Représentation des connaissances du DEC: Concepts fondamentaux du formalisme des Graphes d'Unités

Maxime Lefrançois
WIMMICS, Inria, 2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex
maxime.lefrancois@inria.fr

RÉSUMÉ

Dans cet article nous nous intéressons au choix d'un formalisme de représentation des connaissances qui nous permette de représenter, manipuler, interroger et raisonner sur des connaissances linguistiques du Dictionnaire Explicatif et Combinatoire (DEC) de la Théorie Sens-Texte. Nous montrons que ni les formalismes du web sémantique ni le formalisme des Graphes conceptuels n'est adapté pour cela, et justifions l'introduction d'un nouveau formalisme dit des Graphes d'Unités. Nous introduisons la hiérarchie des Types d'Unités au cœur du formalisme, et présentons les Graphes d'Unités ainsi que la manière dont on peut les utiliser pour représenter certains aspects du DEC.

ABSTRACT

ECD Knowledge Representation : Fundamental Concepts of the Unit Graphs Framework

In this paper we are interested in the choice of a knowledge representation formalism that enables the representation, manipulation, query, and reasoning over linguistic knowledge of the Explanatory and Combinatorial Dictionary (ECD) of the Meaning-Text Theory. We show that neither the semantic web formalisms nor the Conceptual Graphs Formalism suit our needs, and justify the introduction of a new formalism denoted Unit Graphs. We introduce the core of this formalism which is the Unit Types hierarchy, and present Unit Graphs and how one may use them to represent aspects of the ECD.

MOTS-CLÉS : Représentation de Connaissances Linguistiques, Théorie Sens-Texte, Graphes d'Unités, Dictionnaire Explicatif et Combinatoire.

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

1 Introduction

Dans cet article nous nous intéressons au choix d'un formalisme de représentation des connaissances qui nous permette de représenter, manipuler, interroger et raisonner sur des connaissances linguistiques du Dictionnaire Explicatif et Combinatoire (DEC), qui est le lexique au cœur du sujet d'étude de la Théorie Sens-Texte (TST) (c.f. par exemple Mel'čuk et Arbatchewsky-Jumarie, 1999; Mel'čuk, 2006). Nous envisageons deux scénarios de valorisation d'une telle formalisation :

– Dans un projet orienté vers l'édition lexicographique du DEC, il serait possible de semi-automatiser le travail des lexicographes par exemple en vérifiant qu'un ensemble de contraintes est satisfait, ou en leur suggérant des ébauches d'articles (e.g., liens de fonctions lexicales, ébauche de définition lexicographique, tableaux de régime).
– En proposant une syntaxe basée sur les standards de l'ingénierie des connaissances, les connaissances linguistiques ainsi représentées de manière structurée pourraient être publiées sur le web de données [1] comme l'est aujourd'hui WordNet. Ceci encouragerait leur utilisation comme ressource lexicale hautement structurée par les consommateurs de données du nuage du web de données.

La plupart des projets passés ou présents qui ont consisté en l'informatisation du DEC sont orientés vers l'édition lexicographique. Nous citerons en exemple le projet RELIEF (Lux-Pogodalla et Polguère, 2011) qui vise à représenter un graphe de type système lexical dénommé Réseau Lexical du Français (RLF) (Polguère, 2009) tissé par les liens paradigmatiques et syntagmatiques de fonctions lexicales (e.g., Mel'čuk, 1996). Des travaux de formalisation de certains aspects du DEC ont précédé le projet RELIEF. Citons les travaux de Kahane et Polguère (2001) pour les fonctions lexicales, ainsi que le projet Definiens (Barque et Polguère, 2008) de formalisation des définitions lexicographiques avec genre prochain et différences spécifiques pour le TLFi [2].

En complément de ces travaux de formalisation, notre objectif est de proposer une formalisation au sens de l'ingénierie des connaissances, compatible avec des formalismes standards. Le terme *formalisation* signifie ici non seulement *rendre non-ambigu*, mais également *rendre opérationnel*, i.e., *rendre adapté aux opérations logiques ou rationnelles* (e.g., la manipulation, l'interrogation, et le raisonnement des connaissances). Nous adoptons donc une approche d'ingénierie des connaissances appliquée au domaine de la TST, et la question de recherche de cet article est : *Quel formalisme de représentation des connaissances serait adapté pour représenter les connaissances du DEC ?*

Nous nous intéressons à deux familles de formalismes de représentation des connaissances existant :

– les formalismes du web sémantique, car le web de données est construit dessus ;
– le formalisme des Graphes Conceptuels (GC) (Sowa, 1984; Chein et Mugnier, 2008), puisqu'on sera amenés à faire des raisonnements logiques sur des graphes.

Notre question de recherche se décompose alors en deux sous-questions que nous abordons dans cet article :

– Ces deux formalismes de représentation des connaissances sont-ils adaptés pour représenter les connaissances du DEC ?
– Le cas échéant, comment devons-nous en revoir les bases afin d'en dériver un nouveau formalisme de représentation des connaissances qui soit adapté ?

1. Le web de données est une initiative du W3C en pleine effervescence actuellement, http://linkeddata.org
2. Trésor de la Langue Française informatisé, http://atilf.atilf.fr

La suite de l'article est organisée de la manière suivante. Nous verrons dans un premier temps que ni les formalismes du web sémantique ni les GC ne sont adaptés pour représenter les connaissances du DEC, et nous étayerons le choix suivant : *Nous modifions les bases du formalisme des GC, tout en gardant en tête l'idée d'utiliser les formalismes du web sémantique comme syntaxe pour l'échange des connaissances et pour la publication sur le web de données* (§2). Puisque nous représenterons des unités linguistiques de différentes natures (e.g., sémantème, lexie, grammème, mot-forme), nous choisissons d'utiliser le terme *unité* d'une manière générique et nommons le résultat de cette adaptation *formalisme mathématique des Graphes d'Unités (GU)*. Nous introduirons donc les types unités (§3) puis les graphes d'unités *per se* et leur utilité pour représenter des concepts plus avancés de la TST (§4).

Nous attirons l'attention du lecteur sur le fait que cet article introduit l'élaboration du formalisme mathématique des GU qui fait l'objet d'un rapport de recherche (Lefrançois, 2013). Nous l'invitons à s'y référer pour toute précision définitoire et mathématique.

2 Motivations pour introduire un nouveau formalisme de représentation des connaissances

Les formalismes de représentation de connaissances utilisent abondamment la notion de typage. Les objets du domaine représenté sont nommés instances (ou objets ou individus), et sont typés (ou classifiés). Ils sont liés entre eux par des relations qui sont elles-mêmes typées. Dans cette section nous répondons à la question suivante : *En quoi les formalismes du web sémantique et le formalisme des GC ne sont-ils pas directement adaptés pour représenter les connaissances du DEC ?*

2.1 Les formalismes du web sémantique

On observe un engouement mondial pour les formalismes du web sémantique, et la syntaxe RDF [3] est le standard d'échange de données structurées sur le web de données. L'expressivité de RDF serait suffisante pour représenter les connaissances du DEC. Cependant, la sémantique de RDF, au sens logique, se limite à celle des graphes orientés et étiquetés, et nous souhaitons permettre également de manipuler et de raisonner avec les connaissances linguistiques du DEC. Nous devons donc envisager d'introduire plus de sémantique à l'aide de RDFS [4] ou OWL [5], tout en limitant au maximum le niveau d'expressivité pour conserver de bonnes propriétés computationnelles. OWL introduit de la sémantique à l'aide d'axiomes [6] et de constructeurs de classes et de relations [7]. Justement le projet ULiS (Lefrançois et Gandon, 2011) envisageait une architecture de base de connaissances multilingue compatible avec la TST et basée sur OWL. Dans le projet ULiS, les axiomes et constructeurs de classe de OWL sont utilisés pour que chaque lexie supporte la projection de sa définition lexicographique sur elle-même. Nous avons identifié trois problèmes majeurs avec l'utilisation de OWL pour ce faire :

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. e.g., Sous-classe `SubClassOf( CE1 CE2 )` ; Relation fonctionnelle : `FunctionalObjectProperty( OPE)`
7. e.g., Cardinalité exacte `ObjectExactCardinality( n OPE)` ; Relation inverse `ObjectInverseOf( OPE)`

– Pour chaque définition de lexie on doit introduire autant de nouvelles relations sémantiques qu'il existe de nœuds dans le graphe de définition de la lexie. Cela impose une surcharge de relations superflues ;
– Ces relations doivent être combinées à l'aide de l'axiome de sous-relation d'une relation chaînée `SubObjectPropertyOf( ObjectPropertyChain( OPE`$_1$ `...OPE`$_n$ `) OPE)`, afin de projeter petit à petit le graphe de définition de la lexie sur elle même. Or dans OWL, l'ensemble des relations doit être *régulier* [8] pour garantir la décidabilité des problèmes de raisonnement basiques, et nous avons montré (Lefrançois, 2013) que cette régularité n'est pas assurée dans la petite ontologie donnée en exemple par Lefrançois et Gandon (2011). Cette restriction est donc trop importante pour représenter les définitions du DEC.
– Enfin, la sémantique de l'axiome de sous-relation d'une relation chaînée fait que l'inférence n'est de toute façon possible que dans une direction seulement (sous-relation, et non pas équivalence). C'est à dire que lorsqu'on est en présence du graphe de définition de la lexie, on peut inférer la présence de la lexie, mais pas le contraire.

Une alternative pour représenter les définitions d'unités lexicales serait de les représenter à l'aide de deux règles SPARQL [9] CONSTRUCT réciproques. On se rapporte alors au problème des langages de règles et de leur réconciliation avec OWL (c.f., Krisnadhi *et al.*, 2011), qui ne fait aujourd'hui l'œuvre d'aucun consensus ni standard.

Ces différents problèmes nous poussent à considérer un autre formalisme pour représenter les connaissances du DEC. Nous souhaitons néanmoins un export en RDF pour échanger les connaissances linguistiques sur le web de données.

2.2 Les Graphes Conceptuels

Le formalisme des Graphes Conceptuels (GC) (Sowa, 1984; Chein et Mugnier, 2008) présente de grandes ressemblances avec la TST. Dans leur version basique, les GC représentent des instances typées interconnectées par des relations n-aires également typées. D'ailleurs, l'objectif premier de Sowa était le traitement du langage naturel, et il s'est originellement inspiré des mêmes travaux que les fondateurs de la TST pour mettre au point le modèle des GC : les travaux de Tesnière (1959). Deux des ressemblances les plus marquantes entre les GC et la TST sont les suivantes :
– Dans les GC il est possible de définir des types de concepts et de relations à partir d'un graphe conceptuel, ce qui est très similaire aux définitions des lexies dans la TST ;
– La TST utilise intensivement des règles, en particulier pour les correspondances entre niveaux de représentation d'énoncés. Les règles et leur sémantique, au sens logique, ont été très étudiées dans la littérature des GC.

Un autre atout des GC est le fait qu'il existe des transformations entre les GC et RDF/S (c.f., Corby *et al.*, 2000; Baget *et al.*, 2010). On pourrait donc utiliser ces transformations pour réécrire les GC en RDF pour publication sur le web de données. De plus, pour en revenir au projet ULiS, on pourrait adapter l'architecture du projet ULiS aux GC.

Cependant il n'est pas non plus naturel de représenter les connaissances du DEC à l'aide des GC. Voici deux raisons à cela :
– Un sémantème est modélisable à priori comme un type de concept puisqu'il est instancié dans des représentations sémantiques d'énoncés. D'un autre côté, si la lexie associée est prédicative

8. c.f. par exemple, http://www.w3.org/TR/owl2-syntax#The_Restrictions_on_the_Axiom_Closure
9. SPARQL, http://www.w3.org/TR/sparql11-overview/

et possède des positions actancielles sémantiques, le sémantème peut de manière duale être modélisé par une relation *n*-aire de sorte que ses instances lient d'autres sémantèmes. Les GC ne permettent pas de représenter naturellement cette dualité. En effet, dans les GC on doit respecter une alternance concept/relation, et une représentation sémantique d'énoncé comme celle de la figure 1 ne peut pas être directement représentée par un GC.

– Les positions actancielles sémantiques d'une lexie peuvent différer de celles de la lexie dont son sens dérive [10] (c.f., Mel'čuk, 2004a,b). Or dans les GC, le mécanisme d'héritage des types de relations, qui modélise le fait que *un type de relation est plus spécifique qu'un autre*, est contraint de sorte que deux relations d'arité différente doivent être incomparables. On ne peut donc pas utiliser ce mécanisme naturel d'héritage pour modéliser la spécialisation des sémantèmes.

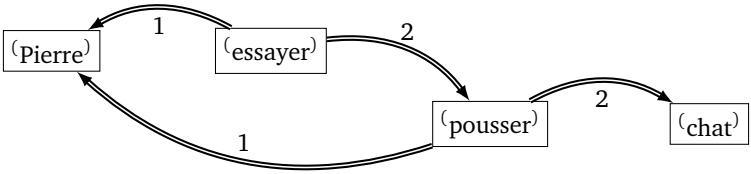

FIGURE 1 – Illustration du rôle dual concept/relation des sémantèmes dans la TST, par la représentation sémantique de *Pierre essaie de pousser le chat*.

2.3 Le nouveau formalisme des Graphes d'Unités

Pour résumer, ni les formalismes du web sémantique ni les GC ne permettent la représentation naturelle des connaissances du DEC. Le formalisme des GC étant le plus proche de la TST, nous décidons donc d'en revisiter les bases afin de le rendre compatible avec la TST.

Puisque nous représenterons des unités linguistiques de différentes nature (e.g., sémantème, lexie, grammème, mot-forme), nous choisissons d'utiliser le terme *unité* d'une manière générique et nommons le résultat de cette adaptation *formalisme mathématique des Graphes d'Unités (GU)*.

Dans un autre travail en cours, nous adaptons les transformations existantes entre les GC et RDF/S (c.f., Corby *et al.*, 2000; Baget *et al.*, 2010) afin d'utiliser les formalismes du web sémantique comme format d'échange des connaissances linguistiques sur le web de données.

Dans la suite de cet article nous apportons une réponse à la question de recherche suivante : *Comment devons-nous revoir les bases du formalisme des GC afin de le rendre adapté à la représentation des connaissances du DEC ?* Cette question se décompose en deux sous-questions :

– Quelle structure mathématique pour une hiérarchie de types d'unités pouvant avoir des positions actancielles (§3) ?
– Quel est l'équivalent des graphes conceptuels pour le formalisme des GU, et comment les utiliser pour formaliser des concepts plus avancés de la TST (§4) ?

10. Par exemple le sémantème (pluie) est plus spécifique que (tomber) mais le sens de *ce qui tombe* et *d'où ça tombe* est figé à (gouttes d'eau) et (ciel/nuage) (Mel'čuk, 2004a).

3 La Hiérarchie des Types d'Unités

Dans cette section nous abordons la question suivante : *Comment devrions-nous revoir les bases du formalisme des GC afin de le rendre adapté à la représentation d'une hiérarchie des types d'unités avec une structure actancielle ?* Tout d'abord, dans le formalisme des Graphes d'Unités (GU), les objets du domaine représenté sont nommés *unités*, et sont typés. A l'instar des formalismes de représentation de connaissances existants et de Mel'čuk (2004a), nous établissons une distinction claire entre :
– Les types d'unités (e.g., type d'unité sémantique, type d'unité lexicale), décrits dans le DEC ;
– Les unités (e.g., unité sémantique, unité lexicale), représentées dans les Graphes d'Unités (GU).
Les types d'unités vont spécifier à travers leurs positions actancielles et signatures comment leurs instances (i.e., unités) devraient être liées entre elles dans un GU. Les types d'unités et leur structure actancielle sont décrites dans une structure dénommée *hiérarchie* et notée $\mathcal{T}$.

3.1 Types d'Unités Primitifs (TPU) et Positions Actancielles

Tout d'abord, $\mathcal{T}$ contient un ensemble fini de Types Primitifs d'Unités (TPU) déclarés noté T_D. Cet ensemble contient des TPU linguistiques de différente nature (e.g., sémantique, lexicale, grammaticale). Afin de nommer les positions actancielles, on introduit un ensemble de relations binaires dénommées *Symboles d'Actants (SymbolA)*, noté $\boldsymbol{S}_{\mathcal{T}}$. $\boldsymbol{S}_{\mathcal{T}}$ contient des numéros pour la structure actancielle des types d'unités sémantiques, et d'autres symboles habituels pour les autres niveaux de représentations considérés (e.g, chiffres romains I à VI pour le niveau syntaxique profond de la TST).

Peut importe qu'il soit sémantique, lexical ou grammatical, un TPU t a un ensemble (qui peut être vide) de *Positions Actancielles (PosA)* dont les symboles sont choisis dans l'ensemble des SymbolA. Certaines PosA peuvent être obligatoires, d'autres optionnelles (Mel'čuk, 2004a), et nous postulons également que certaines PosA peuvent être interdites (i.e., désactivées en quelque sorte). Par exemple le type de lexie TO EAT (⟨manger⟩) a au moins une PosA sémantiques obligatoire qui est l'animal qui mange, et une PosA optionnelle qui est le récipient dans lequel l'animal mange. Si l'on cherche maintenant à affiner le sens de TO EAT pour définir une nouvelle lexie, nous identifions trois cas basiques qui peuvent arriver :
– Une PosA optionnelle peut devenir obligatoire.
– Une PosA optionnelle peut devenir interdite, e.g., le récipient dans TO GRAZE (⟨brouter⟩) ;
– Une nouvelle PosA (à priori optionnelle) peut être introduite ;
Pour représenter ces différents types de PosA et pour que leur présence dans la hiérarchie des types d'unités soit cohérente, on introduit trois fonctions sur l'ensemble des SymbolA :
– $\boldsymbol{\gamma}$ associe à chaque SymbolA $s \in \boldsymbol{S}_{\mathcal{T}}$ son *radix* [11] $\boldsymbol{\gamma}(s)$ qui introduit une PosA de symbole s. On note $\boldsymbol{\Gamma}$ l'ensemble d'arrivée de la fonction $\boldsymbol{\gamma}$, i.e., l'ensemble des *radices* [12].
– $\boldsymbol{\gamma_1}$ associe à chaque SymbolA s son *obligat* [13] $\boldsymbol{\gamma_1}(s)$ qui rend la PosA de symbole s obligatoire. On note $\boldsymbol{\Gamma_1}$ l'ensemble d'arrivée de la fonction $\boldsymbol{\gamma}$, i.e., l'ensemble des *obligant* [14].

11. radix est un mot latin qui signifie ⟨racine⟩.
12. radices est le pluriel de radix.
13. obligat est la forme conjuguée du verbe latin obligo, 3p sing. pres. ind., ⟨il oblige⟩.
14. obligant est la forme conjuguée du verbe latin obligo, 3p plur. pres. ind., ⟨ils obligent⟩.

– $\boldsymbol{\gamma_0}$ associe à chaque SymbolA s son *prohibet* [15] $\boldsymbol{\gamma_0}(s)$ qui rend la PosA de symbole s interdite. On note $\boldsymbol{\Gamma_0}$ l'ensemble d'arrivée de la fonction $\boldsymbol{\gamma_0}$, i.e., l'ensemble des *prohibent* [16].

L'ensemble des *Types Primitifs d'Unités (TPU)* est donc noté $\boldsymbol{T}$ et est égal à l'union disjointe de l'ensemble des TPU déclarés, des radices, des obligant et des prohibent, plus le *TPU universel principal* $\top$ et le *TPU absurde principal* $\bot$.

$$\boldsymbol{T} \stackrel{\text{def}}{=} T_D \uplus \boldsymbol{\Gamma} \uplus \boldsymbol{\Gamma_1} \uplus \boldsymbol{\Gamma_0} \uplus \{\bot\} \uplus \{\top\} \tag{1}$$

On peut alors introduire une relation de spécialisation sur l'ensemble $\boldsymbol{T}$ des TPU sous la forme d'un pré-ordre $\lesssim$. $t_1 \lesssim t_2$ modélise le fait que le TPU t_1 est plus spécifique que le TPU t_2. La relation $\lesssim$ est calculée à partir d'un ensemble de comparaisons déclarées $C_A \subseteq \boldsymbol{T}^2$, et de sorte que :

– $\top$ (resp. $\bot$) soit l'élément maximal (resp. minimal) ;

– pour chaque SymbolA l'obligat et le prohibet soit plus spécifique que le radix.

Chaque PosA ayant un symbole, l'ensemble des PosA d'un TPU $t \in \boldsymbol{T}$ est défini par l'ensemble de leurs symboles $\boldsymbol{\alpha}(t) \subseteq \boldsymbol{S_\mathscr{T}}$. Formellement, l'ensemble $\boldsymbol{\alpha}(t)$ est défini comme l'ensemble des SymbolA dont le radix est plus général ou équivalent à t, et donc tout TPU plus spécifique qu'un SymbolA $s \in \boldsymbol{S_\mathscr{T}}$ hérite d'une PosA de symbole s.

$$\boldsymbol{\alpha}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S_\mathscr{T}} \mid t \lesssim \boldsymbol{\gamma}(s)\} \tag{2}$$

De la même manière, l'ensemble des PosA obligatoires (resp. interdites) d'un TPU t est noté $\boldsymbol{\alpha_1}(t)$ (resp. $\boldsymbol{\alpha_0}(t)$) et est défini comme l'ensemble des SymbolA dont l'obligat (resp. le prohibet) est plus général ou équivalent à t.

$$\boldsymbol{\alpha_1}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S_\mathscr{T}} \mid t \lesssim \boldsymbol{\gamma_1}(s)\} \tag{3}$$

$$\boldsymbol{\alpha_0}(t) \stackrel{\text{def}}{=} \{s \in \boldsymbol{S_\mathscr{T}} \mid t \lesssim \boldsymbol{\gamma_0}(s)\} \tag{4}$$

Finalement, l'ensemble des PosA optionnelles d'un TPU t est noté $\boldsymbol{\alpha_?}(t)$ et est l'ensemble des PosA qui ne sont ni obligatoires ni interdites :

$$\boldsymbol{\alpha_?}(t) \stackrel{\text{def}}{=} \boldsymbol{\alpha}(t) - \boldsymbol{\alpha_1}(t) - \boldsymbol{\alpha_0}(t) \tag{5}$$

Ainsi en descendant la hiérarchie des types d'unités, une PosA de symbole s est introduite par le radix $\boldsymbol{\gamma}(s)$ et définit d'abord une PosA optionnelle pour tout TPU t plus spécifique que $\boldsymbol{\gamma}(s)$, tant que t n'est pas plus spécifique que l'obligat $\boldsymbol{\gamma_1}(s)$ (resp. le prohibet $\boldsymbol{\gamma_0}(s)$) de s auquel cas la PosA devient obligatoire (resp. interdite). La structure actancielle des types d'unités ainsi définie spécifie comment les unités pourront, devront, ou ne devront pas être liées entre elles dans un GU.

15. prohibet est la forme conjuguée du verbe latin prohibeo, 3p sing. pres. ind., ⟨il interdit⟩.
16. prohibent est la forme conjuguée du verbe latin prohibeo, 3p plur. pres. ind., ⟨ils interdisent⟩.

3.2 Signature d'un TPU

Dans les définitions lexicographiques de la TST, le type des unités qui prennent une PosA sémantique est parfois écrit devant le nom de la variable en question. Dans la hiérarchie des types d'unités, les *signatures* des TPU nous permettent de représenter cette information de manière explicite. Plus généralement, les unités qui prennent une certaine PosA d'un TPU doivent avoir un certain type. Par exemple, seules les unités sémantiques peuvent prendre une PosA d'une unité sémantique, et seules les unités de type $^{(}$animal$^{)}$ peuvent prendre la PosA 1 d'une unité de type $^{(}$to eat$^{)}$.

Formellement, l'ensemble des signatures des TPU est noté $\{\boldsymbol{\varsigma}_t\}_{t\in\boldsymbol{T}}$. Pour tout TPU t, $\boldsymbol{\varsigma}_t$ est une fonction qui associe à chaque PosA s de t un ensemble de TPU $\boldsymbol{\varsigma}_t(s)$ qui caractérisent le type des unités qui peuvent prendre cette position. Par exemple la signature de $^{(}$to eat$^{)}$ pour sa PosA 1 est notée $\boldsymbol{\varsigma}_{^{(}\text{to eat}^{)}}(1) = \{^{(}\text{animal}^{)}\}$.

Les signatures participent à la spécialisation de la structure actancielle des TPU, ce qui signifie que si $t_1 \lesssim t_2$ et s est une PosA commune à t_1 et t_2, alors la signature de t_1 pour s doit être plus spécifique que la signature de t_2 pour s. Par exemple, la signature de $^{(}$to sup$^{)}$ pour 1, i.e., $\{^{(}\text{personne}^{)}\}$, est plus spécifique que cette de $^{(}$to eat$^{)}$ qui est $\{^{(}\text{animal}^{)}\}$.

3.3 Hiérarchie des Types d'Unités

Une unité peut en réalité avoir une conjonction, au sens logique, de plusieurs types. En particulier, il peut s'agir d'un type de lexie et de plusieurs types d'unités grammaticales, comme $\{def, plur, \text{CHAT}\}$ pour $^{(}$les chats$^{)}$. Pour représenter ce phénomène, nous introduisons l'ensemble $\boldsymbol{T}^{\cap}$ des *Types Conjonctifs d'Unités (TCU)* possibles sur $\boldsymbol{T}$ comme étant l'ensemble des parties de $\boldsymbol{T}$:

$$\boldsymbol{T}^{\cap} \stackrel{\text{def}}{=} 2^{\boldsymbol{T}} \tag{6}$$

La conjonction des types donne un premier aperçu d'un type d'inférence. En effet, pour une unité de type $\{^{(}\text{personne}^{)}\}$, on peut augmenter son type à $\{^{(}\text{personne}^{)}, ^{(}\text{animal}^{)}\}$ qui est équivalent mais plus "explicite". Plus généralement, un TCU $t^{\cap} \in \boldsymbol{T}^{\cap}$ peut être fermé en y ajoutant tout TPU plus générique qu'au moins un de ses éléments.

Enfin, certains TCU comme $\{def, indef\}$ sont déclarés absurdes, ce qui signifie qu'aucune unité ne peut être à la fois des types def et $indef$. On notera l'ensemble des TCU déclarés absurdes $\bot_A^{\sqcap}$, avec $\bot_A^{\sqcap} \subseteq \boldsymbol{T}^{\cap}$. Par définition, tout type plus spécifique que le TCU absurde principal $\{\bot\}$ est absurde, et pour tout SymbolA $s \in \boldsymbol{S}_{\mathscr{T}}$, le TCU formé de son obligat et son prohibet (i.e., $\{\boldsymbol{\gamma_1}(s), \boldsymbol{\gamma_0}(s)\}$) est absurde.

Nous pouvons maintenant introduire la hiérarchie des types d'unités qui forme le cœur du formalisme des GU. Une hiérarchie de TCU est un n-uplet $\mathscr{T} \stackrel{\text{def}}{=} (T_D, \boldsymbol{S}_{\mathscr{T}}, \boldsymbol{\gamma}, \boldsymbol{\gamma_1}, \boldsymbol{\gamma_0}, C_A, \{\boldsymbol{\varsigma}_t\}_{t\in\boldsymbol{T}}, \bot_A^{\sqcap})$ qui est composé d'un ensemble de Types Primitifs d'Unités déclarés T_D, d'un ensemble de Symboles d'Actants $\boldsymbol{S}_{\mathscr{T}}$, de trois applications $\boldsymbol{\gamma}$, $\boldsymbol{\gamma_1}$ et $\boldsymbol{\gamma_0}$ qui associent à chaque SymbolA ses TPU radix, obligat et prohibet, d'un ensemble de comparaisons déclarées entre TPU C_A, de l'ensemble $\{\boldsymbol{\varsigma}_t\}_{t\in\boldsymbol{T}}$ des signatures des TPU, et d'un ensemble de Types Conjonctifs d'Unités déclarés absurdes $\bot_A^{\sqcap}$,

Les définitions des positions sont étendues aux TCU. L'ensemble des PosA (resp1. PosA obligatoires, resp2. PosA interdites) d'un TCU $t^\cap$ est noté $\boldsymbol{\alpha}^\cap(t^\cap)$ (resp1. $\boldsymbol{\alpha}_1^\cap(t^\cap)$, resp2. $\boldsymbol{\alpha}_0^\cap(t^\cap)$) et est l'union des PosA (resp1. PosA obligatoires, resp2. PosA interdites) des TPU qui le composent. L'ensemble des PosA optionnelles d'un TCU $t^\cap$ est noté $\boldsymbol{\alpha}_?^\cap(t^\cap)$ et est également les PosA qui ne sont ni obligatoires ni interdites. Les signatures sont également naturellement adaptées aux TCU. L'ensemble des signatures des TCU $\{\boldsymbol{\varsigma}_{t^\cap}^\cap\}_{t^\cap \in \boldsymbol{T}^\cap}$ est un ensemble de fonctions de $\boldsymbol{S}_\mathscr{T}$ vers $\boldsymbol{T}^\cap$. Pour chaque TCU $t^\cap$, $\boldsymbol{\varsigma}_{t^\cap}^\cap$ est une fonction avec $domain(\boldsymbol{\varsigma}_{t^\cap}^\cap) = \boldsymbol{\alpha}^\cap(t^\cap)$ qui associe à chaque PosA s de $t^\cap$ l'union des signatures $\boldsymbol{\varsigma}_t(s)$ des TPU t qui composent $t^\cap$.

$$\boldsymbol{\alpha}^\cap(t^\cap) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap} \boldsymbol{\alpha}(t) \tag{7}$$

$$\boldsymbol{\alpha}_1^\cap(t^\cap) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap} \boldsymbol{\alpha}_1(t) \tag{8}$$

$$\boldsymbol{\alpha}_0^\cap(t^\cap) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap} \boldsymbol{\alpha}_0(t) \tag{9}$$

$$\boldsymbol{\alpha}_?^\cap(t^\cap) \stackrel{\text{def}}{=} \boldsymbol{\alpha}^\cap(t^\cap) - \boldsymbol{\alpha}_1^\cap(t^\cap) - \boldsymbol{\alpha}_0^\cap(t^\cap) \tag{10}$$

$$\boldsymbol{\varsigma}_{t^\cap}^\cap(s) \stackrel{\text{def}}{=} \bigcup_{t \in t^\cap | s \in \boldsymbol{\alpha}(t)} \boldsymbol{\varsigma}_t(s) \tag{11}$$

Nous avons également introduit une relation de spécialisation sous la forme d'un pré-ordonnancement $\overset{\cap}{\lesssim}$ de l'ensemble des TCU $\boldsymbol{T}^\cap$ tel que (c.f., Lefrançois, 2013) : $\overset{\cap}{\lesssim}$ contient l'extension naturelle d'un pré-ordre sur un ensemble à un pré-ordre sur l'ensemble de ses sous-ensembles ; le bas de $\boldsymbol{T}^\cap$ est aplati de sorte que chaque TCU déclaré absurde soit inférieur à $\{\bot\}$; si la signature d'un TCU pour une PosA est inférieure à $\{\bot\}$, alors ce TCU est inférieur à $\{\bot\}$. Le bas de l'ensemble pré-ordonné $(\boldsymbol{T}^\cap, \overset{\cap}{\lesssim})$ correspond à l'ensemble des TCU équivalents à $\{\bot\}$ et est aplati : il contient $\{\bot\}$, chacun des TCU déclarés absurdes, et plus généralement l'ensemble des TCU qui ne peuvent pas être instanciés. On le nomme ensemble des TCU absurdes.

Nous montrons que les bonnes propriétés des TPU sont préservées par passage aux TCU, à part pour certains cas dégénérés (i.e., type vide et types absurdes).

4 Graphes d'Unités (GU)

Les TCU ainsi définis typeront les unités qui seront représentées par des nœuds unités dans les Graphes d'Unités. Nous allons clore cet article par la définition des GU *per se* et leur utilité pour formaliser des concepts plus avancés de la TST.

4.1 Hiérarchie des Symboles de Circonstants (SymbolC)

Les types d'unités spécifient comment les nœuds unités *sont* liés à d'autres nœuds unités dans un GU. Comme pour tout argument d'un prédicat, une PosA d'une unité ne peut être occupée que par une seule unité à la fois. Cependant on peut également rencontrer des dépendances

d'un autre type dans certaines représentations d'énoncé : des circonstants (Mel'čuk, 2004a). Les relations circonstancielles sont des relations instance-instance contrairement aux relations actancielles qui sont des relations prédicat-argument. C'est le cas des relations syntaxiques profondes non actancielles ATTR, COORD, APPEND par exemple, mais nous pourrons également utiliser ces relations pour représenter le lien entre une lexie et son sémantème par exemple.

Nous introduisons donc un ensemble fini de *Symboles de Circonstants (SymbolC)* noté $\boldsymbol{S}_{\mathscr{C}}$. Pour catégoriser et hiérarchiser l'ensemble des SymbolC, nous introduisons également un pré-ordre $\overset{\mathscr{C}}{\lesssim}$ sur $\boldsymbol{S}_{\mathscr{C}}$ construit par fermeture reflexo-transitive d'un ensemble de comparaisons déclarées $\boldsymbol{C}_{\boldsymbol{S}_{\mathscr{C}}} \subseteq {\boldsymbol{S}_{\mathscr{C}}}^2$. Enfin, à chaque SymbolC est associé une signature qui spécifie de quel type doivent être les unités liées par une relation avec ce symbole. L'ensemble des signatures des SymbolC $\{\boldsymbol{\sigma}_s\}_{s\in\boldsymbol{S}_{\mathscr{C}}}$ est un ensemble de couples de TCU : $\{(domain(s), range(s))\}_{s\in\boldsymbol{S}_{\mathscr{C}}}$. En descendant la hiérarchie des SymbolC et à l'instar des signatures des TCU, nous imposons que la signature d'un SymbolC ne peut que devenir de plus en plus spécifique.

On peut donc introduire la hiérarchie des SymbolC notée $\mathscr{C} \overset{\text{def}}{=} (\boldsymbol{S}_{\mathscr{C}}, \boldsymbol{C}_{\boldsymbol{S}_{\mathscr{C}}}, \mathscr{T}, \{\boldsymbol{\sigma}_s\}_{s\in\boldsymbol{S}_{\mathscr{C}}})$ et composée d'un ensemble fini de Symboles de Circonstants $\boldsymbol{S}_{\mathscr{C}}$, d'un ensemble de comparaisons déclarées de SymbolC $\boldsymbol{C}_{\boldsymbol{S}_{\mathscr{C}}}$, d'une hiérarchie de types d'unités $\mathscr{T}$, et de l'ensemble des signatures des SymbolC $\{\boldsymbol{\sigma}_s\}_{s\in\boldsymbol{S}_{\mathscr{C}}}$.

4.2 Définition des Graphes d'Unités

Les Graphes d'Unités permettent la description d'énoncés à différents niveaux de représentation. A l'instar des GC, les GU sont définis sur un *support* $\mathscr{S} \overset{\text{def}}{=} (\mathscr{T}, \mathscr{C}, \boldsymbol{M})$ qui est composé d'une hiérarchie de types d'unités $\mathscr{T}$, d'une hiérarchie de SymbolC $\mathscr{C}$, et d'un ensemble de marqueurs d'unités $\boldsymbol{M}$. Précisons ce que nous entendons par marqueurs d'unités. Nous établissons une distinction entre :

- les unités, qui sont les objets du domaine représenté ;
- les marqueurs d'unités, qui sont choisis dans l'ensemble $\boldsymbol{M}$, et qui identifient chacun une unité spécifique ;
- les nœuds unité, qui sont interconnectés dans des GU et qui représentent chacun une unité ;
- les marqueurs de nœuds unité, qui sont choisis dans l'ensemble noté $\boldsymbol{M}^{\cap}$ des parties de $\boldsymbol{M}$: $\boldsymbol{M}^{\cap} \overset{\text{def}}{=} 2^{\boldsymbol{M}}$, et qui étiquettent les nœuds unité afin de spécifier quelle unité chaque nœud unité représente.

Ceci peut paraitre complexe au premier abord, mais il s'agit en réalité d'une extension pour la TST qui nous permet d'être proche des GC, et une articulation simple avec les formalismes du web sémantique. En effet, chaque marqueur d'unité correspondra à une URI. Si un nœud unité est étiqueté par le marqueur de nœud unité $\varnothing$ (on dira que c'est un nœud générique d'unité), alors l'unité représentée est inconnue, il sera traduit par un blank-node en RDF. Dans la littérature de la TST, on considère que tous les nœuds des représentations d'énoncés sont génériques. Par contre, si un nœud unité est marqué $\{m_1, m_2\}$, alors les marqueurs d'unité m_1 et m_2 identifient en réalité la même unité, leurs ressources RDF correspondantes seront liées par une relation `owl:sameas`.

Dans leur version simple, les GC possèdent une relation d'équivalence de nœuds concepts nommée *coréférence*. Puisque cette relation ne correspond pas au terme linguistique et que nous représenterons la coréférence linguistique autrement, nous désactivons l'ambiguïté en parlant

plutôt de *relation d'équivalences déclarées de nœuds unités*, notée Eq. Deux nœuds unité déclarés équivalents représentent la même unité. De plus, contrairement à la relation $coref$ des GC, la relation Eq n'est pas une relation d'équivalence sur les nœuds unités [17]. Cela nous permet de distinguer connaissances explicites et connaissances implicites, et facilite l'articulation avec les formalismes du web sémantique.

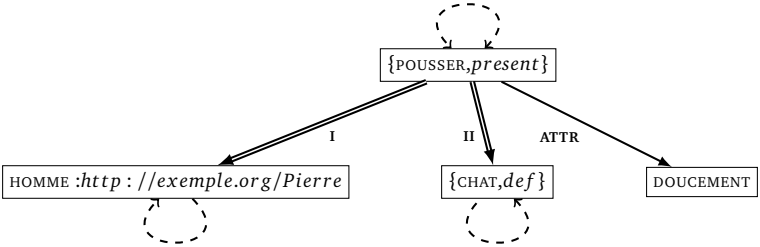

FIGURE 2 – Exemple de Graphe d'Unité : représentation syntaxique profonde de la phrase *Pierre pousse doucement le chat.*

Les GU permettent de représenter des énoncés à différents niveaux de représentation. Nous en avons déjà rencontré un sémantique sur la figure 1, et la figure 2 en représente un syntaxique profond. L'ensemble des GU définis sur un GU-support $\mathscr{S} = (\mathscr{T}, \mathscr{C}, \boldsymbol{M})$ est noté $\mathscr{G}(\mathscr{S})$ et chaque GU $G \in \mathscr{G}(\mathscr{S})$ est un n-uplet $G \stackrel{\text{def}}{=} (U, \boldsymbol{l}, A, C, Eq)$ avec :

– U est l'ensemble des *nœuds unités*. Ils sont illustrés par des rectangles comme sur la figure 2.
– $\boldsymbol{l}$ est une fonction d'étiquetage des noeuds unité. Pour un noeud unité u, $\boldsymbol{l}(u) = (t^\cap, m^\cap)$ est composé d'un TCU $t^\cap \in \boldsymbol{T}^\cap$ qui spécifie la nature de l'unité représentée, et d'un marqueur de nœud unité $m^\cap \in \boldsymbol{M}^\cap$ qui permet d'identifier l'unité représentée le cas échéant. Sur l'exemple de la figure 2, les nœuds unité sont tous typés par des singletons sauf un qui est typé $\{\text{TOMBE},present\}$. De plus, les nœuds unité sont tous génériques sauf pour un marqué $\{http://exemple.org/Pierre\}$.
– A est l'ensemble des *triplets actanciels* $(u, r, v) \in U \times \boldsymbol{S}_{\mathscr{T}} \times U$. Pour tout triplet actanciel (u, r, v), l'unité représentée par v remplit la PosA r de l'unité représentée par u. Ce sont les flèches doubles sur la figure 2.
– C est l'ensemble des *triplets circonstanciels* $(u, r, v) \in U \times \boldsymbol{S}_{\mathscr{C}} \times U$. Pour tout $c = (u, r, v) \in C$, l'unité représentée par u est liée à l'unité représentée par v par une relation circonstancielle de symbole r. Ce sont les flèches simples sur la figure 2.
– $Eq \subseteq U^2$ est l'ensemble d'*équivalences déclarées de nœuds unité*. $(u_1, u_2) \in Eq$ signifie que u_1 et u_2 représentent la même unité. Ce sont les arcs en pointillé sur la figure 2.

4.3 Concepts avancés du formalisme des GU

Les GU sont les briques de base qui vont nous permettre de représenter les connaissance du DEC. Nous allons présenter grossièrement quelques concepts avancés du formalisme des GU.

17. Une relation d'équivalence est une relation réflexive, symétrique et transitive.

Tout d'abord, la définition des GU est permissive et permet par exemple pour un triplet actanciel (u,r,v) d'un GU G, que le type de u n'ait pas de PosA r. A l'instar des formalismes du web sémantique, nous faisons l'hypothèse d'un monde ouvert et considérons que le GU, tout comme la hiérarchie des TCU, représente des connaissances explicites. Nous pouvons donc expliciter dans G le fait que le type de u doive contenir le radix de r. Nous nous inspirons de OWL-RL pour définir l'ensemble des opérations d'explicitation des connaissances, et définissons ainsi la sémantique, au sens logique, des GU. Puisque nous avons dérivé les GC pour définir les GU, nous adapterons les résultats de raisonnement à base de graphes des GC, et définirons la notion d'implication d'un GU par un autre à l'aide d'homomorphismes de graphes. Ces raisonnements logiques seront utiles en particulier pour les représentations sémantiques.

Ensuite, en nous inspirant encore des GC, nous pouvons définir la notion de règles comme un triplet formé d'un GU hypothèse H, d'un GU conclusion C, et d'une bijection κ entre un sous-ensemble de nœuds unités génériques de H et un sous-ensemble de nœuds unités génériques de C. Les règles permettront de représenter les associations sémantème-lexie, et les correspondances entre différents niveaux de représentation (tableaux de régime). Les règles ont été très étudiées pour les GC et nous pourrons adapter aux GU les résultats les concernant.

Nous pouvons également représenter certaines connaissances de la hiérarchie des TCU qui concernent un TPU t dans un GU que l'on nomme *empreinte* de t. L'empreinte de t est un GU avec un nœud central u ayant pour étiquette $\boldsymbol{l}(u) = (\{t\},\varnothing)$, et pour chaque PosA s de t, un autre nœud unité v avec $\boldsymbol{l}(v) = (\varsigma_t(s),\varnothing)$ et un triplet actanciel (u,s,v). Une définition d'une lexie L est alors formée de deux règles réciproques, dont un GU est l'empreinte d'un sémantème t, l'autre GU est la représentation sémantique de la définition lexicographique de la lexie L, et la bijection permet entre autre de repérer le sens du genre prochain de L. A chaque définition correspond donc deux règles d'explicitation des connaissances.

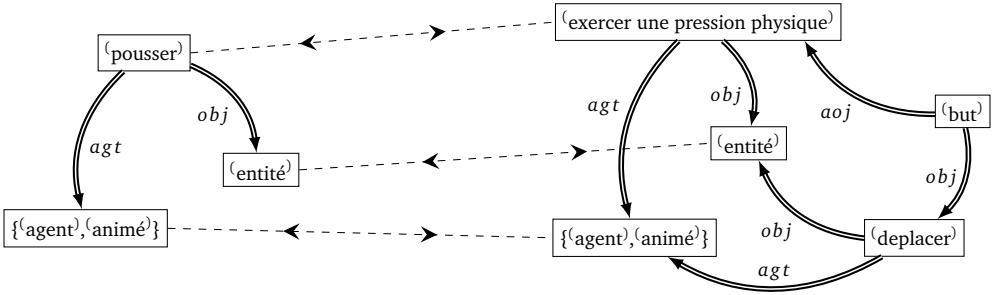

FIGURE 3 – Définition de POUSSER. A gauche, l'empreinte de ⟨pousser⟩.

5 Conclusion

Nous avons donc étudié comment formaliser, au sens de l'ingénierie des connaissances, le Dictionnaire Explicatif et Combinatoire (DEC), et ce afin de pouvoir représenter, manipuler, interroger et raisonner sur des connaissances linguistiques. Nous pouvons maintenant répondre aux questions que nous avons posées.

En quoi les formalismes du web sémantique et le formalisme des Graphes Conceptuels (GC) ne sont-ils pas directement adaptés pour représenter les connaissances du DEC ? La sémantique, au sens logique, de RDF est insuffisante pour représenter les connaissances du DEC, et nous avons montré que l'utilisation de OWL présente des problèmes majeurs. Le formalisme des GC présente des ressemblances avec la Théorie Sens-Texte mais ne permet pas de représenter la dualité concept/relation de la modélisation d'un sémantème. Nous avons proposé de revisiter les bases des GC afin d'en dériver le nouveau formalisme des Graphes d'Unités.

Quelle structure mathématique pour une hiérarchie de types d'unités pouvant avoir des positions actancielles ? Pour prendre en compte la dualité concept/relation des sémantèmes, les relations prédicat-argument sont symbolisées par des Symboles d'Actants (SymbolA), et nous associons à chaque SymbolA trois Types Primitifs d'Unités (TPU) : un radix $\gamma(s)$ qui introduit une Position Actancielle (PosA) de symbol s, un obligat $\gamma_1(s)$ qui rend cette PosA obligatoire, et un prohibet $\gamma_0(s)$ qui rend cette PosA interdite. Ainsi dans l'ensemble pré-ordonné des TPU, une PosA ayant pour SymbolA s est introduite par $\gamma(s)$, et est d'abord optionnelle pour tout TPU plus spécifique que $\gamma(s)$ tant que ce TPU n'est pas plus spécifique que $\gamma_1(s)$ ou $\gamma_0(s)$ auquel cas la PosA devient obligatoire ou interdite. Chaque TPU qui possède des PosA représente donc également un type de relation, qui peut, doit, ou ne doit pas lier une instance de ce type à l'ensemble de ses actants. Enfin, à chaque TPU est associé une signature qui spécifie le type des actants de ses unités. Nous avons étendu les définitions des types d'unités à leur version conjonctive et avons donc introduit la hiérarchie des types d'unités.

Quel est l'équivalent des graphes conceptuels pour le formalisme des Graphes d'Unités (GU), et comment les utiliser pour formaliser des concepts plus avancés de la TST ? Nous avons introduit une hiérarchie des symboles de circonstants. Nous avons ensuite illustré la définition des GU, qui représentent des nœuds unités interconnectés par des relations de dépendance, et des relations d'équivalences déclarée. Nous avons brièvement présenté les concepts plus avancés de la TST que les GU permettent de représenter, et sur lesquels nous travaillons actuellement :

– Nous pouvons définir la sémantique des GU, et donc raisonner avec des représentations d'énoncés.
– Les règles nous permettent de représenter les associations sémantème-lexie, et les correspondances entre différents niveaux de représentation (tableaux de régime).
– Nous pouvons représenter les définitions lexicographiques du DEC à l'aide de deux règles réciproques.

Nous travaillons également sur la factorisation des règles qui nous permettra de représenter des liens de fonctions lexicales, ainsi que sur une syntaxe basée sur les standards du web sémantique pour permettre l'échange standardisé de connaissances du DEC, en particulier sur le web de données.

Remerciements

Je tiens à remercier chaleureusement S. Kahane ainsi que les relecteurs des différentes versions de cet article. Un grand merci également à F. Gandon pour son encadrement, ses précieux conseils et sa disponibilité.

Références

BAGET, J. F., CROITORU, M., GUTIERREZ, A., LECLERE, M. et MUGNIER, M. L. (2010). Translations between RDF (S) and conceptual graphs. *Conceptual Structures : From Information to Intelligence*, page 28–41.

BARQUE, L. et POLGUÈRE, A. (2008). Enrichissement formel des définitions du Trésor de la Langue Française informatisé (TLFi) dans une perspective lexicographique. 22.

CHEIN, M. et MUGNIER, M. L. (2008). *Graph-based Knowledge Representation : Computational Foundations of Conceptual Graphs*. Springer.

CORBY, O., DIENG, R. et HÉBERT, C. (2000). A conceptual graph model for W3C resource description framework. *In* GANTER, B. et MINEAU, G. W., éditeurs : *Conceptual Structures : Logical, Linguistic, and Computational Issues*, numéro 1867 de Lecture Notes in Computer Science, pages 468–482. Springer Berlin Heidelberg.

KAHANE, S. et POLGUÈRE, A. (2001). Formal foundation of lexical functions. *In Proceedings of ACL/EACL 2001 Workshop on Collocation*, page 8–15.

KRISNADHI, A., MAIER, F. et HITZLER, P. (2011). OWL and Rules. *Reasoning Web. Semantic Technologies for the Web of Data*, page 382–415.

LEFRANÇOIS, M. (2013). The Unit Graphs Mathematical Framework. Rapport de recherche RR-8212, INRIA.

LEFRANÇOIS, M. et GANDON, F. (2011). ILexicOn : Toward an ECD-Compliant Interlingual Lexical Ontology Described with Semantic Web Formalisms. *In Proc. of the 5th International Conference on Meaning-Text Theory (MTT 2011)*, page 155–164, Barcelona, Spain. INALCO.

LUX-POGODALLA, V. et POLGUÈRE, A. (2011). Construction of a French Lexical Network : Methodological Issues. *In Proceedings of the International Workshop on Lexical Resources*, Ljubljana.

MEL'ČUK, I. A. (1996). Lexical functions : a tool for the description of lexical relations in a lexicon. *Lexical functions in lexicography and natural language processing*, 31:37–102.

MEL'ČUK, I. A. (2004a). Actants in Semantics and Syntax I : Actants in Semantics. *Linguistics*, 42(1):1–66.

MEL'ČUK, I. A. (2004b). Actants in Semantics and Syntax II : Actants in Syntax. *Linguistics*, 42(2):247–291.

MEL'ČUK, I. A. et ARBATCHEWSKY-JUMARIE, N. (1999). *Dictionnaire explicatif et combinatoire du français contemporain : recherches lexico-sémantiques*, volume 4. PU Montréal.

MEL'ČUK, I. A. (2006). Explanatory Combinatorial Dictionary. *Open problems in linguistics and lexicography*, page 225.

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

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

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