Langages formels, calculabilité et complexité

Examen

31 janvier 2013, durée 3h, sans documents

1 On applique les cours

Exercice 1 – *Induction structurelle*

Démontrez **par l'induction structurelle** sur l'expression régulière le lemme de pompage en forme suivante.

Lemme 1 *Si un langage* L *est définissable par expression régulière, alors*
- *soit* L *est fini ;*
- *soit il existe 3 mots* u, $v \neq \varepsilon$ *et* w *tels que* $\forall k\ uv^kw \in L$.

Exercice 2 – *Automates : des petits calculs*

On considère le langage L sur $\Sigma = \{a, b, c\}$ défini par l'expression régulière $a(b+c)^*a$.

1. Calculez son monoïde syntaxique (les éléments et la table de multiplication).
2. Est L sans étoile ? Justifiez.
3. Définissez-le en logique FO ou MSO.

Exercice 3 – *Calculabilité : une petite analyse*

On considère le prédicat P sur $\mathbb{N}$ défini comme suit

$$P(x) \rightleftharpoons (\varphi_x(x) = x + 5)$$

(en prose : $P(x)$ est vrai si la machine de Turing de numéro x sur l'entrée x donne le résultat $x+5$).

1. Est-ce qu'on peut y appliquer le théorème de Rice ?
2. Est P décidable ? Semi-décidable ? Justifiez.

2 On réfléchit un peu plus

Exercice 4 – *Une devinette*

Démontrez qu'il existe un langage régulier fini G sur $\{a, b\}$ tel que

– tous les mots de G contiennent < 100 lettres ;

– ET tout automate qui reconnaît G contient > 1000000 états

Indication: comptez

Exercice 5 – *Machines multiplicatives*

On considère une classe de machines (ou programmes) très simple. La *machine multiplicative* a un seul registre R capable de stocker un entier naturel positif. Son programme est un ensemble fini d'instructions de types suivants (a et b sont des constantes naturelles positives, on autorise l'utilisation de plusieurs constantes différentes)

```
- q: R := R * a; goto p
- q: R := R/a; goto p
- q: if R mod a = b then goto p else goto t
- q: Stop
```

Une tentative de division impossible dans $\mathbb{N}$ (telle que 5/3) produit une erreur. On étudie le problème de l'arrêt pour cette classe de machines ArretMult(M, R_0) : "Est-ce que la machine multiplicative M en démarrant avec R_0 dans le registre s'arrête (sans erreurs) ?"

1. Pour apprendre à programmer un peu, donnez une machine multiplicative qui à partir de chaque R initial de la forme $R = 2^n$ s'arrête avec $R = 5 \cdot 7^n$.
2. Prouvez que ArretMult est indécidable. *Indication: Simulez la machine à compteurs.*

3 On essaye d'être créatif

Exercice 6 – *Les réels - exercice ouvert*

Les langages réguliers (et hors contexte) vus en cours sont des sous-ensembles du monoïde Σ^*. Dans cet exercice on essaye de faire une théorie similaire pour les sous-ensembles du monoïde $\mathbb{R}_+$ de réels non-négatifs $[0; +\infty]$ munis de l'opération $+$. Pour mieux comprendre les définitions on considère trois exemples de sous-ensembles de $\mathbb{R}_+$ ("languages") :

$$L_1 = [4; 5]; \quad L_2 = \mathbb{N}; \quad L_3 = [0; 1] \cup [2; 3] \cup [4; 5] \cup \ldots$$

1. On dit que $L \subset \mathbb{R}_+$ est Myhill-Nerode-reconnaissable, s'il y a un nombre fini de quotients différents $x \backslash L = \{y | x + y \in L\}$. Est-ce que L_1, L_2, L_3 le sont ?
2. Trouvez tous les $L \subset \mathbb{R}_+$ Myhill-Nerode-reconnaissables.
3. Inventez des expressions régulières pour les sous-ensembles de $\mathbb{R}_+$, décrivez leur syntaxe et sémantique. Est-ce que L_1, L_2, L_3 peuvent être définis par vos expressions ? Y a-t-il un analogue du théorème de Kleene ?
4. Inventez des grammaires hors contexte pour les sous-ensembles de $\mathbb{R}_+$. Est-ce que L_1, L_2, L_3 peuvent être engendrés par vos grammaires ? Comaparez le pouvoir expressif de vos grammaires avec vos expressions régulières.