

## Implanter un analyseur tabulaire

Eric de la Clergerie et André Bittar

11 Novembre 2008

### Introduction

Le but de ce TD est d'explorer des implantation d'analyseur tabulaire de type à charte, en surcouche de DyALog.

Le matériel pour ce TD peut être récupéré sous forme d'archive .tgz à <http://alpage.inria.fr/~clerger/PrologTAL08/td5.tgz>. Il comprend :

- un squelette d'analyseur à charte `chart.pl`
- un second squelette d'analyseur à charte `chart2.pl`
- une mini grammaire du français `french.dcg`
- une mini-segmenteur `lex2db` permettant de construire un treilli de mots à partir d'une chaîne.
- un fichier `Makefile`

### Exercice 1 : Compléter l'analyseur à charte

La première implantation se fait en DyALog, sans faire appel à ses capacités innées de gestion de tabulation. Cette implantation sera donc facilement transposable à d'autres systèmes Prolog.

Un squelette commenté d'analyseur à charte est fourni avec le fichier `chart.pl`. Il s'agit de compléter les prédicats en fonction des commentaires et des notes de cours de manière à rendre le programme fonctionnel.

Tester le programme résultant sur quelques phrases.

### Exercice 2 : Extraction de forêts partagées

Le but de cet exercice est de stocker des pointeurs arrière avec les items de manière à pouvoir extraire une forêt partagée d'analyse en fin d'analyse.

1. Associer à tout item tabulé, un *mutable* pouvant stocker des pointeurs arrières.

Les prédicats utilisables pour gérer les mutables sont

```
%% Associer la valeur Value au mutable Mutable
%% (éventuellement en créant le dit-mutable)
mutable(Mutable, Value)

%% Retourne la valeur Value
%% actuellement associé au mutable Mutable
mutable_read(Mutable, Value)
```

2. Les pointeurs arrières seront typés et de la forme
  - `call` pour les items prédictifs
  - `and(DottedItem, AnswerItem)` pour les items résultants de la combinaison d'une règle pointée avec une réponse.

- `or(Backptr1,Backptr2)` pour une disjonction de backpointers, en cas d'ambiguïtés
- 3. compléter `chart.pl` pour construire ces pointeurs arrières
- 4. écrire un prédicat d'extraction de forest

```
%% Affiche la forêt en partant d'un item réponse  
show_forest(AnswerItem) :- ...
```

### Exercice 3 : Écrire un analyseur tabulaire en style `ask/tell`

En s'inspirant du modèle de concurrence `ask/tell` pouvant être mis en place grâce au prédicat `DyA-Log '$answers'(X)`, écrire un analyseur tabulaire plus simple que celui donné dans `chart.pl`.

On pourra s'appuyer sur le squelette fourni par `chart2.pl`.