

Langages Formels

TP

Nettoyage de grammaires
algébriques

Indications python / NLTK

M1 Linguistique informatique

Paris 7

08/09 - Marie Candito

Calcul de epsilon_plus

- voir l'algo fourni en correction de la feuille 2
- Indications python:
 - `epsilon_plus` représenté comme une `list`
 - tester si `x` est dans une liste `mylist`:
 - `if x in mylist`:
 - ajouter un élément `x` à une liste `mylist` :
 - `mylist.append(x)`
 - tester si tous les élts d'une liste vérifie une contrainte :
 - \Leftrightarrow tester s'il existe un élts qui ne vérifie pas la contrainte
 - boucles `for` :
 - `for x in mylist`:
 - instruction 1
 - rem : `break` permet de sortir de la boucle `for` la plus proche
 - `else`:
 - instruction effectuée si la boucle `for` s'est déroulée sans appel à `break`

Suppression productions- ϵ

Indications python / NLTK

- la chaîne vide ϵ s'écrit :
 - `[]` # liste vide de symboles
- « `sigma_rules := { ϵ }` » s'écrit :
 - `sigma_rules = [[]]` # liste réduite à la chaîne vide
- « pour chaque chaîne de `sigma_rules` » va s'écrire
 - attention on parcourt `sigma_rules`, et on veut la modifier
 - => avant le parcours, copier `sigma_rules` dans `tmp_sigma_rules`, à utiliser pour le parcours (for)
 - et réinitialiser `sigma_rules = []`
 - ensuite on ne « remplace » pas les chaînes, on les ajoute simplement à ce `sigma_rules` réinitialisé
 - ajouter chaîne + x s'écrit:
 - `sigma_rules.append(chaîne + [x])`

Suppression productions- ϵ

Indications python / NLTK

- créer une règle :

```
new_rule = nltk.cfg.Production(partie_gauche, partie_droite)
```

avec `partie_gauche` = un Nonterminal

`partie_droite` = un tuple (=list inchangeable) de symboles

- « ajouter à `new_rules` les règles `sigma_rules`, sauf celles de la forme $X \rightarrow \epsilon$ » va s'écrire

- les éléments de `sigma_rules` sont des parties droites

- il faut construire de nouvelles règles, avec comme partie gauche celle de *old_rule*

- au passage il faut dédoublonner

- on n'ajoute une règle que si elle est nouvelle

```
for partie_droite in sigma_rules:
```

```
    # si partie_droite  $\neq \epsilon$ 
```

```
    if partie_droite <> [ ]:
```

```
        # création d'une nouvelle instance Production
```

```
        new_rule = nltk.cfg.Production(old_rule.lhs(), tuple(partie_droite))
```

```
        # si la règle est nouvelle
```

```
        if new_rule not in new_rules:
```

```
            new_rules.append(new_rule)
```

Suppression productions- ϵ

Indications python / NLTK

- « Si ϵ appartient au langage engendré par grammaire » va s'écrire
 - ... si l'axiome appartient à `epsilon_plus`
 - `if grammar.start() in epsilon_plus`
- Créer un nouvel axiome `S'` :
 - constructeur de `nltk.cfg.Nonterminal`
`new_axiom = nltk.cfg.Nonterminal('Sprime')`
- retourner nouvelle Grammaire :
 - constructeur de `nltk.cfg.Grammar` prend en arguments l'axiome, et la liste des productions
`return nltk.cfg.Grammar(new_axiom, new_rules)`