Listes des algos à utiliser pour la SLR

Via http://univ-biskra.dz/fac/fsesnv/cours/compilation/syntaxique%20doc.pdf

Algorithme de construction des ensembles PREMIER

1. Si X est un non-terminal et X →Y1,Y2...Yn est une production de la grammaire (avec Yi symbole terminal ou non-terminal) alors:

* ajouter les éléments de PREMIER(Y1) sauf dans PREMIER(X)
* s'il existe j (j∈{2,...,n}) tel que pour tout i=1,...,j-1 on a ε∈PREMIER(Yi), alors ajouter les éléments de PREMIER(Yj) sauf ε dans PREMIER(X)
* si pour tout i=1,..., n, ε ∈ PREMIER(Yi), alors ajouter dans PREMIER(X)

1. Si X est un non-terminal et X → ε est une production, ajouter ε dans PREMIER(X)
2. Si X est un terminal, PREMIER(X)={X}
3. Recommencer jusqu'à ce qu'on n'ajoute rien de nouveau dans les ensembles PREMIER.

Algorithme de construction des ensembles SUIVANT :

1. Ajouter un marqueur de fin de chaîne (symbole $ par exemple) à SUIVANT(S) (S est l'axiome)
2. S'il y a une production A→αBβ où B est un non-terminal, alors ajouter le contenu de PREMIER(β) à SUIVANT(B), sauf ε
3. S'il y a une production A→αB, alors ajouter SUIVANT(A) à SUIVANT(B)
4. S'il y a une production A→αBβ avec ∈εPREMIER(β), alors ajouter SUIVANT(A) à SUIVANT(B).
5. Recommencer à partir de l'étape 3 jusqu'à ce qu'on n'ajoute rien de nouveau dans les ensembles SUIVANT.

Collection des items d'une grammaire :

0- Rajouter l'axiome S' avec la production S'→S

1- Mettre dans l'item I0 la Fermeture de S' → .S})

2- Mettre I0 dans Collection

3 - Pour chaque I dans Collection faire

Pour chaque X tel que (I, X) est non vide

ajouter ce (I, X) dans Collection

Fin pour

4 - Recommencer 3 jusqu'à ce qu'on n'ajoute rien de nouveau

Construction de la table d'analyse SLR :

1- Construire la collection d'items {I0, ... In}

2- l'état i est construit à partir de Ii:

a) pour chaque (Ii, a) = Ij : mettre décalage par j dans la case M[i,a]

b) pour chaque (Ii, A) = Ij: mettre aller à j dans la case M[i,A]

c) pour chaque A→α. contenu dans Ii :

pour chaque a de SUIVANT(A) faire

mettre réduction par numéro (de la règle A→α) dans la case M[i,a]

Analyseur syntaxique SLR:

On part dans l'état 0, et on empile et dépile non seulement les symboles (comme lors de l'analyseur LL) mais aussi les états successifs.