

Contrôle

Théorie des langages et compilation

Questions diverses :

1) Parmi les propositions suivantes, sélectionnez la proposition qui correspond à la démarche correcte à suivre pour écrire un analyseur lexical en utilisant Lex :

- A - Donner l'automate reconnaissant les unités; puis écrire la fonction d'analyse.
- B - Donner les expressions régulières des unités; puis écrire la fonction d'analyse.
- C - Ecrire la fonction d'analyse; puis donner les expressions régulières des unités.
- D - Ecrire la fonction d'analyse; puis donner l'automate reconnaissant les unités.

2) Soit le langage L constitué des expressions de la forme $idf1 = idf2 op1 idf3 op2 idf4$ ou de la forme $idf1 = idf2 op1 idf3$ où les notations $idf1, idf2, idf3, idf4$ dénotent des identificateurs du langage. Les identificateurs autorisés du langage doivent commencer par une lettre majuscule éventuellement suivie de lettres minuscules. Les notations $op1$ et $op2$ désignent un opérateur parmi les opérateurs suivants : +, -, *, /

3.1) Donnez l'expression régulière des chaînes de caractères identificateurs de ce langage

3.2) Donnez l'automate d'état fini reconnaissant les identificateurs de ce langage

3.3) entourez pour chacune des instructions suivantes le type d'erreur retournée par un compilateur de L (une seule réponse pour chaque instruction) :

Y = a + B

A - Pas d'erreur **B** - Erreur lexicale C - Erreur syntaxique

Y = + A * C

A - Pas d'erreur B - Erreur lexicale C - Erreur syntaxique

Y = A / C

A - Pas d'erreur B - Erreur lexicale C - Erreur syntaxique

Y = A / C - B

A - Pas d'erreur B - Erreur lexicale C - Erreur syntaxique

REAL R

BOOLEAN B, C

R = B * C

A - Pas d'erreur B - Erreur lexicale C - Erreur syntaxique **D** - Erreur sémantique

REAL R

BOOLEAN B, C

R = B ** C

A - Pas d'erreur B - Erreur lexicale **C** - Erreur syntaxique D - Erreur sémantique

Exercice 1

Soit le programme Lex suivant :

```
%{
#include <stdio.h>
%}
%%
"<ul>"    printf("<ol>");
```

```
"</ul>"    printf("</ol>");
"bold"     printf("<b>bold</b>");
"<b>bold</b>" ECHO;
"italics"  printf("<i>italics</i>");
"<i>italics</i>" ECHO;
%%
```

On fournit au programme précédant le texte source suivant en entrée :

```
<html>
<body>
<ul>
<li> le mot bold se trouve dans ce fichier </li>
<li>Le mot <b>bold</b> se trouve dans ce fichier </li>
</ul>
</body>
</html>
```

Donnez le résultat de l'exécution du programme précédent
(Note : ECHO est une macro qui est équivalente à l'instruction `fprintf(yyout,"%s",yytext)`)

Exercice 2

Dans cet exercice un identificateur est toute chaîne commençant par une lettre suivie éventuellement de lettres ou chiffres. Le programme Lex analyseur.l donné ci-après est incomplet. Il est censé analyser un texte source pour afficher le nombre d'identificateurs de ce texte, le nombre des entiers, le nombre de caractères et le nombre de lignes comme suit :
nombre de lignes = %d, nombre de cars = %d, nombre d'identifs = %d

```
%option nomain
%option noyywrap
%{
    int num_lignes = 0, num_cars = 0;
%}
%%
\n    ++num_lignes; ++num_cars;
.    ++num_cars;
%%
main() {
    yylex();
    printf("nombre de lignes = %d, nombre d'entiers = %d, nombre de cars = %d,
nombre d'identifs = %d\n", num_lignes, num_entiers, num_cars, num_idf);
}
```

1. Compléter le programme précédant pour le rendre fonctionnel.
2. Pour compiler le programme précédant, que faut-il faire ? (Sélectionner une bonne réponse) :
A - lex analyseur
B - lex analyseur.l
C - lex analyseur.l
gcc analyseur.c -o analyseur
D - lex foo.l

E- lex foo.l
gcc analyseur
F- lex analyseur.l
gcc lex.yy.c -o foo
G- lex foo.l
gcc lex.yy.c -o foo

3. Vous disposez d'un terminal sous Fedora. On suppose que l'exécutable de l'analyseur s'appelle ex2 et on souhaite analyser le texte suivant :
- ```
bonjour
lex
au revoir.
```

Quel est le résultat qui sera affiché par le programme après l'avoir exécuté ?

- A - nombre d'identificateurs = 4 nombre de caracteres = 18 nombre de lignes = 3  
B - nombre d'identificateurs = 4 nombre de caracteres = 21 nombre de lignes = 3  
C - nombre d'identificateurs = 3 nombre de caracteres = 19 nombre de lignes = 3  
D - nombre d'identificateurs = 4 nombre de caracteres = 19 nombre de lignes = 3  
E - nombre d'identificateurs = 3 nombre de caracteres = 22 nombre de lignes = 3  
F - nombre d'identificateurs = 4 nombre de caracteres = 22 nombre de lignes = 3

4. L'analyseur lit le texte source depuis l'entrée standard. Que faut-il faire pour pouvoir le lire depuis un fichier appelé texte.txt ?

### Exercice 3

On considère la grammaire  $G1$  des expressions arithmétiques simples

$T = \{\text{nombre } \$ + - \}$

$E \rightarrow \text{nombre Op } F$

$F \rightarrow E \mid \text{nombre } \$$

$Op \rightarrow + \mid -$

où num désigne un nombre

1. Quel est le directeur de  $F \rightarrow E$  ?
2. Quel est le directeur de  $F \rightarrow \text{nombre } \$$  ?
3. La grammaire  $G1$  est-elle  $LL(1)$  ? justifiez votre réponse

Soit maintenant la grammaire  $G2$  définie par :

$T = \{\text{nombre } \$ + - \}$

$E \rightarrow \text{nombre Op nombre } F$

$F \rightarrow Op E \mid \$$

$Op \rightarrow + \mid -$

On souhaite écrire les procédures de l'analyseur syntaxique descendant associé à  $G2$ .

1. La grammaire est elle  $LL(1)$  ?
2. Quel est le nombre des procédures de l'analyseur syntaxique descendant ?
3. Donnez le code de ces procédures.