

## 6.1 Analyse LL(1)

Soient les deux grammaires suivantes où l’on a numéroté les règles :

$G_1 =$

1	$S \rightarrow AD$	2	$S \rightarrow cS$
3	$A \rightarrow a$	4	$A \rightarrow EF$
5	$D \rightarrow d$	6	$D \rightarrow AD$
7	$E \rightarrow a$	8	$E \rightarrow \varepsilon$
9	$F \rightarrow c$	8	$F \rightarrow \varepsilon$

$G_2 =$

1	$S \rightarrow ABC$
2	$A \rightarrow a$
3	$B \rightarrow b$
4	$B \rightarrow \varepsilon$
5	$C \rightarrow cC$
6	$C \rightarrow \varepsilon$

1. Calculer (sans forcément dérouler tout l’algorithme) PREMIER(AD). Etudiez la dérivation pour reconnaître le mot 'd'. En quoi l’on peut voir que la notion de PREMIER est insuffisante pour déterminer quand la règle  $S \rightarrow AD$ .  
**Correction** On obtient bien PREMIER(AD)=a, d, c, mais PREMIER(EF)=a,c, donc avec le seul ensemble PREMIER, on manque le fait que si 'd' est le prochain symbole à apparier, alors, la règle  $A \rightarrow EF$  est possible (et même obligatoire).
2. Pour la seconde grammaire, donner les ensembles PREMIER et SUIVANT pour chaque non terminal, et la table de prédiction LL. Il s’agit à chaque fois de se remémorer/approfondir les algorithmes correspondants.
3. Implémenter le calcul de PREMIER et SUIVANT

On redonne l’algo pour PREMIER :

Pour tout non terminal, initialiser PREMIER(A) comme un ensemble vide  
 continuer=Vrai  
 Tant que continuer=Vrai  
 —continuer=Faux  
 —Pour chaque règle R  
 ———soit A la partie gauche de R  
 ———et RHS la partie droite de R  
 ———Pour chaque élément X de RHS  
 —————ajouter PREMIER(X) à PREMIER(A)  
 —————continuer=Vrai s’il y a augmentation effective  
 —————Si X n’est pas dans epsilonplus :  
 —————Arrêter de parcourir RHS

**SUIVANT :**

INITIALISATION :

- pour tout non terminal, initialiser SUIVANT(A) comme un ensemble vide
  - pour l’axiome S, ajouter \$ à SUIVANT(S)
- PARTIE 1 : étude des parties droites  
 Pour chaque règle  $A \rightarrow X_1 X_2 \dots X_n$   
 –Si  $X_1$  est terminal : passer à la règle suivante –Pour chaque  $X_i$  dans  $X_1 X_2 \dots X_{n-1}$   
 —Pour chaque  $X_j$  dans  $X_{i+1} \dots X_n$

—ajouter PREMIER( $X_j$ ) à SUIVANT( $X_i$ ) —si  $X_j$  n'est dans epsilonplus :  
—Arrêter de parcourir  $X_{i+1}...X_n$

PARTIE 2 :

continuer=Vrai

tant que continuer est vrai :

-Continuer=faux

-Pour chaque règle  $A \rightarrow X_1...X_n$

—( On parcourt la partie droite en partant de la fin)

—( ce qui peut suivre  $A$  peut suivre  $X_n$ )

—Si  $X_n$  est terminal : passer à la règle suivante —Pour chaque  $X_i$  dans  $X_n X_{n-1}...X_1$

—ajouter SUIVANT( $A$ ) à SUIVANT( $X_i$ )

—continuer=Vrai s'il y a eu augmentation effective

—Si  $X_i$  n'est dans epsilonplus :

—Arrêter de parcourir  $X_n X_{n-1}...X_1$

**L'algo pour construire la table :**

On calcule PREMIER pour toutes les parties droite de règle

Puis,

Pour toute règle  $A \rightarrow \alpha$

-Pour tout élément  $f$  de PREMIER( $\alpha$ )

—On ajoute  $\alpha$  à la case  $(A, f)$  —Si  $\alpha$  dérive epsilon (i.e. ne contient que des éléments dans epsilonplus)

—Pour tout élément  $f$  de SUIVANT( $A$ )

—On ajoute  $\alpha$  à la case  $(A, f)$