

Module Système - Architecture - Compilation

Examen de compilation — 14 Septembre 1998 — Durée: 1 heure

Exercice 1 Le terme « mot » dans cet exercice désigne les chaînes composées uniquement de lettres (majuscules ou minuscules). De plus, si un mot commence par une majuscule, ses autres lettres (s'il en a) sont soit toutes des majuscules, soit toutes des minuscules; si un mot commence par une minuscule, ses autres lettres (s'il en a) sont toutes des minuscules.

Écrire à l'aide de `lex` un programme qui compte les mots se terminant par la chaîne *tique* en minuscules (comme par exemple « informatique », « Antique » ou « tique »). Les autres mots ne devront pas être affichés par le programme. Les chaînes qui ne sont pas des mots devront par contre être signalées.

Exercice 2 Donner deux expressions rationnelles qui, en `lex`, permettent de filtrer les motifs composés du caractère * (étoile) suivi immédiatement d'un identificateur `C`, comme par exemple `*p1`.

Exercice 3 On considère la grammaire \mathcal{G}_1 suivante :

$$\begin{array}{l|l} E \longrightarrow \text{id} = E & T \longrightarrow F \\ E \longrightarrow E + T & F \longrightarrow (E) \\ E \longrightarrow T & F \longrightarrow *F \\ T \longrightarrow T * F & F \longrightarrow \text{id} \end{array}$$

où l'ensemble des symboles terminaux est $\{\text{id}, =, +, *,), (\}$.

1. Décrire de façon précise et concise le langage reconnu par \mathcal{G}_1 .
2. Pour chacune des chaînes suivantes, dire si elles sont dans le langage engendré par la grammaire. Si non, justifier, et si oui, donner un arbre de dérivation.
 - a) `id * *id + id`
 - b) `(id = id) = id`
 - c) `id = (id = id)`

On considère maintenant la grammaire \mathcal{G}_2 suivante :

$$\begin{array}{l|l} E \longrightarrow E + T & T \longrightarrow (E) \\ E \longrightarrow T & T \longrightarrow \text{id} \end{array}$$

3. Comparer le langage engendré par \mathcal{G}_1 et le langage engendré par \mathcal{G}_2 .
4. Calculer les ensembles `Premier` et `Suivant` pour les non-terminaux de \mathcal{G}_2 .
5. Construire l'automate des items de \mathcal{G}_2 .
6. La grammaire \mathcal{G}_2 est-elle LR(0)? SLR(1)? ambiguë? Justifier les réponses.
7. Simuler le comportement d'un analyseur syntaxique LR sur l'entrée `id + (id)`.