types that can not be instantiated.

$$\forall t_2 \in t_2^{\cap}, \exists t_1 \in t_1^{\cap}, t_1 \lesssim t_2 \Rightarrow t_1^{\cap} \overset{\cap}{\lesssim} t_2^{\cap} \tag{14}$$

$$\varnothing \overset{\cap}{\lesssim} \{\top\} \tag{15}$$

$$\forall t^{\cap} \in \bot_A^{\sqcap}, t^{\cap} \overset{\cap}{\lesssim} \{\bot\} \tag{16}$$

$$\forall s \in \boldsymbol{S}_{\mathcal{T}}, \{\boldsymbol{\gamma}_{\mathbf{1}}(s), \boldsymbol{\gamma}_{\mathbf{0}}(s)\} \overset{\cap}{\lesssim} \{\bot\} \tag{17}$$

$$\exists s \in \boldsymbol{\alpha}^{\cap}(t^{\cap}), \varsigma_{t^{\cap}}^{\cap}(s) \overset{\cap}{\lesssim} \{\bot\} \Rightarrow t^{\cap} \overset{\cap}{\lesssim} \{\bot\} \tag{18}$$

Lefrançois and Gandon (2013) proved that in the hierarchy of unit types, if $t_1^{\cap} \overset{\cap}{\lesssim} t_2^{\cap}$ then the actantial structure of $t_1^{\cap}$ is more specific than that of $t_2^{\cap}$, except for some degenerated cases (i.e., the void unit type $\varnothing$, and the absurd unit types). Thus as one goes down the hierarchy of unit types, an ASlot with symbol s is introduced by the radix $\{\boldsymbol{\gamma}(s)\}$ and first defines an optional ASlot for any unit type $t^{\cap}$ more specific than $\{\boldsymbol{\gamma}(s)\}$, as long as $t^{\cap}$ is not more specific than the obligat $\{\boldsymbol{\gamma}_{\mathbf{1}}(s)\}$ (resp. the prohibet $\{\boldsymbol{\gamma}_{\mathbf{0}}(s)\}$) of s. If that happens, the ASlot becomes obligatory (resp. prohibited). Moreover, the signature of an ASlot may only be more specific than that of its parents.

Any unit type that possesses ASlots thus represents a linguistic predicate as defined by Mel'čuk (2004, p.8), and unit nodes having that type must (resp1. may, resp2 must not) be linked to its obligatory (resp1. optional, resp2. prohibited) actants in a UG.

[16] The powerset of X is the set of all subsets of X: 2^X

4 Implications for the Different Levels of Representation

As semantic ASymbols are numbers, the pre-order over semantic unit types cannot represent a specialization of meanings. Let us give an example to justify this.

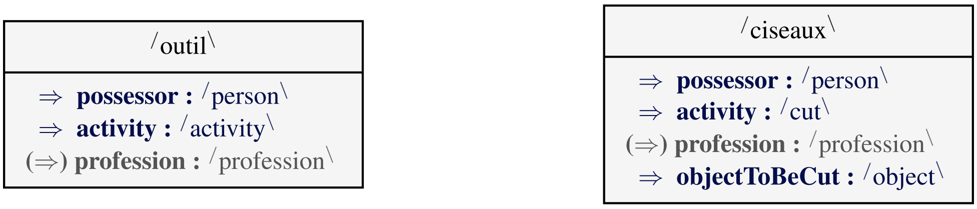
The french semantic unit type ⁽outil⁾ (⁽tool⁾) has an ASlot 1 that corresponds to the person X that uses the tool, and a split ASlot 2 that corresponds either to the activity Y_1 or to the profession Y_2 for which the tool is designed[17]. Now ⁽ciseaux⁾ (⁽scissors⁾) has a stricter meaning than ⁽outil⁾, and also a ASlot 2 that now corresponds to the object Y that it is intended to cut. Thus ⁽ciseaux⁾ cannot be lower than ⁽outil⁾ in the hierarchy of semantic unit types because this would imply that an object is some kind of activity or profession.

We hence propose to introduce a deeper level of representation where one may describe meanings: the *deep semantic level*. We thus establish a distinction between deep and surface semantic unit types. Let us precise their definition and their actantial structure.

Definition 2 (Surface Semantic Unit Types and their ASlots)**.** To every meaningful Lexical Unit Type (LexUT) L is associated a *Surface Semantic Unit Type* that is denoted ⁽L⁾. The ASlots of ⁽L⁾ correspond to the SemASlots of L as defined in (Mel'čuk, 2004, p.39) and are numbered.

Definition 3 (Deep Semantic Unit Types and their ASlots)**.** To every meaningful LexUT L is associated a *Deep Semantic Unit Type (DSemUT)* that is denoted /L\. The set of deep semantic ASymbols are semantic roles (e.g., $agent$, $experiencer$, $object$). The set of ASlots of a DSemUT corresponds to obligatory or optional participants of the linguistic situation denoted by L that are: a) SemASlots of L, or b) SemASlots of a LexUT whose meaning is more generic than that of L.

For instance figure 2 illustrates the actantial structure of /outil\ (⁽An artefact designed for a person X to use it for an activity Y_1 (or for a profession Y_2)⁾) and /ciseaux\, which derives from /tool\. /tool\ has two obligatory actant slots $possessor$ and $activity$, and an optional ASlot $profession$. /ciseaux\ inherits the ASlots of /tool\, and restricts the signature of $activity$ to be /cut\. As CISEAUX also introduces a SemASlot which is the object to be cut, /ciseaux\ also introduces a new ASlot $objectToBeCut$.


/outil\

⇒ **possessor :** /person\
⇒ **activity :** /activity\
(⇒) **profession :** /profession\

/ciseaux\

⇒ **possessor :** /person\
⇒ **activity :** /cut\
(⇒) **profession :** /profession\
⇒ **objectToBeCut :** /object\

Figure 2: Actantial structures of /outil\ and /ciseaux\.

One may need to introduce a new ASymbol every time a SemASlot that conveys a new meaning is introduced. The set of semantic roles thus cannot be bound to a small set of universal semantic roles.

5 Conclusion

In this paper we showed that both semantic web and conceptual graphs formalisms are not adapted to represent knowledge of the ECD while ensuring good computational properties. We

[17] See (Mel'čuk, 2004, p.43) for more details on split ASlots.

hence justified the introduction of the new Unit Graphs (UGs) graph-based knowledge representation formalism.

The Unit Types hierarchy is the core structure of the UGs. It consists in a minimal set of mathematical objects that allows to construct a pre-ordered set of unit types with actantial structures. The actantial structure of a unit type is composed of actant slots that may be optional, obligatory, or signed, and that are signed. Moreover, a unit type inherits and possibly specialize the actantial structure of its parents.

The so-defined Unit Types hierarchy has strong implications for the MTT. In fact, the preorder over unit types can not correspond to a meaning-specialization relation for semantic unit types as defined in the MTT. We therefore introduced a deep-semantic representation level, and defined the deep and surface semantic unit types and their actantial structure.

Current directions of work include the definition of UGs and their semantics, in the logical sense; the definition of lexicographic definitions, that must be at the deep semantic level; and the representation of the semantic derivation part of lexical functions, which constrains the actantial structure and the definitions of deep semantic units.

Bibliography

Apresian, J., Boguslavsky, I., Iomdin, L., Lazursky, A., Sannikov, V., Sizov, V., and Tsinman, L. (2003). ETAP-3 linguistic processor: A full-fledged NLP implementation of the MTT. In *First International Conference on Meaning–Text Theory (MTT'2003)*, pages 279–288.

Baget, J.-F., Croitoru, M., Leclère, M., Mugnier, M.-L., and Gutierrez, A. (2010). Translations between RDF(S) and conceptual graphs. *ICCS'10: 18th International Conference on Conceptual Structures - From Information to Intelligence*, pages 28–41.

Barque, L., Nasr, A., and Polguère, A. (2010). From the Definitions of the 'Trésor de la Langue Française' To a Semantic Database of the French Language. In Fryske Akademy, editor, *Proceedings of the XIV Euralex International Congress*, Fryske Akademy, pages 245–252, Leeuwarden, Pays-Bas. Anne Dykstra et Tanneke Schoonheim, dir.

Barque, L. and Polguère, A. (2008). Enrichissement formel des définitions du Trésor de la Langue Française informatisé (TLFi) dans une perspective lexicographique. *Lexique*, 22.

Boguslavsky, I. (2011). Semantic Analysis Based on Linguistic and Ontological Resources. In Boguslavsky, I. and Wanner, L., editors, *Proceedings of the 5th International Conference on Meaning-Text Theory (MTT'2011)*, pages 25–36, Barcelona, Spain. INALCO.

Boguslavsky, I., Iomdin, L., and Sizov, V. (2004). Multilinguality in ETAP-3: reuse of lexical resources. In Sérasset, G., editor, *Proc. COLING 2004 Multilingual Linguistic Ressources*, pages 1–8, Geneva, Switzerland. COLING.

Bohnet, B. and Wanner, L. (2010). Open source graph transducer interpreter and grammar development environment. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 19–21, Valletta, Malta. European Language Resources Association (ELRA).

Chein, M. and Mugnier, M.-L. (2008). *Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer-Verlag New York Incorporated.

Corby, O., Dieng, R., and Hébert, C. (2000). A Conceptual Graph Model for {W3C} Resource

Description Framework. In Ganter, B. and Mineau, G. W., editors, *Conceptual Structures: Logical, Linguistic, and Computational Issues*, number 1867 in Lecture Notes in Computer Science, pages 468–482. Springer Berlin Heidelberg.

Kahane, S. and Polguère, A. (2001). Formal foundation of lexical functions. In *Proceedings of ACL/EACL 2001 Workshop on Collocation*, pages 8–15.

Krisnadhi, A., Maier, F., and Hitzler, P. (2011). OWL and Rules. *Reasoning Web. Semantic Technologies for the Web of Data*, pages 382–415.

Leclère, M. (1998). Raisonner avec des définitions de types dans le modèle des graphes conceptuels. *Revue d'intelligence artificielle*, 12(2):243–278.

Lefrançois, M. (2013). Représentation des connaissances du DEC: Concepts fondamentaux du formalisme des Graphes d'Unités. In *Actes de la 15ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'2013)*, pages 164–177, Les Sables d'Olonne, France.

Lefrançois, M. and Gandon, F. (2011a). ILexicOn: Toward an ECD-Compliant Interlingual Lexical Ontology Described with Semantic Web Formalisms. In Boguslavsky, I. and Wanner, L., editors, *Proceedings of the 5th International Conference on Meaning-Text Theory (MTT'2011)*, pages 155–164, Barcelona, Spain. INALCO.

Lefrançois, M. and Gandon, F. (2011b). ULiS: An Expert System on Linguistics to Support Multilingual Management of Interlingual Semantic Web Knowledge bases. In Elena Montiel-Ponsoda John McCrae, P. B. and Cimiano, P., editors, *Proc. of the 9th International Conference on Terminology and Artificial Intelligence (TIA 2011)*, volume 775, pages 108–114, Paris, France. CEUR-WS.

Lefrançois, M. and Gandon, F. (2013). The Unit Graphs Mathematical Framework. Research Report RR-8212, Inria.

Lux-Pogodalla, V. and Polguère, A. (2011). Construction of a French Lexical Network: Methodological Issues. In *Proceedings of the International Workshop on Lexical Resources*, Ljubljana.

Mel'čuk, I. (1996). Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In Wanner, L., editor, *Lexical Functions in Lexicography and Natural Language Processing*, pages 37–102. Benjamins Academic Publishers, Amsterdam/Philadelphia.

Mel'čuk, I. (2004). Actants in Semantics and Syntax I: Actants in Semantics. *Linguistics*, 42(1):247–291.

Mel'čuk, I. (2006). Explanatory combinatorial dictionary. *Open Problems in Linguistics and Lexicography*, pages 225–355.

Polguère, A. (2009). Lexical systems: graph models of natural language lexicons. *Language resources and evaluation*, 43(1):41–55.

Sowa, J. (1989). Using a lexicon of canonical graphs in a semantic interpreter. In *Relational models of the lexicon*, pages 113–137. Cambridge University Press New York, NY, USA.

Sowa, J. F. (1984). *Conceptual structures: information processing in mind and machine*. System programming series. Addison-Wesley Pub., Reading, MA.

Tesnière, L. (1959). *Éléments de syntaxe structurale*. C. Klincksieck (Colombes, Impr. ITE).