

Bases formelles du TAL : Mini-Projet « Grammaires propres »
Distribué le 23 mars 2012,
Retour le 31 mai 2012

L'objectif de ce mini-projet est d'implémenter l'ensemble des algorithmes qui permettent de rendre « propre » une grammaire algébrique quelconque.

On rappelle qu'une grammaire propre est une grammaire

- ε -libre,
- dépourvue de symboles inutiles,
- sans cycle.

On admettra qu'au lieu de débarrasser la grammaire de ses cycles éventuels, on se contente de supprimer toutes les productions singulières, ce qui garantit l'absence de cycles.

Les grammaires données en entrée seront codées dans un fichier texte, comprenant une règle par ligne, sous la forme de deux mots séparés par un espace, ε étant représenté par 0. Par exemple, la grammaire donnée ci-dessous sera représentée par le fichier suivant. Conventionnellement, les non-terminaux seront des lettres majuscules, les terminaux des minuscules. On posera que le premier symbole est l'axiome.

$A \rightarrow Ab \mid bAA \mid \varepsilon$	$A \sqcup Ab$
$B \rightarrow abBA$	$A \sqcup bAA$
	$A \sqcup 0$
	$B \sqcup abBA$

Le programme, une fois chargée la grammaire en mémoire, indiquera les nettoyages effectués, et affichera d'une manière lisible la grammaire résultat.

On demande de fournir :

- le code complet, sous la forme d'un exécutable auto-suffisant (jar ou script python selon le choix) ;
- un jeu d'essai, comprenant une dizaine de cas de grammaires permettant de tester le programme ;
- un mode d'emploi ;
- une trace complète de l'application du programme sur le jeu de test, accompagnée d'éventuels précisions ou commentaires, au format **pdf**.