UNIVERSITÉ DE SHERBROOKE

Evaluation FORMATIVE Session S4info- APP 5

GIF 340- Éléments de compilation

Département de génie électrique et de génie informatique

Faculté de génie

Université de Sherbrooke

Été 2022

Documentation autorisée :

Aucune

Ahmed Khoumsi

PROBLÈME 1

On considère l'expression régulière suivante :

$$0\,(0 \mid 1)^{*}\,0$$

Décrire cette expression régulière par des phrases (textuelles) et par un automate à états finis.

PROBLÈME 2

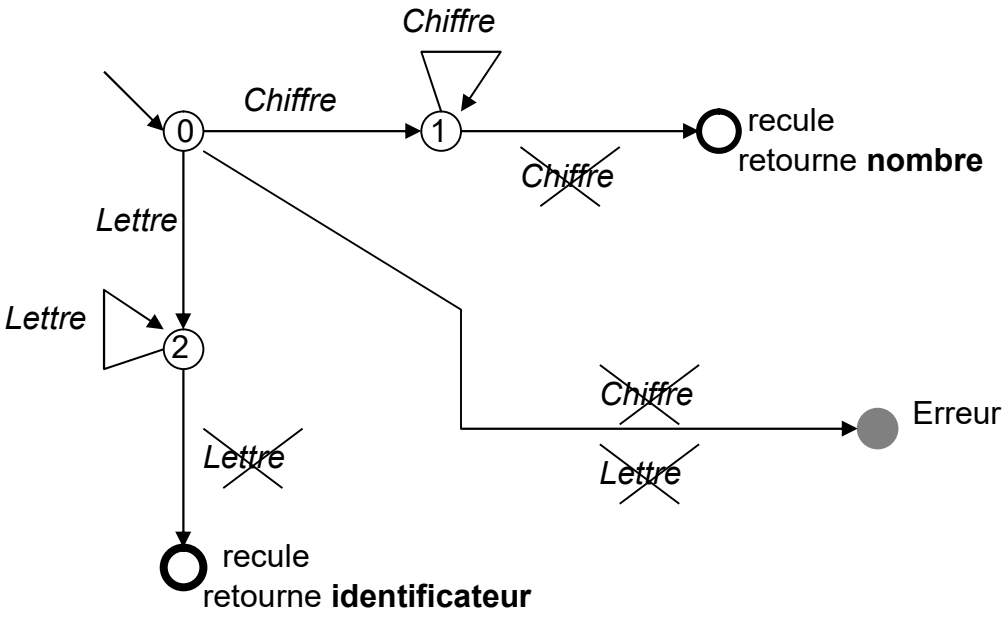
On considère :

- des commentaires qui débutent et se terminent par %, et qui peuvent contenir des lettres majuscules et minuscules et des chiffres entre 0 et 9 ou être vides;
- des nombres constitués de chiffres 0 à 9.

1) Décrire les commentaires par une expression régulière et par un automate à états fini.
2) Décrire les nombres par une expression régulière et par un automate à états finis.
3) Construire un automate à états finis de l'analyseur lexical, c-à-d. qui détecte les erreurs lexicales, et reconnaît et retourne les commentaires et les nombres.

PROBLÈME 3

On considère l'automate à états finis ci-dessous :



~~*Chiffre*~~ signifie : tout caractère qui n'est pas un chiffre

~~*Lettre*~~ signifie : tout caractère qui n'est pas une lettre

« recule » signifie : recule le pointeur de lecture de la phrase analysée

« retourne nombre » signifie : retourne un objet

- indiquant qu'un nombre a été reconnu
- contenant la valeur (ou la chaîne de caractères) du nombre

« retourne identificateur » signifie : retourne un objet

- indiquant qu'un identificateur a été reconnu
- contenant la chaîne de caractères définissant l'identificateur

« Erreur » signifie : a détecté une erreur

Dériver le pseudo-code **ou** le code Java (au choix) d'un analyseur lexical correspondant à l'automate à états finis de la page précédente.

PROBLÈME 4

Soit l'expression suivante : position = acceleration * temps ^ 2 / 2 + vitesse * temps

où : tous les opérateurs sont associatifs à droite

^ est l'opérateur le plus prioritaire

* et / ont même priorité et sont plus prioritaire que +

= est l'opérateur le moins prioritaire

1) Représenter l'arbre syntaxique abstrait correspondant à cette expression arithmétique.
2) Dériver les expressions prefix, infix et postfix correspondant à l'AST ci-dessus.

PROBLÈME 5

On considère les arbres syntaxiques abstraits dont chaque feuille correspond à un nombre qui consiste en une séquence de chiffres, et chaque nœud correspond à un opérateur binaire qui peut être + - ^ * / .

1) Décrire en pseudo-code ou en code Java (au choix) les attributs et les méthodes d'une classe Feuille qui implante les feuilles d'arbre syntaxique abstrait.
2) Décrire en pseudo-code ou en code Java (au choix) les attributs et les méthodes d'une classe Noeud qui implante les nœuds d'arbre syntaxique abstrait.

PROBLÈME 6

On considère la grammaire suivante :

U → V
U → V + V
V → W
V → W * W
W → (U)
W → id

où id , + , * , (,) sont des symboles terminaux, U, V, W sont des symboles non-terminaux , et U est le symbole non-terminal de départ.

1) Calculer les ensembles PREMIER et SUIVANT de chacun des symboles non-terminaux.
2) Effectuer la séquence de dérivations à gauche qui permet d'obtenir id + (id * id).
3) Est-ce que la grammaire ci-dessus est LL(1) ? Si ce n'est pas le cas, transformer la grammaire de manière à ce qu'elle devienne LL(1).
4) Dériver le pseudo-code **ou** le code Java (au choix) des fonctions d'un analyseur syntaxique descendant qui correspondent aux symboles non-terminaux de la grammaire LL(1).

EXERCICES

grammaires, expressions régulières, automates

Exercice 1 (Utilisation de grammaire)

On considère la grammaire suivante où S est le symbole de départ, les lettres minuscules sont des symboles terminaux, et les lettres majuscules sont des symboles non-terminaux :

S → AB A → Ca B → Ba B → Cb B → b C → cb C → b

Montrer que la chaîne cbab appartient au langage créé par cette grammaire.

Exercice 2 (Construction de grammaire)

Trouver une grammaire qui définit le langage constitué des chaînes binaires contenant autant de 0 que de 1.

Exercice 3 (Types de grammaires)

Déterminer le type de chacune des grammaires suivantes, où S est le symbole de départ, les lettres minuscules sont des symboles terminaux, et les lettres majuscules sont des symboles non-terminaux :

S → aAB A → Bb B → ε

S → aB A → a A → b

S → ABa AB → a

S → ABa AB → ab

Exercice 4 (Construction de grammaire et d'expression régulière à partir d'AEF)

On considère l'AEF défini par :

- Ensemble d'états Q = {0, 1}, État initial = 0
- Ensemble d'états finaux F = {1}
- Alphabet Σ = {a, b}
- Fonction de transition δ : Q x Σ → Q

  δ(0, a) = 1 δ(0, b) = 0 δ (1, a) = 1 δ (1,b) = 0

1. Construire le diagramme d'états de cet AEF
2. Construire une grammaire définissant le langage de cet AEF
3. Construire une expression régulière définissant le langage de cet AEF

fin