Maîtrise d'Informatique

Analyse Syntaxique et Compilation (ASC)

Jeudi 25 janvier 2001

Durée 3 heures, documents manuscrits autorisés

Problème 1 : 6 points

On considère le langage L des expressions arithmétiques en notation infixe sans parenthèses et dont les variables ont pour nom a. L'alphabet terminal est donc $A = \{a, +, *\}$ et les régles de grammaire

$$E \to T \quad E \to E + T$$
$$T \to T * a \quad T \to a$$

Question 1. Montrer que la grammaire ci-dessus n'est pas $LL(1)$ de deux façons différentes: d'abord en utilisant la caractérisation donnée en cours ensuite en donnant un exemple où il n'est pas possible d'effectuer l'analyse descendante en ne connaissant que la première lettre du mot à engendrer.

Question 2. Calculer les langages $R_0(E), R_0(T)$ sans utiliser les 0-items, (on donnera la forme des dérivations droites que l'on peut obtenir pour cette grammaire); puis les langages $R_0(i)$ pour chacune des 4 règles de grammaire.

Question 3. Cette grammaire est-elle $LR(0)$?

On donne la définition récursive suivante :

une liste d'expressions arithmétiques est ou bien un élément de L, ou bien une suite non vide (mais pouvant ne contenir qu'un seul élément) de listes d'expressions arithmétiques séparées par des virgules le tout étant mis entre parenthèses.

Question 4. Donner une grammaire non ambigue pour les listes d'expressions arithmétiques.

Question 5. Dessiner l'arbre d'analyse syntaxique de la liste d'expressions suivante pour la grammaire que vous avez proposée à la Question 4.

`(a + a*a, (a, (a,a)), (a))`

Problème 2: 6 points

Sur certains supports d'information électroniques où sont mémorisées des suites de 0 et de 1 il est déconseillé d'avoir des suites de 0 (ou de 1) consécutifs trop longues. On se propose de coder les suites de 0 et de 1 quelconques de façon à ne pas avoir trop de 0 ou de 1 consécutifs. Dans ce problème on étudie différents codages possibles qui évitent 4 symboles égaux consécutifs. On considère ainsi des automates avec sortie qui effectuent les codages et décodages correspondants.

Question 1. Une première idée est d'ajouter des 1 tous les trois 0 dans chaque facteur de la suite contenant plus de 3 lettres 0 consécutives et des 0 dans chaque facteur contenant plus de trois lettres 1. Ainsi la suite

$$1\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0$$

est transformée en

$$1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0$$

Donner un automate avec sortie qui réalise cette transformation. Peut-on retrouver de manière unique une suite à partir de sa transformée?

Question 2. On choisit une autre technique consistant à utiliser le code $01, 10, 011, 100$. Ainsi la suite de 0 et de 1 est décomposée en mots de deux lettres, ensuite 00 est transformé en 100, 11 en 011; 01 et 10 sont inchangés. Sur le même exemple que tout à l'heure cela donne ainsi

$$1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 0$$

a) Montrer que l'on obtient bien un codage avec décodage unique.

b) Montrer que toute suite obtenue par codage ne contient pas plus de trois symboles égaux consécutifs.

c) Donner un automate avec sortie pour ce codage : on utilisera un symbole de fin de mot et les mots à coder seront supposés de longueur paire.

d) Montrer, à l'aide d'exemples que le décodage bien qu'unique ne peut pas être réalisé par un automate fini avec sortie.

Question 3. On décide de modifier le codage précédent pour que le décodage puisse être fait avec un automate fini. On utilise alors le code $01, 10, 0111, 1000$. On obtient ainsi des suites de 0 et de 1 sans plus de 4 symboles égaux consécutifs. Donner les automates avec sortie pour le codage et pour le décodage, en utilisant un symbole de fin de mot.

Question 4. Proposer un autre codage qui satisfait toutes les propriétés voulues et qui n'utilise que des mots de 2 ou trois lettres.

Problème 3: 8 points

On étudie ici une famille de grammaires indéxées par un entier n et notées $\mathcal{G}_n$. Ces grammaires ont un alphabet terminal composé de $2n$ lettres notées $a_1, a_2, \ldots, a_n$ et $b_1, b_2, \ldots, b_n$. Il y a aussi $n+1$ lettres dans l'alphabet auxiliaire, ce sont : S l'axiome et $A_1, A_2, \ldots, A_n$. Les règles de la grammaire sont au nombre de $n + 2n(n-1)$, les n premières sont données par:

$$S \rightarrow A_i b_i, \quad 1 \leq i \leq n$$

et il y a $2(n-1)$ règles pour chaque lettre A_i qui sont:

$$A_i \rightarrow a_j A_i, \quad A_i \rightarrow a_j \quad 1 \leq j \neq i \leq n$$

On considère d'abord le cas $n = 2$, il y a alors 6 règles:

$$S \rightarrow A_1 b_1 \quad S \rightarrow A_2 b_2 \quad A_1 \rightarrow a_2 A_1 \quad A_1 \rightarrow a_2 \quad A_2 \rightarrow a_1 A_2 \quad A_2 \rightarrow a_1$$

Question 1. On vous demande de calculer les ensembles de 0-items de cette grammaire, et l'automate qui reconnaît l'union des $R_0(i)$ pour $1 \leq i \leq 6$. On notera les ensembles de 0 items de la façon suivante:

I_0 est initial, et pour $i = 1, 2$: $I_i = \Delta(I_0, A_i)$, $I'_i = \Delta(I_i, b_i)$, $J_i = \Delta(I_0, a_i), K_i = \Delta(J_j, A_i)$, pour $j \neq i$.

Question 2. Montrer que la grammaire $\mathcal{G}_2$ n'est pas $LR(0)$, est elle $SLR(1)$?

On considère maintenant $n > 2$ quelconque

Question 3. Pour chacun des mots suivants dire s'ils appartiennent au langage L_3 engendré par $\mathcal{G}_3$

$$a_1a_3b_2 \quad b_1a_3b_2 \quad a_1a_3b_1 \quad a_1a_3b_3$$

Donner une caractérisation des mots engendrés par la grammaire $\mathcal{G}_n$

Question 4. On adopte ici la notation suivante pour les ensembles de 0-items de $\mathcal{G}_n$: $I_i = \Delta(I_0, A_i)$, $I'_i = \Delta(I_i, b_i)$, $J_i = \Delta(I_0, a_i), K_i = \Delta(J_j, A_i)$, puis pour chaque mot $f = a_{i_1}a_{i_2} \ldots a_{i_p}$ composé de lettres toutes différentes On note $J_{i_1,i_2,\ldots,i_p}$ l'état $\Delta(I_0, f)$.

Soit X et Y deux sous-ensembles de $\{1, 2, \ldots, n\}$ et soit un entier j appartennant à X et pas à Y, donner un 0-item qui appartient à J_Y et pas à J_X.

Question 5. Déduire de ce qui précède qu'une grammaire ayant m règles de grammaire peut donner lieu à un automate reconnaissant l'union des $R_0(i)$ ayant un nombre d'états supérieur à tout polynôme en m.