

1 Grammaire sémantique avec nltk

Le but de ce TP est de travailler sur l'interface syntaxe-sémantique d'un (assez petit) fragment du Français, dans le cadre de la suite `nltk`. Plus précisément, le but de ce TP est de se livrer d'une part à un exercice d'« ingénierie grammaticale », et d'autre part à des manipulations sur les arbres syntaxiques, en vue d'une ébauche de traduction par transfert.

1.1 Chargement d'une grammaire

Le script python `script0.py` contient les instructions pour faire tourner l'analyse `nltk` sur un exemple minimal.

```
import nltk
mes_phrases = ["Yael ronfle", "un vent souffle"]
gram = nltk.data.load('file:./gram0.fcfg')
analyse = nltk.sem.text_interpret(mes_phrases, gram)
for p in mes_phrases:
    (arbre, formule) = analyse[p][0]
    print formule
```

La grammaire est donnée à la figure 1. La méthode de chargement crée une représentation en mémoire de la grammaire, et la méthode `text_interpret` prend une liste de phrases et les analyse successivement (syntaxe et sémantique), le résultat étant donné sous forme de dictionnaire.

1.2 Ingénierie grammaticale

Il s'agit ensuite d'augmenter la couverture de la grammaire, en enrichissant le lexique et les règles de syntaxe, pour couvrir (au moins) les phénomènes suivants :

- verbes transitifs
- verbes obliques (*parle à Marie*)
- négation (en *ne... pas*)
- adjectifs épithètes
- adjectifs attributs (du sujet)
- adjectifs relationnels (*fier de jean*)
- coordination de phrase

1.2.1 Manipulation des arbres syntaxiques

Pour préparer une manipulation des arbres syntaxiques fournis par `nltk`, on commencera par réaliser les opérations suivantes :

- écriture de la phrase correspondant à un arbre syntaxique donné
- remplacement d'un sous-arbre complet par un autre
- remplacement d'une configuration par une autre

1.2.2 Traduction par transfert

L'idée de la traduction par transfert est que l'on établisse une correspondance directe entre la langue de départ et la langue d'arrivée. Cette correspondance peut être établie au niveau des mots (cf. TP du cours de Langues Formelles), mais les limites sont évidentes.

Une alternative est de formuler les correspondances au niveau de sous-arbres syntaxiques. Par exemple, pour que *donne à Jean* soit correctement traduit en *give John*, il faut spécifier la correspondance au niveau des arbres :

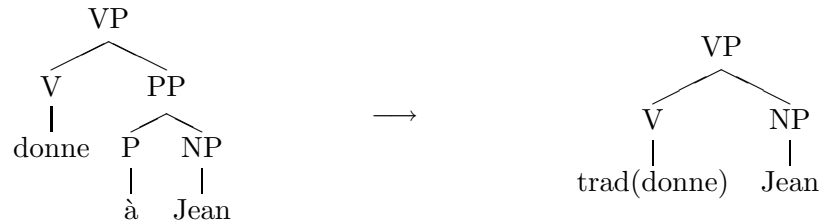


FIG. 1 – La grammaire minimale `gram0`

```

# gram0.fcfg : grammaire minimale pour démarrer le TP
# Il faut indiquer l'axiome avec la ligne suivante :

%start S

# Grammaire
# -----

S[sem=<app(?s, ?v)>] -> SN[sem=?s] SV[sem=?v]

SN[sem=?n]          -> NP[sem=?n]
SN[sem=<app(?d,?n)>] -> Det[sem=?d] N[sem=?n]

SV[sem=?v]         -> Vi[sem=?v]

# Lexique
# -----

NP[sem=<\P.P(yael)>] -> 'Yael'
Det[sem=<\P Q. exists x. (P(x) and Q(x))>] -> 'un'
N[sem=<\x. wind(x)>] -> 'vent'
Vi[sem=<\x. blow(x)>] -> 'souffle'
Vi[sem=<\x. snore(x)>] -> 'ronfle'
    
```