Contrôle continu de travaux pratiques
Sujets de projet

Les projets devront fonctionner sous Linux en utilisant des logiciels libres ou téléchargeables librement. L'archive contenant les programmes, un manuel d'utilisation, un jeu de test et un petit rapport de conception seront à envoyer au plus tard le vendredi 20 décembre 2013 à minuit à l'adresse électronique `Denis.Bechet@univ-nantes.fr`.

Projets sur la lexicalisation des grammaires algébriques

Sujet 1

Ce projet consiste à écrire un programme permettant de transformer une grammaire algébrique quelconque en une grammaire lexicalisée. On proposera plusieurs transformations :

- forme normale de Greibach,
- forme normale de Greibach double,
- notation polonaise inversée (opérateurs à droite)

Travail facultatif : généraliser ce type de transformations en proposant un algorithme qui choisit, suivant les règles, une "ancre lexicale", l'idée étant d'attacher les règles à tel ou tel mot du lexique suivant le contexte. Ici on veut un algorithme qui, si on lui donne en entrée une grammaire lexicalisée, va retourner la même grammaire. Sinon, l'algorithme devra retourner une grammaire lexicalisée aussi proche que possible de la grammaire d'origine.

Sujet 2

Il faudra écrire un programme reconnaissant un langage décrit par une grammaire sous forme normale de Greichbach à l'aide d'un automate à pile. Les programmes à créer sont :

1. Un automate à pile déterministe prenant en entrée une grammaire sous forme normale de Greibach déterministe et une chaîne de caractères et qui indique si la chaîne appartient au langage de la grammaire.
2. Un automate à pile avec retour en arrière (donc exponentiel) permettant d'utiliser une grammaire sous forme normale de Greibach pas forcément déterministe.

Il faudra aussi retourner une analyse de la chaîne en entrée lorsqu'elle appartient au langage. Lorsque plusieurs solutions existent, on pourra afficher toutes les solutions ou les n premières.
Travail facultatif : écrire un programme simulant un automate à pile avec méthode tabulaire (à la CYK) permettant d'utiliser une grammaire sous forme normale de Greibach pas forcément déterministe mais en temps polynomial. Le programme devra vérifier si la chaîne appartient au langage et éventuellement donner une solution (un arbre de dérivation).

Projet sur les LFG

Sujet 3

Le projet consiste à écrire un analyseur LFG pour le français. Il faudra utiliser le lexique Lefff téléchargeable à l'adresse `http://alpage.inria.fr/~sagot/lefff.html` qui contient des informations sur les fonctions syntaxiques des mots. Il faudra aussi écrire une grammaire (rudimentaire) du français. Il n'est pas demandé ici d'être complet mais plutôt de présenter les structures canoniques du français : sujet/verbe/compléments pour une phrase et déterminant/adjectif/nom/adjectif pour un groupe nominal. Il faudra bien sûr écrire un analyseur tabulaire (à la CYK par exemple). L'analyseur devra indiquer si la phrase est analysable et dans ce cas fournir une analyse (c-structure et f-structure), ou toutes les analyses.
Travail facultatif : prendre en compte des structures complexes avec les relatives, les subordonnées, les infinitives, le passif, les compléments du nom, etc.

Projet sur les TAG

Projet 4

Définir un programme prenant une grammaire TAG lexicalisée et une chaîne de caractères en entrée et affichant si la chaîne est analysable ou non avec la grammaire donnée. Le gros travail va consister à définir cet algorithme en effectuant une recherche bibliographique sur les algorithmes d'analyse des TAG. Le programme sera testé sur une grammaire "jouet".
Travail facultatif : le programme fournira un ou tous les arbres syntaxiques.

Projets sur les grammaires catégorielles

Sujet 5

Définir un analyseur de grammaire catégorielle classique (avec les deux règles classiques) utilisant une méthode tabulaire (algorithme CYK). Le programme devra indiquer si une chaîne en entrée est analysable avec la grammaire et si oui, devra donner au moins une dérivation.
Travail facultatif : le programme affichera une ou toutes les dérivations sous forme graphique.

Sujet 6

Écrire un programme permettant d'apprendre une grammaire catégorielle classique à partir d'exemples de structures de dérivation (algorithme de Wojciech Buszkowski & Gerald Penn de 1990). On supposera qu'il ne peut y avoir qu'un seul type par mot. L'algorithme prendra en entrée une liste de structures de dérivation et fournira en sortie une grammaire catégorielle compatible avec les exemples. Le programme devra indiquer s'il n'est pas possible de trouver une solution au problème (les types induits pour un même mot ne sont pas unifiables).
Travail facultatif : étendre le programme en abandonnant l'hypothèse d'un seul type par mot. Il faudra proposer une solution minimisant le nombre de types produit pour un mot. On expliquera clairement comment l'algorithme d'apprentissage fonctionne.