

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

1. Pour cette 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.
  
2. 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 regle R

——soit A la partie gauche de R

——et RHS la partie droite de R

——Pour chaque element 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