Modèles de Langage et Analyse Syntaxique

Cours 2 - Syntaxe

Antoine Rozenknop
antoine.rozenknop@lipn.univ-paris13.fr

7 octobre 2010

Plan

1 Introduction

Dans les exemples vus jusqu'ici on s'est limité à des règles de réécritures qui avaient toutes la forme :

$$\mathrm{X} \rightarrow \mathrm{Y}_1 \dots \mathrm{Y}_n$$

où X est un symbole *non-terminal* de la grammaire, et $\mathrm{Y}_1 \dots \mathrm{Y}_n$ sont soit des symboles non-terminaux, soit des symboles terminaux.

Il s'agit en fait d'un type bien particulier de grammaires de réécriture, appelées *grammaires hors-contexte*, pour lequel on maîtrise des algorithmes d'analyse efficaces, comme on va le voir (plus tard !). Il en existe d'autres types.

Le pionnier en matière de grammaire de réécritures est Noam Chomsky. Dans la suite, on va voir :
- la classification des grammaires formelles issue de ses travaux ;
- ce qu'est la *forme normale de Chomsky* ;
- comment on peut transformer une grammaire hors-contexte quelconque pour obtenir une grammaire équivalente sous forme normale de Chomsky.

L'intérêt de cette transformation, qui n'est pas évident au premier abord, apparaîtra clairement lorsqu'on s'intéressera aux algorithmes d'analyse syntaxique des grammaires hors-contexte. Mais chaque chose en son temps !

2 Grammaires formelles

2.1 Définitions :

Grammaire formelle
- **V** : Vocabulaire
- $\mathbf{V}_T$: Vocabulaire terminal (ensemble de symboles terminaux)
- $\mathbf{V}_N$: Vocabulaire non-terminlal (ensemble de symboles non-terminaux)
- **P** : Axiome (élément de V_N) (un symbole non-terminal)
- **R** : ensemble de règles de récritures de la forme : $\alpha \rightarrow \beta$ avec
  - $\alpha, \beta \in \mathrm{V}^*$
  - $\alpha \neq \emptyset$

Définition 1. *Une grammaire formelle* ***G*** *est un quadruplet (V_N, V_T, R, P).*

Dérivation

Définition 2. *Une chaîne u_1* ***se réécrit*** *en une chaîne u_2 ($u_1 \Rightarrow u_2$) si et seulement si il existe des chaînes de symboles v_1, v_2, α, β tels que :*

1. $u_1 = v_1\ \alpha\ v_2$
   $u_2 = v_1\ \beta\ v_2$
2. $\alpha = \beta$ *ou* $(\alpha \rightarrow \beta) \in R$

Définition 3. *Une chaîne u_1* ***se dérive*** *en une chaîne u_2 ($u_1 \Rightarrow^* u_2$) si une* ***succession*** *de réécritures permet d'obtenir u_2 à partir de u_1.*

Définition 4. *Une dérivation est une succession de réécritures.*

Langage

Définition 5. *On appelle* ***langage engendré par G*** *l'ensemble de toutes les suites de symboles qui dérivent de l'axiome de G :*

$$\boldsymbol{L(G)}=\{x/x \in V^*\ et\ P \Rightarrow^* x\}$$

Définition 6. *On appelle* ***langage sur V_T*** *l'ensemble (potentiellement infini) des chaînes de* ***longueurs finies*** *formées avec des éléments du vocabulaire terminal V_T.*

Décidabilité

Définition 7. *Un langage est* ***décidable*** *si pour toute phrase on peut savoir au bout d'un* ***temps fini*** *si elle appartient ou nom au langage.*

2.2 Hiérarchie des grammaires formelles

La hiérarchie des grammaires formelles décrite par N. Chomsky classe les grammaires de réécriture selon la forme que peut prendre leurs règles. On les donne en ordre inverse, c'est-à-dire de la forme la plus restrictive à la moins restrictive.

2.2.1 type 4

Dans une grammaire de type 4, **les parties droites de toutes les règles sont des terminaux**. Les éléments de R ont donc la forme :

$$X \rightarrow \alpha \text{ avec } \begin{array}{l} \mathrm{X} \in \mathrm{V}_N \\ \alpha \in \mathrm{V}_T{}^* \end{array}$$

Une telle grammaire ne fait qu'**énumérer** les phrases de son langage sur V_T.

2.2.2 type 3 : grammaires régulières = grammaires rationnelles

Définition 8. *Dans une grammaire* ***régulière à gauche****, les règles ont l'une des formes suivantes :*

$$\left.\begin{array}{lcl} X & \rightarrow & Y\ a \\ X & \rightarrow & a \end{array}\right\} \ avec \left\{\begin{array}{l} X, Y \in V_N \\ a \in V_T \end{array}\right.$$

Définition 9. *Dans une grammaire* ***régulière à droite****, les règles ont l'une des formes suivantes :*

$$\left.\begin{array}{lcl} X & \rightarrow & a\ Y \\ X & \rightarrow & a \end{array}\right\} \ avec \left\{\begin{array}{l} X, Y \in V_N \\ a \in V_T \end{array}\right.$$

Exemple de grammaire régulière à droite

$$\begin{array}{lcl} P & \rightarrow & a\ P \\ B & \rightarrow & b \end{array} \qquad \begin{array}{lcl} P & \rightarrow & B \\ B & \rightarrow & b\ B \end{array}$$

Cette grammaire engendre les chaînes $a^n b^m$.

ab aab abb aaabbbbbb ...

Utilité en TAL

Les grammaires régulières sont utilisées dans :
- la représentation compacte des **lexiques**
- la construction de **correcteurs orthographiques** ou de lexiques **robustes** aux erreurs
- des **grammaires locales** : mots-composés, séquences acceptables de chiffres (nombres,dates)
- des grammaires pour des **domaines très restreints** (annonces de vols dans les aéroports)

Limitations

Les grammaires de type 3 ne peuvent traiter :
- les chaînes $a^n b^n$ de type $a_1 \dots a_n b_1 \dots b_n$ ("respectivement")
- les chaînes $a^n b^n$ de type $a_1 \dots a_n b_n \dots b_1$ (expressions **parenthésées**, structures enchâssées) :

  En anglais :

  The dog the stick the fire burned beat bit the cat

  En français :

  Le chat que le voisin que le maire que le préfet qui a été condamné a félicité a attrapé est blanc.

- les chaînes *abac*

  soit ... soit ... ni ... ni ...

- le **rejet** en fin de phrase des prépositions (en anglais) ou des particules séparables (en allemand) :

  The girl that I do not want to be caught with.

- Les **dépendances à longue distance** (interrogatives, clivées...)

  Jean veut savoir quelle fille Marie croit que Paul a vue.

2.2.3 type 2 : grammaires hors-contexte = grammaires algébriques

Définition 10. *Une grammaire de type 2, dite* ***hors-contexte*** *ou* ***algébrique****, est une grammaire de réécriture dont les parties gauches des règles contiennent un unique non-terminal :*

$$X \rightarrow \alpha \ \textit{avec} \ \begin{cases} X \in V_N \\ \alpha \in V^* \end{cases}$$

Exemple de grammaire hors-contexte

$$\begin{array}{lcl} P & \rightarrow & a\ P\ b \\ P & \rightarrow & a\ b \\ P & \rightarrow & \end{array}$$

Cette grammaire engendre les chaînes $a^n b^n$.

"" "ab" "aabb" "aaabbb" ...

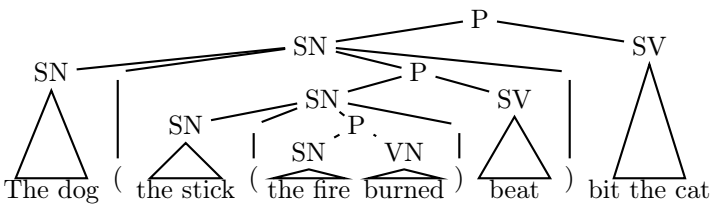
Utilité en TAL

Les grammaires hors-contexte sont adaptées pour :

– les chaînes $a^n b^n$ de type $a_1 \dots a_n b_n \dots b_1$ (expressions parenthésées, structures enchâssées)

P → SN SV
SN → SN (P) peut analyser :

The dog the stick the fire burned beat bit the cat



– les chaînes *abac*

soit ... soit ... (X →soit Y soit Y) ni ... ni ... (X →ni Y ni Y)

Limitations

– Les grammaires hors-contexte traitent avec difficulté le **rejet** en fin de phrase des prépositions (en anglais) ou des particules séparables (en allemand).

Solution : dupliquer les règles pour chaque préposition ou particule

– Elles ne peuvent traiter les **dépendances à longue distance** (interrogatives, clivées...) , le problème étant par exemple de contraindre un accord selon un syntagme qui sort du contexte de cet accord :

Jean veut savoir quelle fille Marie croit que Paul a vue.

– Ni les rares langues qui ont des structures de type $a^n b^m c^n d^m$
– Ni les chaînes $a^n b^n$ de type $a_1 \dots a_n b_1 \dots b_n$ ("respectivement")

Henri et Sophie sont repectivement indifférent et séduite par le film.

Ces arguments ne sont pas vraiment décisifs pour mettre de côté ces grammaires pour le TAL, car en pratique, les ^{n} et ^{m} ne sont jamais grands.

2.2.4 type 1 : grammaires contextuelles

Définition 11. *Les grammaires de type 1, dites* ***grammaires contextuelles*** *se définissent par des règles du type :*

$$\alpha \to \beta \ \textit{avec} \begin{cases} \alpha \in V^+ \\ \beta \in V^* \\ |\beta| \geq |\alpha| \end{cases}$$

(La partie gauche est non vide, et la partie droite doit contenir plus de symboles que la partie gauche)

Exemple :

La grammaire suivante engendre le langage $a^n b^n c^n$

P →a B C	C B →B C	a B →a b
P →a P B C		b B →b b
		b C →b c
		c C →c c

Remarques

- Ces grammaires sont **décidables**

  le nombre de symboles ne peut que croître dans une dérivation.

  Pour déterminer si une phrase de longueur n appartient au langage, il suffit de produire toutes les dérivations en s'arrêtant dès que le nombre de symboles produits dépasse n, ce qui se fait en un temps fini. Cependant, un temps fini ne veut pas dire qu'il soit court ! En pratique, cette génération exhaustive a une complexité exponentielle en n (le temps d'analyse est proportionnel à l'exponentielle du nombre de mots de la phrase à analyser).
- Les grammaires sur lesquelles portent la majorité des recherches en TAL se situent entre le type 1 et le type 2.

2.2.5 type 0 : grammaire non-contrainte

Définition 12. *Les grammaires de type 0 se définissent par des règles du type :*

$$\alpha \rightarrow \beta \ avec \begin{cases} \alpha \in V^* \\ \beta \in V^* \end{cases}$$

Remarque : **Ces grammaires ne sont pas décidables.**

On a pu remarquer que toutes les grammaires qui ont été écrites pour des langues naturelles pouvaient être récrites en utilisant le formalisme des règles contextuelles.

3 Forme Normale de Chomsky

Noam Chomsky a défini un type particulier de grammaires hors-contexte (type 2) : les grammaires hors-contexte sous **forme normale**. En TAL, ces grammaires sont très pratiques pour faire de l'analyse syntaxique : elles permettent l'utilisation d'algorithmes efficaces pour cette tâche, de complexité $O(n^3)$.

Forme Normale de Chomsky (CNF)

Définition 13. *Une grammaire hors-contexte est sous **Forme Normale de Chomsky** si ses règles ont l'une des deux formes :*

$$\left.\begin{array}{lcl} X & \rightarrow & Y\ Z \\ X & \rightarrow & a \end{array}\right\} \ avec \begin{cases} X \in V_N \\ Y \in V_N \\ Z \in V_N \\ a \in V_T \end{cases}$$

Définition 14. *Une grammaire hors-contexte est sous **CNF étendue** si ses règles peuvent également prendre les formes :*

$$\left.\begin{array}{lcl} X & \rightarrow & Y\ Z \\ X & \rightarrow & Y \\ X & \rightarrow & a \end{array}\right\} \ avec \begin{cases} X \in V_N \\ Y \in V_N \\ Z \in V_N \\ a \in V_T \end{cases}$$

Mise sous forme normale

La plupart des grammaires hors-contexte ne sont pas sous forme normale de Chomsky, que ce soient des grammaires du langage naturel ou de langages artificiels (langages de programmation par exemple), ce qui pourrait limiter l'intérêt des algorithmes efficaces que nous verrons plus loin.

Heureusement, les grammaires de type 2 ont la propriété intéressante de pouvoir être mises sous forme normale, c'est-à-dire qu'il existe toujours une grammaire CNF équivalente.

Définition 15. *Deux grammaire sont dites* ***équivalentes*** *si elles peuvent produire les mêmes chaînes de symboles terminaux.*

Dans le cas des grammaires hors-contexte, on peut de plus facilement trouver un arbre d'analyse à partir d'un arbre créé avec la grammaire CNF équivalente.

La mise sous forme normale se fait en
3 temps :

1. Suppression des règles de type : $X \to \alpha\ t_i\ \beta$ (où t_i est un terminal et α et/ou β sont non vides)
   (a) Créer un non-terminal T_i
   (b) Ajouter la règle $T_i \to t_i$
   (c) Remplacer la règle $X \to \alpha\ t_i\ \beta$ par $X \to \alpha\ T_i\ \beta$
2. Suppression des règles de type : $X \to Y$
   (a) Pour chaque règle $Z \to \alpha\ X\ \beta$, ajouter une règle $Z \to \alpha\ Y\ \beta$.
   (b) Supprimer $X \to Y$.
3. Suppression des règles de type : $X \to Y\ Z\ \alpha$
   (a) Créer un nouveau non-terminal X_i
   (b) Ajouter la règle $X_i \to Z\ \alpha$
   (c) Remplacer la règle $X \to Y\ Z\ \alpha$ par $X \to Y\ X_i$

La mise sous forme normale d'une grammaire
augmente considérablement le nombre de non-terminaux et de règles.

Exemple :

Forme initiale				Forme normale de Chomsky			
$\mathbf{R}_1$:	P	→	SN SV	$\mathbf{R}_1$:	P	→	SN SV
$\mathbf{R}_2$:	SN	→	Det N	$\mathbf{R}_2$:	SN	→	Det N
$\mathbf{R}_3$:	SN	→	Det N SP	$\mathbf{R}_{3.1}$:	X_1	→	N SP
				$\mathbf{R}_{3.2}$:	SN	→	Det X_1
$\mathbf{R}_4$:	SP	→	Prep SN	$\mathbf{R}_4$:	SP	→	Prep SN
$\mathbf{R}_5$:	SV	→	V	$\mathbf{R}_{1.2}$:	P	→	SN V
$\mathbf{R}_6$:	SV	→	V SN	$\mathbf{R}_6$:	SV	→	V SN
$\mathbf{R}_7$:	SV	→	V SN SP	$\mathbf{R}_{7.2}$:	X_2	→	SN SP
				$\mathbf{R}_{7.1}$:	SV	→	V X_2
$\mathbf{L}_5$:	V	→	mange	$\mathbf{L}_5$:	V	→	mange