UNIVERSITE DE BOURGOGNE
U.F.R. Sciences et Techniques
Filière : Licence 3 Informatique

Année 2018-2019
13 mai 2019

Session : 1

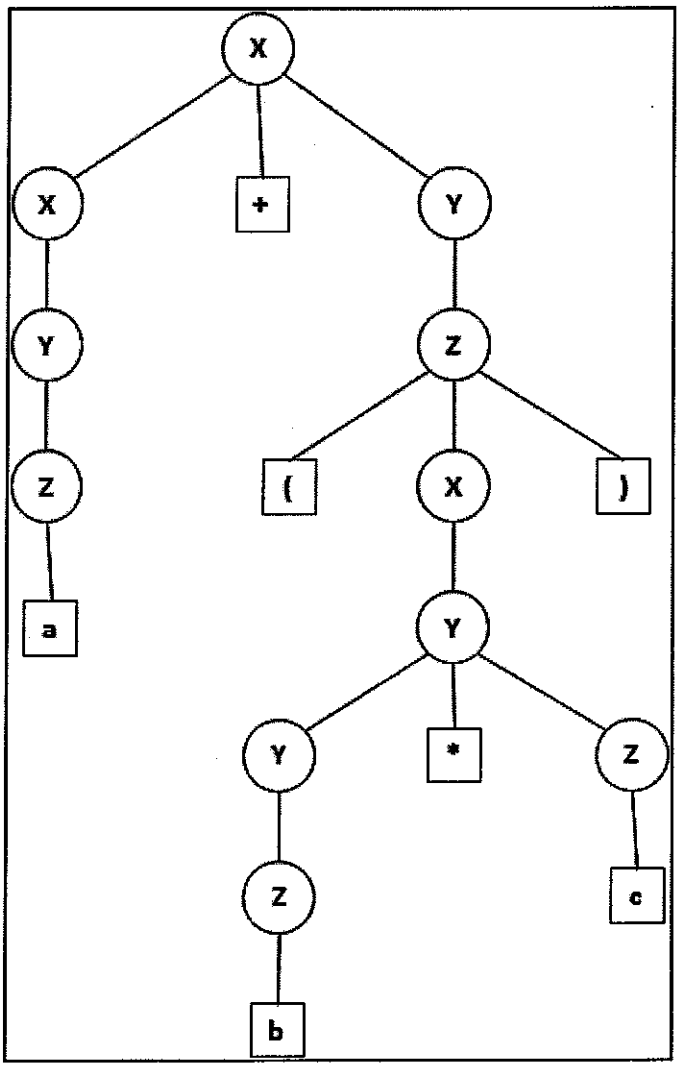
EPREUVE : Langages Formels et Compilation

Durée : 2 h 00 – (documents papiers - sauf livres - autorisés ; appareils électroniques interdits)

Les exercices sont indépendants. Le barème est donné à titre indicatif.

Exercice 1 – 3.5 points

La figure ci-contre représente un arbre de dérivation.



1. De quel mot cet arbre représente-t-il la dérivation ?
2. Donnez la dérivation la plus à gauche associée à cet arbre.
3. Sachant que toutes les règles de la grammaire ont été utilisées au moins une fois dans la dérivation, donnez la représentation formelle de la grammaire à partir de laquelle l'arbre a été construit.
4. De quel type est cette grammaire ?

Exercice 2 – 5.5 points

Soit la grammaire G_2=({S, X, Y}, {a, b, c}, S, {S→aa|aXa, X→aXa|Y, Y→bYcc|λ})

1. Donnez 3 mots du langage engendré par G_2.
2. Transformez la grammaire G_2 de manière à obtenir une grammaire équivalente G_2' sous forme normale de Greibach. Ecrivez les différentes étapes de la transformation.
3. A partir de G_2', construisez un automate à pile reconnaissant les mots du langage par pile vide.

Exercice 3 – 3.5 points

Proposez un automate à mémoire linéairement bornée qui reconnaît les mots formés de 0 et de 1, dans n'importe quel ordre, dont le nombre de 0 est égal au nombre de 1.

Exercice 4 – 4 points

Soit la grammaire G_4=({S, X, Y, A, B}, {a, b}, S, {S→AX|BY|a|b, X→BY|b, Y→AX|a, A→a, B→b})

1. Construisez la pyramide de Coke-Kasami-youger pour le mot ababba.
2. D'après la pyramide obtenue, quels sous-mots de ababba appartiennent au langage engendré par G_4 ? Justifiez votre réponse.

Exercice 5 – 3.5 points

Soient les programmes lex et yacc ci-dessous :

```
%%
a[^ab]*a {yylval=yyleng-2; return A;}
b[^ab]+b {yylval=yyleng-2; return B;}
. {printf("%s",yytext);}
```

```
%token A B
%start X
%%
X : A Y {$$=$1+$2;printf("%d\n",$$);}
  | B Z {$$=$2-$1;printf("%d\n",$$);} ;
Y : B Z {$$=$2-$1;} | {$$=0;} ;
Z : A Y {$$=$1+$2;} | {$$=0;} ;
%%
#include "lex.yy.c"
int yyerror()
  {printf("syntax error\n");return 0;}
main()
  {yyparse();}
```

Et les fichiers textes :

Fichier test1

```
moathkopfrabyhngtbaabklmbaefghjabb
```

Fichier test2

```
bbaabhjkiuytrbadddafin
```

On applique l'analyseur obtenu à partir des fichiers lex et yacc sur les fichiers test1 et test2.

1. Qu'affichera l'analyseur lexical
   - à la lecture de test1 ?
   - à la lecture de test2 ?
2. Qu'enverra l'analyseur lexical à l'analyseur syntaxique
   - à la lecture de test1 ?
   - à la lecture de test2 ?
3. Qu'affichera l'analyseur syntaxique
   - à la lecture de test1 ?
   - à la lecture de test2 ?