Approche oecuménique au problème du codage des structures linguistiques

Gilles Sérasset

Approche œcuménique au problème du codage des structures linguistiques

Gilles Sérasset
GETA - IMAG (UJF & CNRS)
BP 53
38041 Grenoble Cedex 9
e-mail: Gilles.Serasset@imag.fr

Le Traitement Automatique du Langage Naturel en France Aujourd'hui
Marseille
7 - 8 Avril 1994

Résumé

En étudiant les principaux projets de bases lexicales en Europe ou au Japon, on se rend compte de l'importance croissante des recherches sur le domaine. Par contre, on remarque aussi la quasi absence de solutions génériques au problème de la définition des structures linguistiques des éléments d'une base lexicale.

Certes, le projet Multilex et le système "Le Lexicaliste" offrent au linguiste un langage pour définir ses structures linguistiques. Mais ces langages imposent une structure logique avec laquelle le linguiste doit coder ses structures linguistiques.

Nous proposons une approche œcuménique au problème de la définition des structures linguistiques d'un dictionnaire. Cette approche passe par un langage spécialisé qui permet l'utilisation de structures logiques différentes (arbres, graphes, automates, ensembles, listes, arcs, …) dans les structures linguistiques du lexique.

Ainsi, le linguiste peut choisir la structure logique la plus adéquate pour coder sa théorie linguistique. Il peut de plus mélanger différentes structures logiques dans une même structure linguistique. Nous donnons dans cet article un aperçu du langage spécialisé utilisé dans le projet NADIA.

Nous justifions enfin l'intérêt du mélange de différentes structures logiques dans une même structure linguistique. Pour cela, nous proposons un codage assez naturel d'une entrée du Dictionnaire Explicatif et Combinatoire (DEC) du français contemporain par Igor Mel'čuk.

Mots Clés

Bases de données lexicales multilingues, définition de structures linguistiques.

Introduction

Les besoins en ressources lexicales de grande taille pour le Traitement Automatique des Langues Naturelles (TALN) en général, et pour la Traduction Automatique (TA) en particulier, augmentent chaque jour. Ces ressources représentent généralement la plus grande partie du coût d'un système de TALN. Aussi, on peut observer un intérêt croissant pour le développement de dictionnaires réutilisables.

De nombreux projets, nationaux et internationaux ont développé des systèmes de bases de données lexicales monolingues ou multilingues indépendants d'une application linguistique particulière. Ces systèmes proposent des structures logiques et des structures linguistiques variées. Par contre, force est de constater que ces systèmes permettent en général l'utilisation d'un seul type de structure logique, ce qui contraint fortement les structures linguistiques représentables.

Aussi, nous proposons dans cet article une approche œcuménique permettant à un linguiste de choisir parmi un ensemble important de structures logiques celle(s) qui lui sera(seront) utile(s) pour coder ses structures linguistiques. Cette approche nécessite une description (par le linguiste) des structures présentes dans la base de données lexicales. Moyennant celle-ci (à l'aide d'un langage spécialisé), le système NADIA sera à même de fournir au lexicographe des outils de manipulation des entrées du lexique.

L'utilisation d'un langage de description de la structure des entrées lexicales a un avantage immédiat : le système informatique est clairement séparé des théories linguistiques (d'où une grande adaptabilité du logiciel). L'approche œcuménique (utilisation de structures logiques différentes) adoptée par le projet NADIA permet à un linguiste de coder ses structures linguistiques de manière plus aisée en ayant la possibilité de mélanger les structures logiques dans une seule entrée.

Nous commencerons cet article en passant en revue les différentes solutions utilisées dans les projets de bases lexicales, en terme de structures logiques et linguistiques. Nous continuerons en discutant l'intérêt d'une approche générique et œcuménique au problème du codage des informations lexicales. Enfin, nous terminerons par un exemple de l'utilisation, dans une même unité lexicale, de structures logiques différentes.

I. État des lieux

1. Multilex

Multilex est un projet européen ESPRIT (DG XIII). L'objectif de ce projet est de définir des standards (informatiques et linguistiques) pour les bases lexicales multilingues, les langues considérées étant essentiellement celles de la Communauté Européenne.

```
<language> EN
<item>
     <word-sense>
          <lu>
               <gr-canon> ballast
               <homograph-number> 0
               <meaning-number> 1
          <gpmu-id>
               <gr-canon> ballast
               <gpmu-number> 1
          <syntactic-description>
               <syntax>
                    <cat> noun
                    <major>
                         <subcat>
                              <arity> 0
                              <syn-cat>
                              <syn-rel>
                              <selectional-restrictions>
                              <subcat-rules>
```

Description de l'entrée anglaise "ballast" sous forme MLEXd.

Dans sa première phase, le projet Multilex a défini des standards sur les structures logiques d'une unité lexicale. Il a été décidé d'utiliser des structures de traits typés.

Pour Multilex, l'unité lexicale est le sens d'un mot. L'entrée dans le lexique se fait via une GPMU (Graphic-Phonologic-Morphological Unit). Multilex a défini un ensemble important de traits (avec leur valeurs possibles) qui sont associés aux éléments du lexique (GPMU, sens).

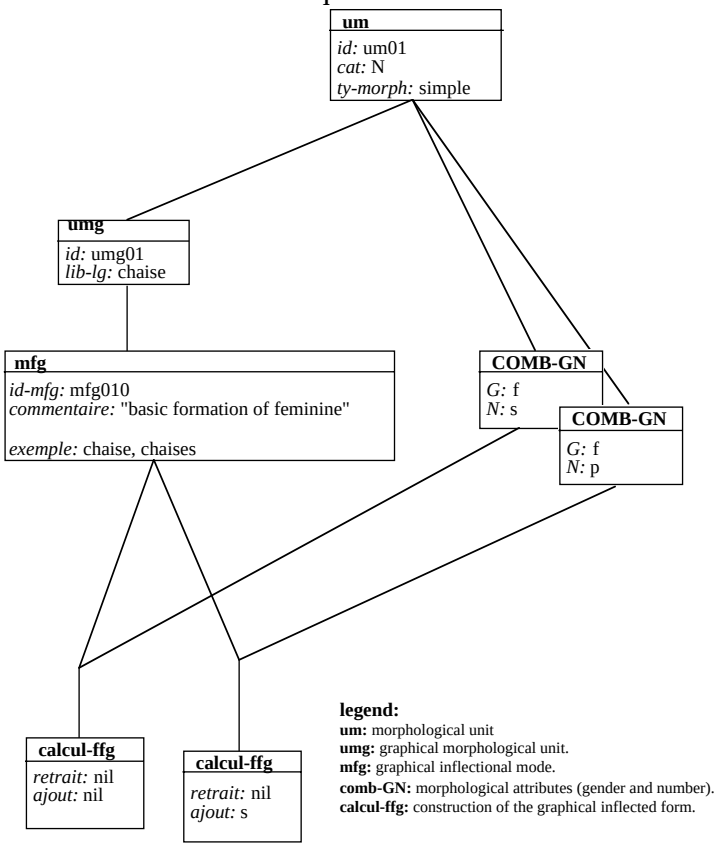
La seconde phase du projet, visant l'implémentation d'une base lexicale illustrant les différents standards a décidé de ne pas utiliser la théorie sous-jacente aux structures de traits typés (ayant une sémantique proche des langages à objets). Cette phase a mis en œuvre des structures de traits codées en SGML (appelée MLEXd) qui offrent une expressivité analogue en évitant les problèmes dus à la complexité des algorithmes d'unification qu'impliquent le modèle théorique.

2. Genelex

Genelex est un projet européen EUREKA impliquant des partenaires Français, Italiens et Espagnols. Son objectif est de construire un dictionnaire générique pour différentes langues Européennes (pour l'instant le français, l'italien et l'espagnol).

Pour Genelex, une unité lexicale est le sens d'un mot, défini par les relations entre une unité morphologique, une unité syntaxique et une unité sémantique [Lay et al. 1992, Zaysser et al. 1992].

Genelex a choisi de coder ses dictionnaires dans un format entité-relation. Ce choix permet la visualisation de l'unité lexicale comme un graphe. Ainsi, chaque élément d'information est placé à un niveau équivalent (pas de nœud privilégié pour la recherche d'information, alors qu'une structure d'arbre ou une structure de traits typés privilégie la racine), ce qui facilite l'écriture des requête.



Exemple d'unité morphologique.

3. EDR

Le projet japonais EDR (Electronic Dictionary Research), est un projet de base lexicale multilingue de grande dimension. Il a abouti à la construction de dictionnaires monolingues Anglais et Japonais de 300 000 termes chacun, d'un dictionnaire de 400 000 concepts et des dictionnaires bilingues (un par sens) de 400 000 entrées bilingues [EDR 1993].

Les dictionnaires monolingues contiennent des unités sémantiques constituées de quatre parties : informations sur l'entrée, informations grammaticales, informations sémantiques (un identificateur de concept) et des informations supplémentaires (fréquences, …). Ces informations sont codées sous forme d'un ensemble d'attributs simples. Les informations grammaticales d'une entrée composée consistent en un arbre syntaxique décrivant les composants de l'entrée.

Word	**Headword**	
	Retrieval Entry	**Constituent Information**
study	stud(ELV1,ECV5)	
eat	eat(ELV2,ECV7)	
ate	ate(ELV2,ERV3)	
give up	give(ELV1,ECV9)	/w#suf(*,*)/(ELB1,ERB1) /up(ELW1,ERD5)

Exemple d'informations sur l'entrée d'un article du dictionnaire monolingue anglais de EDR.

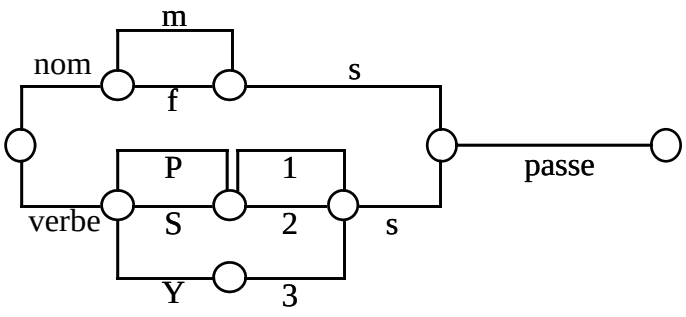
Le dictionnaire de concepts de EDR contient une description des concepts. Cette description se fait au travers de relations entre concepts. EDR utilise 32 types de relations et 5 types d'attributs pour cette description. Les concepts sont classifiés dans une hiérarchie.

Ainsi, les structures logiques mises en œuvre dans cette base lexicale sont :
- des ensembles de valeurs atomiques,
- des graphes étiquetés (relations entre concepts),
- une hiérarchie de concepts.

4. Dictionnaires du LADL

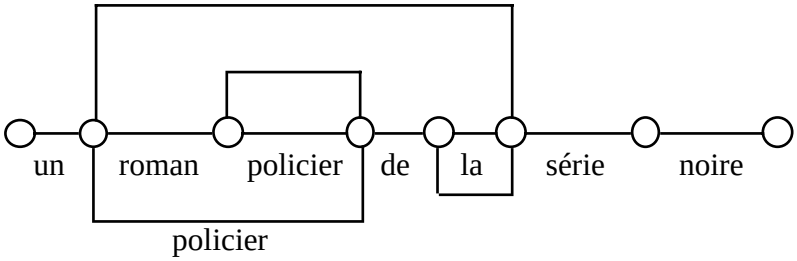
Le LADL (Laboratoire d'Automatique Documentaire et Linguistique) a développé les dictionnaires DELAF (dictionnaire de 600 000 formes simples fléchies) et DELACF (dictionnaire de 150 000 formes composées fléchies) [Gross 1987].

Dans ces dictionnaires, les entrées sont représentées comme des automates d'états finis. Cette représentation permet un codage puissant des informations associées aux entrées. Une entrée simple est donnée ci-dessous (avec informations morphologiques et syntaxiques).



Représentation de la forme simple fléchie "passe".
Les abréviations sont : m: masculin; f: féminin; s: singulier; 1,2,3: personne; P: présent; S: subjonctif et Y: impératif

Ces mêmes automates sont aussi utilisés pour représenter une forme composée ayant plusieurs variantes.



Représentation de la forme complexe “un roman policier de la série noire”. Cette forme peut être lue : “un série noire”, “un policier de série noire”, “un roman de série noire”, …

5. Le Lexicaliste

Le Lexicaliste est un système de gestion de bases lexicales monolingues développé et commercialisé par la société SITE.

Le Lexicaliste s’appuie sur une description des entrées du lexique. Les articles sont des arbres décorés dont la racine correspond à l’entrée du dictionnaire (lemme) et les nœuds aux différents sens de l’article. Les décorations sont des structures attributs-valeurs simples.

Le linguiste définit quels sont les attributs (et les valeurs) qui sont utilisés dans une base lexicale particulière (cette description est appelée référentiel). Il peut aussi donner des propriétés (attributs monovalués, multivalués, relations acycliques,…) sur les attributs de la base (ces propriétés sont contenues dans le méta-référentiel).

Les attributs sont séparés en 5 catégorie distinctes:
- attributs des lemmes (ex : `catégorie`),
- attributs des sens (ex : `transitivité, définition`),
- attributs des règles flexionnelles (ex : `nombre, genre`),
- relations lexicales (ex : `abréviation, dérivation`),
- relations sémantiques (ex : `hyperonymie, synonymie`).

Les relations lexicales et sémantiques définissent deux réseaux sur l’ensemble de la base lexicale.

Lorsque ce travail de définition a été accompli, les tables SQL et l’interface sont automatiquement générées par le système.

II. Approche œcuménique

1. Vue générale

Parmi les projets de bases lexicales étudiés ci-dessus, seul Le Lexicaliste utilise un langage permettant au linguiste de définir ses structures, ce qui donne à ce système une très grande souplesse.

Par contre, ce système utilise des arbres décorés par des structures attributs-valeur classiques, et ce choix ne peut être remis en cause par le linguiste. Il serait donc impossible d’utiliser ce système pour stocker le dictionnaire DELAF du LADL par exemple.

Aussi, nous proposons un système de gestion de bases de données lexicales générique : le projet NADIA [Sérasset et Blanc 1993]. Ce projet vise 4 objectifs principaux :

- **Généricité** : les structures linguistiques utilisées par les bases seront définies par un linguiste, via un langage spécialisé.
- **Indépendance vis-à-vis des théories** : le système ne doit pas introduire de restrictions sur la théorie linguistique sous-jacente à une base. Il doit au contraire permettre l’utilisation de nombreuses théories linguistiques qui mettent en œuvre de nombreuses structures logiques.

- **Indépendance vis-à-vis des applications** : le système n'introduit pas de restriction sur les applications qui utiliseront les bases définies. Toute utilisation des bases est possible (pour traduction, correction, apprentissage des langues par l'homme), pourvu que l'information linguistique nécessaire soit présente.
- **Multilinguisme** : le système gère des bases de données multilingues. Aussi devrons-nous prendre en compte les différents systèmes d'écriture et les différentes procédures de tri.

La définition des structures linguistiques d'un dictionnaire est une démarche analogue à la définition de la structure d'un document structuré [André et al. 1989], ou à la DTD (Document Type Definition) d'un document SGML (Standard Generalized Markup Language) [Herwijnen 1990].

Le langage utilisé permet, par les mécanisme d'héritage et de spécialisation d'un langage à objet, de définir des structures linguistiques basées sur :
- un arbre,
- un graphe,
- un automate,
- une structure de traits (typée ou non),
- un ensemble,
- une liste,
- …

Chaque noeud ou arc de ces structures peut être décoré par une autre structure complexe.

2. Description des structures du lexique

Nous donnons dans ce paragraphe une vue d'ensemble du langage spécialisé pour la définition des structures linguistiques du lexique. Nous donnons la forme LISP de cette définition (une autre forme sera définie pour le linguiste).

1.1. Definition d'une base multilingue

Avant de définir les structures linguistiques d'une base multilingue, le linguiste doit définir la base elle-même. Celle-ci consiste en un certain nombre de dictionnaires monolingues.

En plus du nom, de la langue et du propriétaire d'un dictionnaire, on donne le type correspondant à ses entrées (accès au dictionnaire) et à ses acceptions (les sens d'un mot : les unités lexicales).

L'exemple suivant définit une base trilingue :

```
(database example
  :owner "GETA"
  :comment "base lexicale trilingue"
  :dictionaries
    (dictionary French
      :language "Français"
      :owner "GETA"
      :entry 'French-entry
      :acception 'French-acception)
    (dictionary English
      :language "English"
      :owner "GETA"
      :entry 'English-entry
      :acception 'English-acception)
    (dictionary German
      :language "Deutch"
      :owner "GETA"
      :entry 'German-entry
      :acception 'German-acception)
)
```

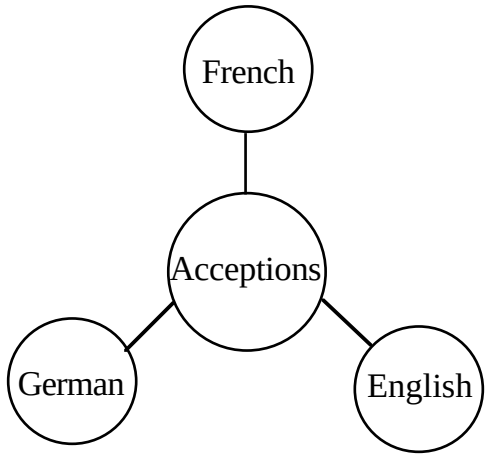


Schéma de la base trilingue définie ci-contre

1.2. Structure d'un dictionnaire monolingue

Le linguiste défini les structures linguistiques du dictionnaire avec un langage à objets.

Deux types de bases sont prédéfinies dans le système : *entry* et *acception*. Ces types répondent à des opérateurs standards permettant l'accès au lexique. Le linguiste dérive ces types de base pour définir la structure du lexique.

Dans l'exemple suivant, le linguiste défini l'entrée du dictionnaire français comme un arbre dont la racine est décorée par une structure de traits typée et dont les feuilles sont des acceptions françaises.

Pour cela, le type *french-entry* hérite des types *entry* (donc des opérateurs prédéfinis d'une entrée de dictionnaire) et *tree*, ce qui nous permet de manipuler une structure d'arbre et de contraindre les types de décoration de l'arbre (ici, la racine et les feuilles).

```
(type french-entry (entry tree)
  :root   (feature-structure (graphic-form string)
                             (category category))
  :leaves french-acception)

(type category ()
  (one-of 'nc 'np 'vb 'vbimp 'vbrefl 'adj 'card 'deict 'repr 'sub 'coord))
```

Le type prédéfini *acception* représente l'unité du lexique. Le linguiste dérive ce type en lui associant une structure. Nous donnons ci-dessous un exemple simple d'une structure représentée sous forme d'une structure de trait contenant des informations sur les valences, les dérivations, etc.

```
(type french-acception (acception)
  (feature-structure
   (cat category)
   ;; informations de dérivation.
   (drvv (feature-structure
          (deriv-kind
            (one-of 'naction 'nresult 'nlieu 'nagent 'ninstr 'adject
                    'adjpass 'adjpotpas 'adjresact 'verbe))
          (deriv-from symbol)))
   (drvn (feature-structure
          (deriv-kind
            (one-of 'ncond 'nlieu 'ninstr 'ncollect
                    'nperson 'adjrelat 'adjqual 'verbe))
          (deriv-from symbol)))
   (drva (feature-structure
          (deriv-kind
            (one-of 'nabst 'nperson 'verbe))
          (deriv-from symbol)))
   ;; informations sur les valences
   (val0 valency)
   (val1 valency)
   (val2 valency)
   (val3 valency)
   (val4 valency)
   ;; autres informations
   (gnr (any-of 'masc 'fem))
   (nbr (any-of 'sg 'pl))
   (aux (one-of 'être 'avoir))
   (reciproque (one-of 'arg0-arg1 'arg1-arg1))
   (aspect (one-of 'achevé 'inachevé 'début 'fin 'duratif 'fréquent
                   'instantané))))

(type valency ()
  (any-of 'nom 'à+nom 'avec+nom 'comme+nom
      'contre+nom 'dans+nom 'de+nom 'en+nom
      'entre+nom 'par+nom 'parmi+nom 'pour+nom
      'sur+nom 'inf 'à+inf 'de+inf 'adj 'que+ind
      'que+subj 'se-moy 'se-pass 'lieu-stat 'lieu-dyn
      'manière 'zéro))
```

Nous avons ainsi défini la structure d'un dictionnaire français simple.

III. Un exemple

1. Une entrée lexicale complexe

Les informations que l'on souhaite coder ne sont pas forcément simples, ni même facilement représentables sous formes de structures de traits. Pour nous en convaincre, reprenons un article du dictionnaire explicatif et combinatoire du français contemporain [Mel'č uk 1984, Mel'č uk 1988].

Les articles du DEC sont composés de deux parties principales. La première est l'entrée. La seconde est une liste de sens (acceptions).

Une entrée consiste en une forme graphique (avec des information morphologique, suivie d'un ensemble de sens numérotés :

ASSISTANCE, nom, fém, pas de pl.

I. Ensemble de personnes qui assistent I… [Sa conférence a charmé l'assistance]
II. 1. S0(assister II.1) [l'assistance d'un technicien pour évaluer le travail des employés]
2. S0(assister II.2) [l'assistance financière aux sinistrés]
3. Organisme ou programme visant l'aide 1b à Y… [l'assistance sociale]

Chacun des sens est repris ensuite en détail. La définition est reprise, suivie d'informations sur le régime et les fonctions lexicales de l'acception étudiée. Dans certains cas, des contraintes sont rajoutées au tableau codant le régime pour interdire des constructions invalides.

II.1. Assistance de X [à Y] pour U = S0(assister II.1).

Régime :

1 = X	**2 = Y**	**3 = U**
1. de N		1. dans N
2. Aposs	----------	2. pour N
		3. pour V_{inf}

C1 : l'assistance d'une infirmière <d'un technicien>, son assistance

C3 : l'assistance dans le travail <pour l'opération, pour évaluer le travail des employés>

C1 + C3 : l'assistance d'une infirmière <son assistance> dans ce travail <pour l'opération>, l'assistance d'un technicien <son assistance> pour évaluer le travail des employés

Fonctions lexicales :

	$Syn_{\cap}$	:	aide 1b
	S_1	:	assistant II.1
	$S_{1\subset}$	:	assistant II.2
	Oper1	:	apporter [A_{poss} ~ à N]
	Oper2	:	recevoir [ART ~]
tenter de	Caus2Func0	:	demander [l' ~ de N]
	Adv2Real2	:	avec [l'~] | C1 ≠ Λ

2. Un codage assez naturel

Il est assez complexe de coder cet article de dictionnaire avec l'un des outils présentés dans la première partie. En effet, chacune des structure utilisée peut-être assez agréable pour coder une partie de ce dictionnaire, mais elle sera moins adapté pour une autre partie.

L'approche œcuménique utilisée dans le projet NADIA apporte une réponse élégante à ce problème en offrant au linguiste la possibilité de mélanger différentes structures logiques dans la définition d'une même structure linguistique. Ainsi, le linguiste est à

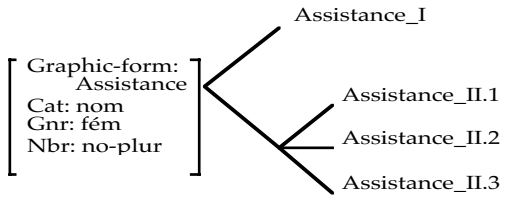
même de choisir la structure logique la plus adaptée à chacune des différentes parties d'un article.

Ainsi, dans l'exemple qui nous intéresse, un entrée sera codée comme un arbre dont la racine est décoré par une graphie et des informations morphologiques et dont les feuilles sont des acceptions définies plus bas.

```
(type melcuk-entry (entry tree)
  :root   (feature-structure
            (graphic-form string)
            (cat category)
            (gnr gender)
            (nbr number))
  :leaves melcuk-acception)

(type category ()
  (one-of 'nom 'nom-propre 'verbe
          'adjectif))

(type gender ()
  (any-of 'fem 'masc))
```



Un exemple d'une entrée du DEC représentée graphiquement selon la description données ci-dessous.

Les acceptions du DEC sont des structures plus complexes. En surface, une acception est constituées de différents éléments (définition, régime, fonctions lexicales). Il est donc assez naturel de coder cette acception comme une structure de traits avec des valeurs complexes.

```
(type melcuk-acception (acception)
  (feature-structure (acception-id symbol)
                     (déf definition)
                     (régime reg)
                     (fonctions_lexicales lexfns)))
```

La définition d'une acception peut être représentée par une simple chaîne de caractères :

```
(type definition (string))
```

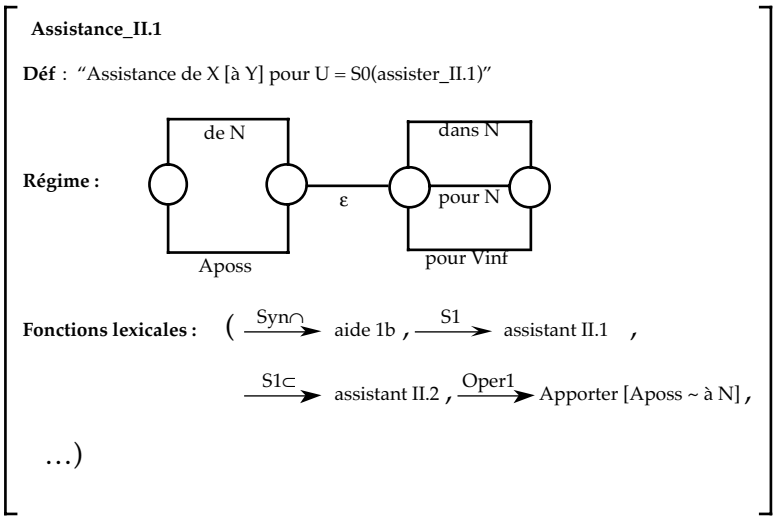
Le régime d'une acception est représenté dans le DEC par un tableau donnant les différentes réalisations des arguments, avec éventuellement des contraintes précisant l'incompatibilité de certaines réalisations. Un automate semble très adapté pour la représentation d'une telle information. En effet, avec une telle structure, on représente aisément les réalisation et leurs incompatibilités (chaque chemin définit une réalisation valide des arguments).

```
(type reg (automaton)
   :arcs (feature-structure (realisation string)))
```

Les fonctions lexicales sont en général perçues comme des arcs d'un graphe lexical recouvrant l'ensembles des acceptions du dictionnaire. Nous allons donc les représenter en tant qu'arcs dont les étiquettes sont des noms de fonctions lexicales.

```
(type lexfns (set)
   :elements lexfn-arc)

(type lexfn-arc (arc)
   :decoration string
   :target string))
```



Un exemple d'une acception du DEC représentée graphiquement selon la description données ci-dessus.

Conclusion

Dans ce papier, nous avons présenté une partie de nos études sur le développement d'un système de gestion de bases lexicales multilingues : NADIA.

Ce système introduit de nouvelles perspectives dans le domaine des bases lexicales multilingues. Par l'utilisation d'un langage spécialisé, le linguiste est capable de définir la structure des unités de son lexique. Ce langage lui permet de choisir les structures logiques les mieux adaptées à sa théorie linguistique parmi un ensemble de structures logiques de base qu'il peut composer et dériver.

Notre objectif actuel est le prototypage de ce langage et de l'environnement de gestion de base lexicale qui lui est associé (définition et évaluation de contraintes de cohérence sur les éléments de la base, visualisation des différentes structures, import/export de/vers des fichiers SGML, …).

Parmi les différentes tâches que cela représente, nous nous attacherons plus particulièrement au problème de la visualisation d'information complexe. Pour cela, nous faisons un parallèle entre dictionnaires et documents structurés (ce qui nous permet d'utiliser les résultats des recherches de ce domaine). Nous nous intéressons de plus à l'étude des possibilité de visualisation d'une même structure sous différentes formes (graphiques et logiques).

Références

André J., Furuta R. et Quint V. (1989). *Structured Documents*, The Cambridge Series on Electronic Publishing, Cambridge, Cambridge University Press, ISBN 0-521-36554-6.

EDR (1993). *EDR Electronic Dictionary Technical Guide*, Japan Electronic Dictionary Research Institute Ltd., Project report n° TR-042, August 16, 1993 : 144 p.

Gross M. (1987). *The Use of Finite Automata in the Lexical Representation of Natural Language*, Electronic Dictionaries and Automata in Computational Linguistics- LITP Spring School on Theoretical Computer Science, St Pierre d'Oléron : pp. 34-50, ISBN 3-540-51465-1.

Herwijnen E. V. (1990). *Practical SGML*, Kluwer Academic Publishers, Dordrecht(Nl.), ISBN 0-7923-0635-X.

Lay M.-H., Zaysser L. et Flores S. (1992). *Projet Eureka Genelex, le modèle syntaxique*, Projet Eureka Genelex, Rapport technique, 10 juin 1992 : 107 p.

Mel'č uk I. (1984). *DEC : Dictionnaire explicatif et combinatoire du français contemporain*, Montréal(Quebec), Canada, Presses de l'université de Montréal, ISBN 2-7606-0659-7.

Mel'č uk I. (1988). *DEC : Dictionnaire explicatif et combinatoire du français contemporain*, Montréal(Quebec), Canada, Presses de l'université de Montréal, ISBN 2-7606-0804-2.

Sérasset G. et Blanc É. (1993). *Une approche par acceptions pour les bases lexicales multilingues*, T-TA-TAO 93, Montréal, 30/9-2/10/93, 15 p.

Zaysser L. et al. (1992). *Projet Eureka Genelex, couche morphologique*, Projet Eureka Genelex, Rapport Technique, 2 juin 1992 : 97 p.