

T.P. Traitement Automatique des Langues (Retoré) **Master Info 1ère année — Université Bordeaux 1**

Il s'agit d'illustrer le cours par une grammaire de clauses définies à écrire en Sicstus Prolog. En fait Prolog permet d'écrire la DCG comme un ensemble de règles de production, mais pour comprendre le fonctionnement des grammaires en Prolog, on écrira les clauses correspondantes; du reste cette syntaxe Prolog pour les grammaires n'est qu'un confort purement syntaxique: si on liste le programme entré, ce sont les clauses et non règles de la DCG qui apparaissent. Le manuel de Sicstus Prolog est disponible sur internet à l'adresse: <http://www.sics.se/isl/sicstus/>

On souhaite modéliser par des clauses écrites en Prolog la grammaire de phrases simples avec des relatives qui sont aussi bien introduite par le pronom « qui » (dont l'antécédent est le sujet de la relative) que par le pronom « que » (dont l'antécédent est l'objet de la relative). Pierre regarde Marie qui dort. Marie regarde Pierre qui mange une pomme. Marie qui regarde Pierre tombe. Pierre dort. Les pommes tombent. Outre les pronoms relatifs mentionnés, les mots considérés sont les articles définis et indéfinis, deux noms propres « Pierre » et « Marie », les noms communs « pomme », « femme », les verbes transitifs « regarder », et « manger », les verbes intransitifs « dormir » et « tomber ».

1 Donner une grammaire algébrique (ou non contextuelle) engendrant ces phrases et le moins possible de phrases incorrectes, sans se soucier des phénomènes d'accord ni de vraisemblance.

2 Donner un programme Prolog (un ensemble de clauses) modélisant cette grammaire de sorte que pour une phrase modélisée par des clauses sans corps $entre(mot_{i+1}, place_i, place_{i+1})$. on ait $S(0, n)$ si et seulement si cette phrase appartient au langage engendré par la grammaire.

3 On ajoute la clause $entre(Mot, [Mot|Reste], Reste)$ et on supprime les clauses $entre(mot_{i+1}, place_i, place_{i+1})$. servant à entrer la phrase. Montrer que la requête $s([mot_1, \dots, mot_n], [])$ répond « yes » ou « no » suivant que la phrase est ou non engendrée par la grammaire.

4 La phrase: « Marie dorment. » est elle produite par votre grammaire? La phrase: « Le pomme mange Marie. » est-elle reconnue? Modifier la grammaire Prolog pour éviter ce genre d'inconvénients en utilisant des traits (appelés aussi attributs), c'est-à-dire des variables supplémentaires. Il faut donc veiller à ce qu'un

nom ait un article dont le nombre et le genre soient les mêmes. De même il faut que le verbe s'accorde avec son sujet, y compris dans les relatives; mais il faut aussi faire en sorte qu'un verbe dont le sujet (ou l'objet) doit être animé ou au contraire inanimé ait bien un sujet (ou un objet) animé ou non, y compris dans les relatives. Pour simplifier, vu le vocabulaire choisi, on se contera d'accorder le verbe en nombre (tout les noms sont des troisièmes personnes), et de vérifier un trait humain ou non humain. On notera que ces deux types de problèmes sont de nature différentes: l'accord est un phénomène morphologique (ou morpho-syntaxique) et la nature du sujet ou de l'objet d'un verbe est un phénomène sémantique; ils sont néanmoins traités de manière similaire dans les grammaires d'unification dont les DCG.

5 Plutôt qu'un simple « yes » ou « no » on souhaite obtenir l'arbre d'analyse de la phrase de sorte que $S(T, [mot_1, \dots, mot_n], [])$ donne pour T l'analyse syntaxique de la phrase, puis « yes ».

```
|?- s(T, [une, femme, que, pierre, regarde, tombe], []).
T=
[phrase,
 [syntagme_nominal, une, femme,
 [relative, que,
 [syntagme_nominal, pierre, [t]], regarde]],
 [syntagme_verbal, tombe]]
```