Université Paris 7-Denis Diderot
L3
d'Informatique

Examen du xx mai 2007
— Durée 3 h —
sans documents

Avertissement :
Questions de cours, notées sur 7 points : on demande d'être clair, précis et concis.
Exercice, noté sur 3 points : Il est demandé une brève explication de ce qui est à faire et de faire les calculs en les présentant de manière claire et commentée
Problème, noté sur 10 points : Il est demandé de faire des preuves rigoureuses. Une explication en français correct, claire et précise de ce qui est à faire et de comment le faire pour mener à bien une preuve sera appréciée. On peut évidemment sauter une question en indiquant qu'on admet le résultat.

1 Questions de cours

Qu'est-ce qu'une relation réflexive et transitive ?
Qu'est-ce que la fermeture réflexive et transitive d'une relation ?
Si S désigne la relation successeur sur $\mathbb{Z}$, que vaut la fermeture réflexive et transitive de S ?
Quand dit-on d'une relation qu'elle est une relation d'ordre ?
Qu'est-ce qu'un bon ordre ?
Expliquer pourquoi S^* est un ordre qui n'est pas un bon ordre.
Définir sur $\mathbb{Z}$ un bon ordre.

2 Exercice

On considère la grammaire algébrique $G = < A, V, P >$ où $A = \{a, b, c\}$, $V = \{x, y, z, t, u, v\}$ et P est défini par :

$$\begin{array}{lcl} x & \longrightarrow & auv + xy + ybz \\ y & \longrightarrow & zz + xt \\ z & \longrightarrow & tz + yt + atct + tt \\ t & \longrightarrow & att + \varepsilon \\ u & \longrightarrow & aubv \\ v & \longrightarrow & uct + \varepsilon \end{array}$$

Réduire cette grammaire vis à vis de x.
Rendre propre la grammaire obtenue.

3 Problème

Soit $G =< A, V, P >$ une grammaire algébrique dont on suppose qu'aucune variable n'engendre un langage vide. A toute variable $v \in V$ on associe les ensembles :

- $Prem(v) = \{a \in A \mid \exists w \in A^* \ : v \longrightarrow^* aw\}$
- $Dern(v) = \{a \in A \mid \exists w \in A^* \ : v \longrightarrow^* wa\}$

et à toute lettre $a \in A$ on associe l'ensemble :

- $Suiv(a) = \{b \in A \mid \exists w_1, w_2 \in A^*, \exists v \in V \ : v \longrightarrow^* w_1abw_2\}$

On suppose dans un premier temps (jusqu'à la question 8) que la grammaire G est pré-propre, *i.e.* qu'elle ne contient aucune règle de membre droit égal au mot vide ε (mais peut contenir un membre droit qui est une lettre de V).

Question 1 : Donner, en justifiant votre réponse, une valeur d'une borne *a priori* M telle que $Prem(v) = \{a \in A \mid \exists w \in A^*, \exists p \leq M \ : v \longrightarrow^p aw\}$.

Question 2 : En déduire un algorithme permettant de calculer les ensembles $Prem(v)$ et $Dern(v)$.

Question 3 : Démontrer que s'il existe un membre droit de règle m qui s'écrit $m = m_1avm_2$, alors $b \in Prem(v) \Longrightarrow b \in Suiv(a)$.
En admettant que pour tout $v \in V$, $Prem(v)$ et $Dern(v)$ sont effectivement calculables, démontrer que pour tout $a \in A$ $Suiv(a)$ l'est aussi.

Question 4 : Donner, en justifiant votre réponse, une valeur d'une borne *a priori* M' telle que $Suiv(a) = \{b \in A \mid \exists w_1, w_2 \in A^*, \exists v \in V, \exists p \leq M' \ : v \longrightarrow^p w_1abw_2\}$.

Soit n le nombre de variables de G et notons E l'ensemble des n-uples de parties de A. A tout n-uple $t = (A_1, \ldots, A_n)$ de parties de A, on associe le n-uple $t' = (A'_1, \ldots, A'_n)$ où $A'_i = A_i \cup \{a \in A \mid \exists j : a \in A_j \ et \ \exists v_i \longrightarrow v_jm \in P\}$.

Question 5 : Démontrer que l'application φ qui à t associe t' est une application croissante et continue de E dans lui-même. En se souvenant que les n-uples de parties d'un ensemble constitue un treillis complet, en déduire que, pour tout n-uple $t_0 = (B_1, \ldots, B_n) \in E$, φ admet un plus petit point fixe qui contient t_0.

On pose $t_0 = (B_1, \ldots, B_n)$ où $B_i = \{a \in A \mid \exists v_i \longrightarrow am \in P\}$.

Question 6 : Calculer t_0, puis calculer ce plus petit point fixe contenant t_0 pour la grammaire particulière $G =< A, V, P >$ où $A = \{a, b, c\}$, $V = \{x, y, z\}$ et P est défini par :

$$\begin{array}{lcl} x & \longrightarrow & ax + xy + ybz \\ y & \longrightarrow & zz + xt \\ z & \longrightarrow & yz + axcy + cc \end{array}$$

Question 7 : En déduire un nouvel algorithme permettant de calculer $Prem(v)$ et en donner la complexité.

Question 8 : Si on ne suppose plus maintenant que la grammaire G est pré-propre, que faut-il changer pour la résolution des questions précédentes ?
Application : Calculer $Suiv(a)$, pour tout $a \in A$, pour la grammaire G précédente à laquelle on a ajouté la règle $z \longrightarrow \varepsilon$.