

Analyse Syntaxique - Année 2010-2011

Devoir Surveillé

(Durée 1 heure 30)

Exercice 1

Nous considérerons dans la suite qu'un tableau en html est défini comme suit :

- Le tableau est encadré par les balises <TABLE> et </TABLE>.
- Chaque ligne est encadrée par les balises <TR> et </TR>
- Les cellules de valeur sont encadrées par les balises <TD> et </TD>

Exemple : le code html ci-dessous

```
<TABLE>
  <TR>

    <TD> Valeur 11 </TD>
    <TD> Valeur 12 </TD>
    <TD> Valeur 13 </TD>
  </TR>
  <TR>
    <TD> Valeur 11 </TD>
    <TD> Valeur 12 </TD>

    <TD> Valeur 13 </TD>

  </TR>
</TABLE>
```

génère le tableau suivant :

```
Valeur 11 Valeur 12 Valeur 13
Valeur 21 Valeur 22 Valeur 23
```

Certaines parties du fichier peuvent être des commentaires. Les commentaires html sont encadrés par <!-- et -->.

Ecrivez un programme lex qui prend en entrée un programme HTML que l'on supposera bien structuré et qui pour chaque tableau, écrit la phrase : "le tableau commence à la ligne *i* et contient *k* lignes" où *i* est le

numéro de la ligne du code HTML où commence le tableau et k est le nombre de lignes dans ce tableau (pour l'exemple précédent, $k = 2$).
 Cependant vous ne devrez pas comptabiliser les lignes en commentaires, ni produire un affichage pour des tableaux qui seraient entièrement en commentaires.

Exercice 2

On considère la grammaire G_0 suivante :

S : L 'b'
 L : 'a' L | 'a'

1. Calculer l'automate des LR(0) items et montrer que cette grammaire n'est pas LR(0).
2. Calculer les suivants de L et montrer qu'elle est SLR(1).
3. Détailler l'analyse du mot aab (on donnera à chaque étape le contenu de la pile et l'action).

On considère maintenant la grammaire G_1 suivante :

S : L 'a' 'b'
 L : 'a' L | 'a'

4. Expliquer les différences entre l'automate des LR(0) items de G_1 et celui de G_0 .
5. Calculer les suivants de L et montrer que G_1 n'est pas SLR(1).
6. Calculer maintenant l'ensemble des mots de 2 lettres suivant L et montrer que si l'automate connaît à tout moment deux symboles d'avance, alors il peut lever l'ambiguïté SLR(1). On dit que G_1 est SLR(2).
7. Proposer une grammaire G_k qui n'est pas SLR(k-1) mais qui est SLR(k) pour $k > 0$ (i.e. qui nécessite la connaissance d'au moins k symboles d'avance pour lever les ambiguïtés de l'automate).

On considère maintenant la grammaire suivante :

S : L X
 L : 'a' L | 'a'
 X : 'a' X 'b' | 'c'

8. Montrer que cette grammaire n'est SLR(k) pour aucun entier k .