Ensimag 1ère année mai 2014

Théorie des langages 2

Durée : 3h.
Documents : tous documents autorisés.

Exercice (6 points) Calculabilité

▷ **Question 1 (2 points)** On considère un vocabulaire Σ ayant au moins 2 éléments ; tous les langages dont on parle sont des sous-ensembles de Σ^*. En particulier, le complémentaire d'un langage L s'entend comme complémentaire dans Σ^*.

Les affirmations suivantes sont-elles vraies ou fausses ? Justifiez... Chaque bonne réponse justifiée vaut $\frac{1}{4}$ point, chaque bonne réponse non (ou mal) justifiée vaut $\frac{1}{8}$ point, chaque mauvaise réponse vaut $-\frac{1}{4}$ point, une absence de réponse vaut $-\frac{1}{8}$ point, le total sur la question ne pouvant pas être négatif.

1.1 si L et M sont non récursifs, alors $L \cap M$ est non récursif

1.2 si L et M sont récursifs, alors $L \cup M$ est récursif

1.3 si L est non récursif, alors son complémentaire est non récursif

1.4 si L est fini, alors son complémentaire est récursif

1.5 si $L \subseteq M$ et M est récursif, alors L est récursif

1.6 une réduction est une application injective

1.7 une réduction est une application surjective

1.8 on sait simuler une MTN par une MTD, avec une perte de coût quadratiaque

▷ **Question 2 (4 points)** Soit L un langage récursivement énumérable.

2.1 (2 points)
Montrer que si, $\forall n \in \mathbb{N}$ L contient *exactement un* mot de longueur n, alors L est récursif.

2.2 (1 point)
Que peut-on dire si en question 2.1 on remplace "*exactement un*" par "*exactement n*" ?

2.3 (0,5 point)
Que peut-on dire si en question 2.1 on remplace "*exactement un*" par "*au plus n*" ?

2.4 (0,5 point)
Que peut-on dire si en question 2.1 on remplace "*exactement un*" par "*au moins n*" ?

Problème (14 points) Langages hors-contexte et reconnaisseurs

On s'intéresse à un sous-langage des expressions entières du langage C. Soit G_1 la grammaire suivante, E_1 étant l'axiome :

1) $E_1 \rightarrow E_1 + E_2$
2) $E_1 \rightarrow E_2$
3) $E_2 \rightarrow (E_1)$
4) $E_2 \rightarrow$ *place*
5) *place* $\rightarrow$ **idf**
6) *place* $\rightarrow$ ++ **idf**
7) *place* $\rightarrow$ **idf** ++

Le vocabulaire terminal est VT_1={**idf**, +, (,), ++}. Les notations ont leur sens habituel. On rappelle la sémantique des opérateurs de pré-incrémentation et de post-incrémentation : 1) l'expression ++**a** où **a** vaut **n** a pour valeur **n+1** et cette nouvelle valeur est effectée à **a**, 2) l'expression **a**++ où **a** vaut **n** a pour valeur **n** et la valeur **n+1** est affectée à **a**. Attention, + et ++ sont deux symboles distincts du vocabulaire terminal.

▷ **Question 3 (2 points)**
Donnez une grammaire LL(1) pour le langage $L(G_1)$. On prouvera le caractère LL(1) de la grammaire proposée.

▷ **Question 4 (2 points)**
Une expression a des effets de bord si l'évaluation de cette expression entraine une modification de la mémoire. Donnez un calcul d'attributs **sur la grammaire d'origine** G_1 permettant de calculer si une expression peut provoquer ou non un effet de bord (on utilisera la valeur **true** pour les effets de bord et **false** pour les expressions sans effet de bord).

▷ **Question 5 (3 points)**
Ecrire en Ada un analyseur LL(1) pour la grammaire de la question 2 donnant en résultat un booléan indiquant si l'expression analysée contient, ou non, des effets de bord. **On utilisera la méthode d'écriture d'analyseurs vue en cours. On écrira toutes les procédures d'analyse sauf la procédure associée au non-terminal E_2 dont on précisera les paramètres, si besoin est.** On suppose donnée la fonction `lire_mot return Token` avec `type Token is (Idf, Plus, Par_Ouv, Par_Ferm, PlusPlus, Dollar);` `Dollar` indiquant la fin de fichier.

▷ **Question 6 (2 points)**
On ajoute dans le langage l'opérateur d'affectation = en étendant la grammaire G_1 par les deux règles suivantes :

a) $E_0 \rightarrow$ *place* $= E_0$
b) $E_0 \rightarrow E_1$

et en remplaçant la règle 3) par $E_2 \rightarrow (E_0)$.

E_0 est maintenant l'axiome et $VT_0 = VT_1 \cup \{=\}$. Soit G_0 cette grammaire. On rappelle qu'une affectation de la forme *place* = *exp*, vue comme une expression, réalise l'affectation et produit la valeur de l'expression.

1. Quelle est la priorité de l'opérateur = par rapport à l'opérateur + ?
2. Donner une grammaire LL(1) pour le langage $L(G_0)$. On prouvera son caractère LL(1).

▷ **Question 7 (2 points)**

Les expressions avec effet de bord peuvent être problématiques car elles sont sensibles à l'ordre d'évaluation. Certaines normes de programmation restreignent leur utilisation : une expression ne peut contenir qu'un effet de bord au plus haut niveau de l'expression.

Exemples admis : **x++** **x=y+7**

Exemples interdits : **x++ + z** **(x=y)+z** **x=y=z+z**

Proposer un calcul d'attributs sur la grammaire G_0 permettant de vérifier cette restriction.

▷ **Question 8 (3 points)**

Soit G une grammaire quelconque LL(1), d'axiome S et telle que $\epsilon \notin L(G)$. Montrer que le langage $(L(G))^+$ est LL(1). Que se passe t-il si $\epsilon \in L(G)$? Peut-on en déduire quelque chose sur le caractère LL(1) du langage $(L(G))^+$?