Grammaires Locales: Principes, Modélisation et Utilisation[1]

Pierre Lison (étudiant)
plison@student.fsa.ucl.ac.be

Université Catholique de Louvain
Faculté de Philosophie et Lettres
Centre de Traitement Automatique du Langage

21 décembre 2004

[1]Ce travail a été réalisé dans le cadre du cours "Introduction au Traitement du Langage Naturel" (FLTR 2620) du Prof. Cédrick Fairon durant l'année académique 2004-2005.

Table des matières

1. Introduction

1.1 Le Traitement Automatique des Langues

On peut catégoriser les recherches actuelles en Traitement Automatique des Langues suivant deux grandes tendances :

- Les approches **symboliques** se basent sur des descriptions et des modélisations explicites de la langue naturelle. Les grammaires catégorielles, à états finis, d'unification, de dépendance ... font ainsi partie de cette (large) catégorie. Elles partent d'hypothèses et de méthodologies très diverses, depuis l'utilisation de simples automates à l'implémentation de modèles sophistiqués, mais ont toutes en commun le recours à des ressources linguistiques (plus ou moins fines) .

- Les approches **statistiques** n'utilisent au contraire que peu ou pas d'informations linguistiques explicites, mais construisent leurs modèles par apprentissage automatique à partir de données contenues dans des *corpus*. Ceux-ci sont étiquetés manuellement (par des linguistes), et le système est ensuité"entraîné" (de manière supervisée ou non-supervisée) à analyser des textes. La disambiguation des unités lexicales s'opère en déterminant la plus probable des interprétations possibles. Ces algorithmes sont dit "robustes" car ils sont supposés capables de fonctionner (avec plus ou moins de succès !) sur n'importe quel texte.

Notons que des approches hybrides existent également et semblent avoir quelque succès [Klavans 96]. Un critère de mesure important pour l'évaluation de ces approches,du point de vue de la qualité de leur analyse syntaxique, est la notion de *couverture* [Jurafsky 00] : il s'agit du pourcentage d'analyse (selon des codes de catégories grammaticales préalablement définis) d'unités lexicales "correctes" par rapport au jugement humain sur celles-ci. Les meilleurs "taggeurs" atteignent pour l'instant des pourcentages de couvertures d'environ 96-97 % pour des codes grammaticaux simples. Chiffre à prendre avec des pincettes, bien sûr, vu le nombre de facteurs externes pouvant influencer ce résultat.

Dans le cadre de ce travail, nous nous inscrirons dans une méthodologie appartenant à la première catégorie : le *lexique-grammaire*, et nous pencherons sur un formalisme simple permettant de décrire avec précision certains phénomènes linguistiques : les **grammaires locales**.

1.2 Le lexique-grammaire

Le **lexique-grammaire** est "une approche formelle, transformationnelle et empirique de la linguistique qui met en avant le caractère fondamental du lexique. L'objectif est de recenser exhaustivement et systématiquement l'ensemble des comportements syntaxiques des phrases simples" [Constant 03].

Cette méthode, élaborée à partir des années 70 au LADL (Laboratoire d'Automatique Documentaire et Linguistique, CNRS) par Maurice Gross et son équipe, met donc l'accent sur la nécessité d'établir des *inventaires* descriptifs systématiques des faits linguistiques, à l'opposé de la démarche chomskyienne d'élaboration d'un modèle abstrait et universel du langage, où la syntaxe se pose comme entièrement autonome de la lexicologie. Les recherches de M. Gross ont en effet démontré l'irrégularité de nombreux phénomènes linguistiques, et donc l'impossibilité de généraliser naïvement ceux-ci[1] [Gross 75].

[1]On peut se convaincre intuitivement des problèmes liés à une telle formalisation en examinant le résultat des règles transformationnelles (passivation, pronominalisation, négation,...) appliquées à certaines phrases pourtant très simples (l'exemple provient de [Watrin 03]) :

1) Ce problème concerne Luc. $\Rightarrow_{passivation}$ Luc est concerné par ce problème
2) Ce problème regarde Luc. $\Rightarrow_{passivation}$ *Luc est regardé par ce problème

Un des travaux les plus connus de Maurice Gross concerne l'étude de **la syntaxe du verbe**. En répertoriant les comportement syntaxiques (sujet, nombre et types de compléments admis,...) de 5.000 verbes simples du français, il obtient environ 15.000 emplois différents, qu'il encode dans des grandes matrices (les fameuses tables du lexique-grammaire), regroupés selon leur structure définitionnelle. Les lignes de ces matrices sont les emplois de verbes, et les colonnes les propriétés syntaxiques (ex : $N_0 V entre N_1 et N_2$). Un signe '+' indique que l'unité lexicale accepte cette propriété, et '-' indique qu'elle ne l'accepte pas.

N1 V	N0 en mouvement	Aux = : avoir	entrée	N1 = : Nhum	N0 (ni) V N1pc	N0 V N1 Loc N1pc	N0 V N0pc	N1 = : N–hum	N1 = : le fait que P	N1 = : V–n	Ppv = : le	N0 V N1	N0 V N1 c par N1pc	N0 V N1 c sur N1pc	N0 V N1 c à N1pc	N0 V N1 de V–n	N0 V N1 de comp de V–n	V–n instrument	N0 V N1 de N2	N1 est Vpp W	V = V(+)	Exemple
-	+	+	accrocher	-	+	+	-	+	-	-	+	+	+	-	+	-	-	-	-	-	+	Max a$accroché$une branche avec son hameçon*
-	+	+	agripper	-	+	+	+	-	+	+	+	+	+	-	+	+	+	+	+	-	+	Paul a$agrippé$le bras de Marie
-	-	+	ausculter	-	+	+	-	-	-	-	+	+	-	+	+	-	-	-	-	-	-	Max a$ausculté$les bronches de Luc
-	-	+	baffer	-	+	+	-	-	-	-	+	+	-	-	+	+	-	-	+	-	-	Max a$baffé$Léa
-	-	+	baiser	+	+	+	-	+	-	-	+	+	-	+	+	-	-	-	-	-	-	Max$baise$la main d'Ida*
-	-	+	battre	-	-	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	Le sabrebatles cuisses de Luc
-	-	+	bécoter	-	+	+	-	+	-	-	+	+	-	-	+	+	-	-	+	-	-	Max$bécote$les joues de Léa
-	-	+	biser	-	+	+	-	-	-	-	+	+	-	+	+	-	-	-	-	-	-	Max a$bisé$Ida
-	-	+	bouchonner	-	+	-	-	-	-	-	+	+	-	+	-	-	+	+	-	-	-	Max a$bouchonné$son cheval
-	-	+	boxer	-	+	+	-	+	-	-	+	+	-	-	+	-	-	-	+	-	-	Max a$boxé$Luc sur le nez
-	-	+	branler	-	+	+	-	-	-	-	+	+	-	-	+	-	-	-	-	-	-	Ida$branle$Luc*
+	-	+	branler	-	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	Max$branle$la tête*
-	-	+	brutaliser	-	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	Max$brutalise$Ida
-	-	+	caresser	-	+	+	-	+	-	-	+	+	-	+	+	+	-	-	+	-	-	Max$caresse$la peau d'Ida
-	-	+	chatouiller	+	+	+	-	-	-	-	+	+	-	+	+	-	-	-	-	-	+	Max$chatouille$Ida dans le dos
-	-	+	chevaucher	+	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	Max$chevauche$une jument*
-	-	+	cingler	-	+	+	-	-	-	-	+	+	-	-	+	-	-	-	-	-	+	Max a$cinglé$Luc(E+de trois coups de fouet)
-	-	+	claquer	-	+	+	+	-	-	-	+	+	-	+	-	+	-	-	+	-	-	Max a$claqué$Luc au visage
-	-	+	cogner	-	+	+	-	+	-	-	+	+	-	+	+	-	-	-	-	-	-	Max a$cogné$Luc
-	-	+	couper	+	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	Le vent$coupe$le visage*
-	-	+	couper	+	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	Max$coupe$Luc au doigt*
-	-	+	couronner	-	+	+	-	+	-	-	+	+	-	-	-	+	-	-	-	+	-	Sa chute a$couronné$les genoux de Max
-	-	+	cravacher	-	+	+	-	+	-	-	+	+	-	+	-	-	+	-	+	-	-	Max$cravache$son cheval
-	-	+	embrasser	-	+	+	-	-	-	-	+	+	-	+	+	-	-	-	-	-	-	Max$embrasse$Léa sur la bouche
-	-	+	empoigner	-	+	+	-	+	+	-	+	+	+	-	+	-	-	-	-	-	+	Max a$empoigné$la bouteille par le goulot
-	-	+	enlacer	-	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	Max$enlace$Ida dans ses bras
-	-	+	éperonner	-	+	+	-	+	-	-	+	+	-	-	+	-	-	+	-	-	+	Luc$éperonne$son cheval
-	-	+	épouser	-	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	-	La robe$épouse$les formes de Léa
-	-	+	étreindre	-	+	+	-	+	-	-	+	+	-	-	+	-	-	-	-	-	+	Max$étreint$Ida dans ses bras
-	-	+	étriller	-	+	+	-	-	-	-	+	+	-	-	+	-	+	+	+	-	+	Max$étrille$son cheval

FIG. 1.1 – extrait de la table 32CL du lexique-grammaire

De ces observations, Maurice Gross tire une conclusion limpide : les verbes du français possèdent des comportements syntaxiques **quasiment uniques**, et il devient donc impossible de les prédire à partir de règles générales - seule l'accumulation systématique, au sein d'un lexique, des constructions syntaxiques possibles pour un prédicat déterminé est à même de rendre compte de la réalité de la langue (et donc de permettre un traitement automatique précis de celle-ci).

De plus, il s'aperçoit également de l'importance essentielle qu'y jouent les **expressions figées** (telles que "*y aller par quatres chemins*" ou ", *"être mis à la porte"*). Il recense ainsi plus de 25.000 expressions verbales figées [Gross 84]., et 20.000 expressions en "être" et "avoir" . Il en vient à considérer la phrase simple comme **l'unité élementaire de base**, non seulement au niveau syntaxique (hypothèse déjà présente chez Z. Harris), mais également au niveau sémantique.

Toutes ces analyses ont été étendues, avec le même résultat, à d'autres catégories grammaticales : noms prédicatifs, adverbes, adjectifs,... ainsi qu'à d'autres langues (italien, espagnol, portugais, allemand, coréen, malgache, grec, hongrois, chinois,...).

Terminons ce (très rapide) survol du lexique-grammaire par deux remarques générales :
- L'approche de Maurice Gross est essentiellement *empirique* : il construit son modèle à partir d'expériences linguistiques précises et rigoureuses, en cherchant à éviter toute généralisation abusive.
- Elle ne peut évidemment être fructueuse que sur le long terme ; elle exige en effet un investissement considérable de ressources (humaines) pour l'élaboration de tels "inventaires" linguistiques.

1.3 Grammaires locales

Nous venons de parler des *expressions figées* et du rôle important qu'elles occupent dans la langue. On peut en effet observer des "distributions contraintes" sur de nombreux mots. Prenons quelques exemples :

*directeur de (compagnie + thèse + conscience + *chocolat)*
*pomme (ϵ + *verte) de terre*
*Le 13 Mars (prochain + 1995 + *ampoulé)*

Rappelons la définition des **expressions figées** de [Gross 96] :
"Unités polylexicales présentant un caractère figé définies selon deux types de contraintes : syntaxiques (liberté restreinte) et sémantique (opacité)".

Nous pouvons faire à cet égard quelques commentaires :

- Certaines expressions sont strictement figées (ex : pomme de terre) : on peut en fait les classifier comme "mots composés" sans aucun perte de généralité. Par contre, d'autres ne sont que partiellement figées : elles n'acceptent pas n'importe quel complément (on voit clairement apparaître des contraintes sémantiques), mais offrent une certaine lattitude : on pourrait ainsi parler du *directeur de la petite compagnie*, du *directeur de la thèse de doctorat*,...On les appellera *expressions semi-figées*.

- Pour certaines expressions (telles les dates, voir notamment [Maurel 90]), il paraît impossible de répertorier individuellement l'ensemble des constructions possibles, la combinatoire rend en effet le nombre de celles-ci trop important. Une représentation sous forme d'automates est bien plus efficace !

D'autres phénomènes linguistiques "locaux" gagneraient également à être traitées via des automates plutôt que par des listes d'énumération, et ce à plusieurs niveaux. Au niveau morphologique par exemple, les formes fléchies des noms et verbes peuvent être très adéquatement formulées avec des transducteurs. Idem pour le découpage du texte en phrase, via l'analyse locale de la ponctuation, des majuscules, et de quelques autres critères.

Les **grammaires locales** permettent de réaliser ce genre de traitement. Nous pouvons les définir comme **un formalisme simple** permettant d'analyser avec précision des phénomènes locaux - flexion de mots, fin de phrase, expression figées ou semi-figées,... Les grammaires locales sont aussi appelées grammaires **lexicalisées** - dans le sens où elles font appel à des ressources linguistiques telles que les dictionnaires électroniques pour leur fonctionnement.

En combinant de manière cohérente plusieurs d'entre elles, il devient possible d'effectuer une analyse syntaxique de haute précision sur de nombreux corpus. Ceci peut être utilisé, par exemple, pour la difficile tâche de *levée d'ambiguités* : en élargissant le contexte du mot problématique aux unités lexicales qui l'entourent, on peut le désambiguiser.

Il nous est impossible de discuter ici en détail de toute l'argumentation linguistique déployée par Maurice Gross pour démontrer la nécessité d'utiliser de telles grammaires. Le lecteur intéressé pourra utilement se référer à [Gross 97].

Notons aussi que [Paumier 03] a montré qu'il était possible de convertir efficacement les tables du lexique-grammaire en graphes paramétrés, offrant ainsi la possibilité d'analyser des phrases libres et d'en extraire le prédicat et les arguments. De nombreuses problèmes (l'analyse du GN, pour ne citer qu'un exemple) doivent néammoins être résolus avant d'envisager une analyse syntaxique complète et exacte de toutes les phrases du français.

2. Modélisation

Nous commençons notre discussion des grammaires locales par un rappel de quelques notions fondamentales des langages formels : grammaires, automates, transducteurs. Ensuite, nous examinons l'adéquation des modèles basés soit sur des grammaires hors-contexte, soit sur des automates finis pour l'analyse des langues naturelles, et les problèmes qui peuvent se poser dans chaque cas. Enfin, nous détaillons la solution médiane choisie pour les grammaires locales : les Réseaux de Transitions Récursifs (*RTN, Recursive Transition Networks*).

2.1 Types de grammaires formelles

Formellement, une grammaire est un quadruplet $G = (V, \Sigma, R, S)$, avec[1] :

- V est un alphabet (ensemble fini de symboles)
- $\Sigma \subseteq V$ est l'ensemble des symboles terminaux (symboles faisant partie de l'alphabet sur lequel le langage généré est défini). Bien sûr, $V - \Sigma$ est alors l'ensemble des symboles non-terminaux.
- $R \subseteq (V^+ \times V^*)$ est un ensemble fini de règles, ou productions.
- $S \in V - \Sigma$ est le symbole de départ.

A partir de cette définition, les grammaires sont habituellement divisées en 4 types (appelées "hiérarchie de Chomsky", voir notamment [Jurafsky 00] ou [Russell 03]) :

Type 0 Aucun restriction sur les règles grammaticales.

Type 1 *Grammaires sensibles au contexte*. Les règles $\alpha \rightarrow \beta$ doivent satisfaire la condition $|\alpha| \leq |\beta|$. Cela signifie intuitivement que le membre de droite doit contenir au moins autant de symboles que le membre de gauche.

Type 2 *Grammaires hors-contexte*. Toutes les règles doivent avoir la forme $A \rightarrow \beta$, où $A \in V - \Sigma$. Intuitivement, cela signifie donc qu'une grammaire est hors-contexte si le membre de gauche de chaque règle est constitué d'un seul symbole non-terminal.

Type 3 *Grammaires régulières*. Toutes les règles doivent avoir une des deux formes suivantes : $A \rightarrow wB$, ou $A \rightarrow w$, avec $A, B \in V - \Sigma$ et $w \in \Sigma*$

La relation entre ces 4 types de grammaire est bien sûr la suivante :

$$Type3 \subset Type2 \subset Type1 \subset Type0 \tag{2.1}$$

2.2 Automates et transducteurs

Un **automate fini** déterministe est défini par le quintuplet $M = (Q, \Sigma, \delta, s, F)$ où
- Q est un ensemble fini d'états
- Σ est un alphabet
- $\delta : Q \times \Sigma \rightarrow Q$ est la fonction de transition
- $s \in Q$ est l'état initial
- $F \subseteq Q$ est l'ensemble des états accepteurs

[1] Cette formalisation est tirée de [Wolper 91] et de [Grune 02]

Un automate peut donc être vu *une machine qui résout un problème*, c'est-à-dire qui reconnait un certain langage. Il peut être représenté par un *graphe* où chaque état est représenté par un sommet, et chaque relation de trensition par un arc étiqueté. Ainsi, pour tout $p, q \in Q$ et $\sigma \in \Sigma$ tel que $\delta(p, \sigma) = q$, on trouve un arc étiqueté par σ reliant les sommets p et q.

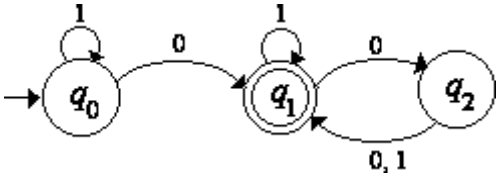

FIG. 2.1 – Exemple d'automate. Ici $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $F = \{q1\}$ et q_0 est l'état initial. Quant à la fonction de transition, elle est la suivante : $\delta(q_0, 0) = q_1$, $\delta(q_0, 1) = q_0$, $\delta(q_1, 0) = q_2$, $\delta(q_1, 1) = q_1$, $\delta(q_2, 0) = q_1$ et $\delta(q_2, 1) = q_1$

On peut facilement prouver (voir par exemple [Wolper 91]) qu'on peut faire correspondre à chaque automate fini **un langage régulier** (ie. généré par une grammaire régulière) - et inversément.

Un automate à états finis (*finite-state automaton*, ou FSA) se contente de *reconnaître* les élements d'un langage. Un **transducteur** est un automate "étendu" permettant, pour chaque étape de transition, d'également produire un symbole en sortie[2]. Plusieurs formalismes existent, prenons celui de la *machine de Mealy* :

Un transducteur à états finis est un quintuplet $M = (Q, \Sigma, \delta, s, F)$ où
- Q est un ensemble fini d'états
- Σ est un alphabet de symboles complexes. Chaque symbole est constitué d'une paire $i : o$ avec $i \in$ à un alphabet d'entrée I, et $o \in$ à un alphabet de sortie O. Donc, $\Sigma \subseteq I \times O$.
- $\delta : Q \times \Sigma \rightarrow Q$ est la fonction de transition
- $s \in Q$ est l'état initial
- $F \subseteq Q$ est l'ensemble des états accepteurs

2.3 Grammaires et récursivité

Quelle grammaire est la plus adapté au traitement syntaxique de la langue naturelle ? On peut observer expérimentalement que les transducteurs à états finis permettent de représenter adéquatement de nombreux phénomènes linguistiques : flexion, variantes phonétiques et lexicales, ambiguités,...

Dans ses travaux sur la grammaire générative, Chomsky a néammoins pointé le problème de la *récursivité* du langage. Celle-ci se produit lorsque la dérivation d'un non-terminal contient le non-terminal lui-même, comme dans $Nominal \rightarrow NominalPP$.

En 1959, il a prouvé qu'un langage hors-contexte L ne pouvait être généré par un automate fini que si (et seulement si) il existe une grammaire hors-contexte générant L dont les règles ne contiennent aucune récursion *centrée* (de type $A \rightarrow \alpha A \beta$).

La pertinence de l'argument de la récursivité est discutable. Ainsi, on peut facilement observer qu'une phrase utilisant plus d'une ou deux imbrications devient rapidement incompréhensible :

(?) La femme dont le portefeuille qui était fourni a été volé est triste.*

Pratiquement, le langage ne serait donc pas infini.

[2]Prenons l'exemple de l'analyse morphologique : pour chaque mot reconnu par le transducteur, celui-ci produirait alors en sortie le lemme et la catégorie grammaticale du mot.

2.4 Réseaux de transition récursifs

La solution utilisée par [Paumier 03] pour régler le problème de la récursivité est l'utilisation d'un modèle qui ressemble à un modèle à états-finis, mais qui est isomorphiquement équivalent à une grammaire hors-contexte. Ce modèle s'appelle le réseau de transition récursif (RTN), inventé par [Woods 70]. Un RTN est défini par ensemble de graphes semblables à ceux d'un automate fini, où chaque arc contient un noeud terminal ou non-terminal.

La différence par rapport à un FSA se situe au niveau du traitement des non-terminaux : le RTN traite chaque non-terminal comme une sous-routine. Cette approche permet notamment de supprimer les appels récursifs se situant à un "trop grand" (ie. humainement incompréhensible) niveau de profondeur.

Les RTN sont donc utilisés, pour prendre une expression de [Jurafsky 00], comme une sorte de "métaphore graphique" de grammaires hors-contexte.

Comme nous l'avons fait pour les FSA, nous pouvons également étendre les RTN pour obtenir la possibilité de produire des symboles en sortie. Nous les appelerons des **transducteurs RTN**.

On peut les définir formellement par un n-uplet : $M = (Q, I, \Sigma, \delta, s, F)$ où[3]
- Q est un ensemble fini d'états
- I est l'ensemble des états sous-initiaux (état qui étiquette au moins une transition du transducteur RTN, et représente donc un appel récursif au sous-RTN)
- Σ est un alphabet de symboles complexes. Chaque symbole est constitué d'une paire $i : o$ avec $i \in$ à un alphabet d'entrée I, et $o \in$ à un alphabet de sortie O. Donc, $\Sigma \subseteq I \times O$.
- $\delta : Q \times (\Sigma \cup I \cup \{\epsilon\}) \rightarrow Q$ est la fonction de transition
- $s \in Q$ est l'état initial
- $F \subseteq Q$ est l'ensemble des états accepteurs

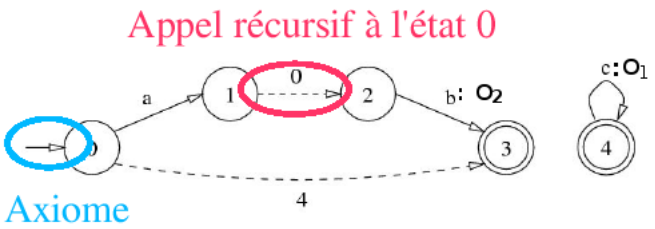

FIG. 2.2 – Un transducteur RTN. Il est défini par $Q = \{0, 1, 2, 3, 4\}$, $I = \{0, 4\}$, $\Sigma = \{(a, \epsilon), (b, O_2), (c, O_1)\}$, $I = \{0, 4\}$, $F = \{3, 4\}$ et $S = \{0\}$. Quelques exemples de séquences reconnues et transformées : $ab \rightarrow O_2$ $accb \rightarrow O_1O_1O_2$ $c \rightarrow O_1$ ...

Notons que l'analyse descendante des RTN n'est garantie qu'à deux conditions :
- La grammaire ne doit pas être récursive à gauche
- La grammaire ne doit pas comporter de boucle sur le mot vide ϵ

Dans `Unitex` (voir chapitre suivant), ces deux conditions sont vérifiées à chaque passe, et signalées/corrigées le cas échéant.

[3] Formalisme extrait de [Fairon 04]

3. Implémentation

Un système de construction et d'analyse des grammaires locales - sous forme de RTN - a été intégré dans `Unitex`. Il est inutile de décrire ici l'ensemble des fonctionnalités de ce logiciel, nous vous renvoyons à [Paumier 04]. Contentons-nous d'en dégager l'essentiel.

`Unitex` est un logiciel de **traitement de corpus** utilisant trois types de ressources linguistiques :

- Des *dictionnaires électroniques* des mots simples et composés, associant à chaque entrée (1) un lemme et (2) des codes grammaticaux, sémantiques et flexionnels. Le formalisme DELA est utilisé.

- Des grammaires, sous la forme de RTN. Celles-ci peuvent être construites et modifiées graphiquement par un éditeur de graphes. Ces grammaires peuvent alors être compilées et appliquées à des larges données textuelles pour l'établissement de condordances.

- Des tables du lexique-grammaire, sous la forme de matrices (cfr notre introduction). La particularité d'`Unitex` est de pouvoir générer automatiquement des graphes paramétrés à partir de ces tables.

Logiciel libre sous licence GPL, il est téléchargeable à l'adresse :
http ://www-igm.univ-mlv.fr/~unitex/

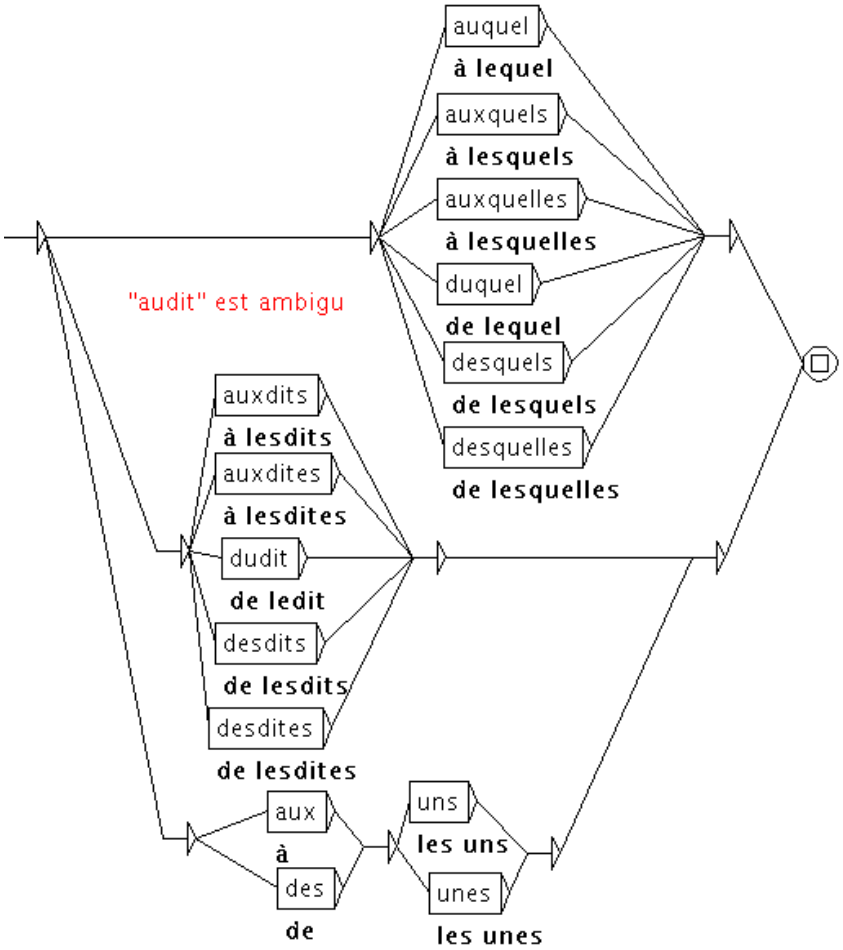

FIG. 3.1 – Exemple de graphe `Unitex`, utilisé pour le prétraitement de textes.

4. Applications

On peut trouver dans [Constant 03] un aperçu des applications développées par le réseau RELEX sur le formalisme des grammaires locales, que nous résumons ici :

- *Découpage du texte en phrase*, via l'analyse de la ponctuation, des majuscules, et de quelques autres critères locaux.
- *Génération automatique de formes fléchies* des noms, verbes, adjectifs,... via des règles morphologiques simples.
- *étiquetage lexical* de texte sur base de dictionnaires électroniques DELA.
- *extraction d'information*, tels que l'extraction de noms propres ou la création de patrons d'extraction - voir [Watrin 03].
- *filtrage d'information* sur la base de critères préalablement définis.
- *Levée d'ambiguité* via l'utilisation de batteries de transducteurs (voir le système `ELAG` intégré dans `Unitex`).
- *Traduction automatique* : vu l'extraordinaire difficulté de la tâche, les recherches actuelles s'orientent plutôt vers des systèmes d'aide à la traduction - voir par exemple [Fairon 99].

4.1 Conversion des tables du lexique-grammaire

Terminons ce travail par l'examen du **système de conversion** des tables du lexique-grammaire en grammaires locales, intégré à `Unitex`. L'implémentation de celui-ci est détaillée dans [Paumier 03] et dans [Paumier 04] :

"La conversion d'une table en graphes s'effectue au moyen du mécanisme des graphes paramétrés. Le principe est le suivant : on construit un graphe qui décrit des constructions possibles. Ce graphe fait référence aux colonnes de la table grâce à des variables. On génère ensuite pour chaque ligne de la tableune copie de ce graphe dans laquelle les variables sont remplacées en fonction du contenu des cellules situées à l'intersection des colonnes correspondantes et de la ligne traitée. Si une cellule de la table contient le signe +, la variable coresspondante est remplacée par <E>. Si la cellule contient le signe -, la boîte contenant la variable correspondante est supprimée, ce qui détruit du même coup les chemins passant par cette boîte. Dans tous les autres cas, la variable est remplacée par le contenu de la cellule."

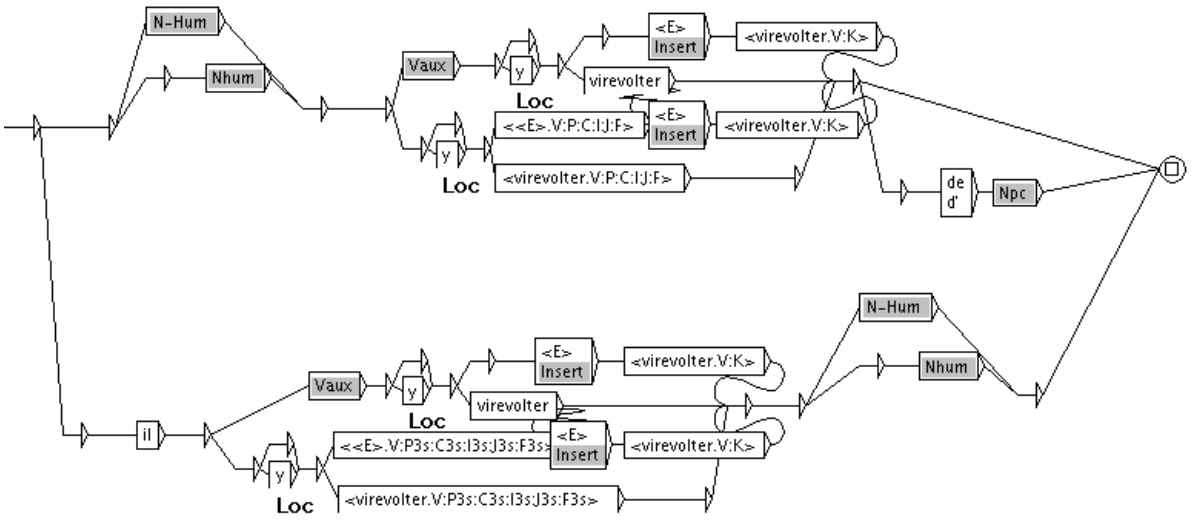

FIG. 4.1 – Exemple de sous-graphe généré `Unitex` pour la table 31H

Bibliographie

[Constant 03] Matthieu Constant. *Grammaires locales pour l'analyse automatique de textes*. PhD thesis, Université de Marne-la-Vallée, 2003.

[Fairon 99] C Fairon & J. senellart. *Classes d'expressions bilingues gérées par des transducteurs à états finis, dates et titres de personnalité*. Linguistique contrastive et traduction, 1999.

[Fairon 04] Cédrick Fairon, Anne Dister & Sébastien Paumier. Notes et transparents du cours "introduction au traitement des langues naturelles". Université Catholique de Louvain, 2004.

[Gross 75] Maurice Gross. Méthodes en syntaxe. Hermann, 1975.

[Gross 84] Maurice Gross. *Une classification des phrases figées du français*. In C. Muller P. Attal, editeur, De la syntaxe à la pragmatique. John Benjamins, Amsterdam, 1984.

[Gross 96] Gaston Gross. Les expressions figées en français : noms composés et autres locutions. Collection l'essentiel français. Ophrys, 1996.

[Gross 97] Maurice Gross. *The construction of local grammars*. In E. Roche & Y. Schabès, editeurs, Finite-State Language Processing, Language, Speech, and Communication, chapitre 11, pages 329–354. MIT Press, 1997.

[Grune 02] Dick Grune, Henri E. Bal, Ceriel J.H. Jacobs & Koen G. Langendoen. Compilateurs. Dunod, 2002.

[Jurafsky 00] Daniel Jurafsky & James H. Martin. Speech and language processing. Prentice Hall, 2000.

[Klavans 96] Judith Klavans & Philip Resnik. The balancing act : Combining symbolic and statistical approaches to language. MIT Press, 1996.

[Maurel 90] D. Maurel. *Adverbes de date : étude préliminaire à leur traitement automatique*. Linguisticae Investigationes, vol. 14, no. 1, pages 31–63, 1990.

[Paumier 03] Sébastien Paumier. *De la reconnaissance des formes linguistiques à l'analyse syntaxique*. PhD thesis, Université de Marne-la-Vallée, 2003.

[Paumier 04] Sébastien Paumier. *Unitex 1.2 - Manuel d'utilisation*, 2004.

[Russell 03] Stuart Russell & Peter Norvig. Artificial intelligence : A modern approach. Prentice Hall, 2003.

[Watrin 03] Patrick Watrin. Entre lexique et syntaxe : vers la création de patrons d'extraction. Master's thesis, Université Catholique de Louvain, 2003.

[Wolper 91] Pierre Wolper. Introduction à la calculabilité. Dunod, 1991.

[Woods 70] W. A. Woods. *Transitive network grammars for natural language analysis*. Communications of the ACM, vol. 13, pages 591–606, 1970.