TP4 - Analyse syntaxique

Programmation pour le TAL - M1 SCA & SDL

12 avril 2013

Pour chacun des exercices, sauvegardez le script Python dans un fichier qui a comme nom *exo < numéro de l'exercice >.py* puis mettre tous les fichiers dans une archive .zip qui a comme nom *TP4_<nom de l'étudiant>_<prénom de l'étudiant >.zip*. A la fin de la séance, envoyez l'archive en fichier attaché à l'adresse *perrier@loria.fr* en mettant comme objet du message *programmation pour le TAL : TP4*.

1 Arbre syntaxique

On veut créer une classe *ArbreSyntaxique* qui permette de manipuler des arbres syntaxiques dont les feuilles sont étiquetées par des mots et les nœuds sont étiquetés par des catégories syntaxiques prises dans la liste suivante :

CL	complétive (proposition objet introduite par une conjonction de subordination)
CONJ	conjonction de subordination
DET	déterminant
N	nom commun
NP	groupe nominal
PP	groupe prépositionnel
PREP	préposition
RELPRO	pronom relatif
REL	proposition relative
VINTR	verbe sans complément
VTR1	verbe à complément d'objet direct nominal seulement
VTR2	verbe à complément d'objet direct nominal et complément d'attribution
VC	verbe transitif à complément propositionnel
VP	groupe verbal (le verbe et ses compléments)
S	phrase

La classe *ArbreSyntaxique* a deux attributs : *label* qui est une chaîne de caractères représentant l'étiquette de la racine de l'arbre et *fils* qui est une liste éventuellement vide d'objets de la classe *ArbreSyntaxique* qui sont les sous-arbres fils de la racine.

Elle a deux méthodes : *afficher* qui permet d'afficher l'arbre syntaxique et *projection* qui donne la liste des mots étiquettes des feuilles de l'arbre syntaxique.

Exercice 1.1 *Définissez la classe* ArbreSyntaxique *sans utiliser la bibliothèque NLTK et appliquez la pour créer et afficher les différents arbres syntaxiques des phrases suivantes :*

Jean connaît la patronne.
Jean demande la voiture à la patronne.
Jean croit que la fille que Jean connaît travaille.

2 Grammaires algébriques

Nous voulons créer une classe *Grammaire* dont les objets sont des grammaires algébriques. Cette classe a deux attributs : *axiome* qui a comme valeur une chaîne de caractères représentant le symbole de départ de la grammaire, *regles* qui a comme valeur un dictionnaire représentant les règles de la grammaire.

Chaque entrée du dictionnaire représente un ensemble de règles ayant la même partie gauche, qui constitue la clé de cette entrée. La valeur est une liste de listes représentant les parties droites des règles. Par exemple, les règles $S \rightarrow NP\ VP \mid S\ PP$ sont représentées par le dictionnaire $\{S : [['NP','VP'],['S','PP']]\}$

La classe *Grammaire* a une méthode *afficher* qui permet d'afficher la grammaire Elle a deux autres méthodes : *decomposer* qui prend comme argument une catégorie grammaticale C et qui retourne la liste de toutes les parties droites de règles de la grammaires ayant pour partie gauche C et *composer* qui prend comme argument une liste L de symboles et retourne une liste de non terminaux C tel que C $\rightarrow$ L est une règle de la grammaire.

Exercice 2.1 *Définissez la classe* Grammaire *et appliquez la à la création d'une grammaire qui permette d'analyser les phrases de l'exercice 1.1. Affichez cette grammaire à l'aide de la méthode* afficher *. Recherchez toutes les règles qui ont comme partie gauche NP à l'aide de la méthode* decomposer*. Recherchez toutes les règles qui ont comme partie droite NP PP à l'aide de la méthode* composer*.*

3 Analyse syntaxique

Nous allons construire un analyseur syntaxique de type *shift-reduce* utilisant une grammaire algébrique. Le principe est le suivant : on lit une phrase de gauche à droite ; lorsqu'on lit un mot, on le met au sommet d'une pile (shift) ; si à un moment donné, apparaît sur la pile la partie droite d'une règle, on la remplace par la partie gauche de la même règle (reduce). Si plusieurs règles peuvent s'appliquer, on choisira la première venue. L'analyse réussit si on atteint la fin de la phrase avec comme seul symbole dans la pile le symbole de départ de la grammaire.

Exercice 3.1 *Ajoutez à la classe* Grammaire *une méthode* reconnaitre *qui prenne en argument une phrase sous forme d'une liste de mots et qui retourne un booléen indiquant si la phrase a pu être analysée avec l'algorithme shift-reduce.*

Appliquez cettte méthode à l'analyse des trois phrases de l'exercice 1.1