

Calcul compositionnel d'une forme logique

Le projet est de mettre en œuvre le calcul rappelé dans la spécification précédente : étant donné la grammaire d'un fragment de langue naturelle (plus naturel que la version initiale de ce projet), étant donné un lexique qui associe à chaque token (mot) une lambda-expression, le parser construit (*on line*) une formule de logique des prédicats représentant la sémantique de la phrase analysée.

Lexique

Le chargement du lexique peut faire appel au parser défini à l'étape précédente. Pour des raisons de mise en œuvre, le plus simple est de prévoir parmi les actions reconnues des instructions de stockage dans le lexique dont la forme pourrait être :

```
lex chat N : L x [chat] x.
lex un Det : L P L Q E x ([P] x & [Q] x).
lex dort Vi : L x [dort] x.
lex jean NP : L P [P] j.
```

Pour uniformiser le traitement, on peut envisager d'entrer de la même manière les constantes non logiques et les variables dans la table des symboles. Alors, en préparant le "lexique" à l'avance, par couper/couler, on peut le charger dans le parser. Notez que dorénavant, le parser est donc capable de reconnaître **deux** langages distincts.

Grammaire

On se contentera d'une grammaire d'inspiration X' , très simplifiée, qui sera progressivement étendue. Par exemple, $S \rightarrow SN SV$
 $SN \rightarrow NP$
 $SV \rightarrow Vi$

Composition

Les séances précédentes nous ont fourni les outils nécessaires : il suffit d'associer à chaque règle de composition (par exemple $S \rightarrow SN SV$) une action qui (1) forme par application fonctionnelle un nouveau (lambda-)terme (par exemple $$$ = \text{cree_n}(\text{AF}, 0, \$1, \$2)$ pour la règle précédente), et (2) bêta-réduit ce terme (on peut aussi préférer faire toutes les β -réductions à la fin, cela ne devrait pas faire de différence, mais donne des formules plus difficiles à contrôler).

Typage

Les types (au sens du lambda-calcul typé de Montague) sont bien sur pertinents pour ce type de calcul, mais il n'est pas nécessaire de les rendre explicites. En particulier, il n'est pas nécessaire (au moins dans un premier temps) de typer toutes les constantes non logiques et les variables.

