

TD3 Licence INF208 Théorie des langages, 2006/07

Exo 1

On considère les arbres binaires t_n ($n \geq 0$) définis inductivement par $t_0 = a$ et $t_{n+1} = f(t_n, t_n)$.

1. Quelle est la taille $|t_n|$ de l'arbre t_n (la taille correspondant au nombre des nœuds) ?
2. Construisez un automate d'arbre \mathcal{A}_n avec le moins d'états possible, qui reconnaît $\{t_n\}$.

Exo 2

Soit $\Sigma = \Sigma_0 = \Sigma_2 = \{a, b, c\}$.

1. Construisez un automate d'arbre qui reconnaît l'ensemble des arbres binaires sur Σ , dont la branche la plus à droite est étiquetée par $(a + b + c)^*abb(a + b + c)^*$.
2. Construisez un automate d'arbre montant qui reconnaît l'ensemble des arbres binaires sur Σ , dont la frontière ne contient pas le facteur ab .

Exo 3

On considère des arbres binaires (chaque nœud a 0 ou 2 fils) dont les nœuds sont étiquetés par le même symbole. Les feuilles sont étiquetées par des lettres d'un alphabet Σ .

1. Construisez des formules de MSOL décrivant les langages d'arbres suivants :
 - L'ensemble des arbres dont la frontière commence par la lettre $a \in \Sigma$.
 - L'ensemble des arbres dont la frontière ne contient pas le facteur ab , où $a, b \in \Sigma$.
2. En utilisant les questions précédentes et l'exercice 1 du TD2, montrer qu'étant donné un langage rationnel de mots $L \subseteq \Sigma^*$, l'ensemble des arbres dont la frontière est dans L est un langage d'arbres reconnaissable.

Exo 4

1. Soit \mathcal{A} un NFA (sur les mots) à n états tel que $\mathcal{L}(\mathcal{A}) \neq \emptyset$. Donnez une majoration (la meilleure possible) de $\min\{|w| \mid w \in \mathcal{L}(\mathcal{A})\}$.
2. Même question que 1), si \mathcal{A} est un NTA à n états sur l'alphabet $\Sigma = \Sigma_0 \cup \Sigma_2$ (arbres binaires).
3. Soit $\Sigma = \cup_{k=0} \Sigma_k$ un alphabet et \bullet un symbole n'appartenant pas à Σ . Un *contexte* C sur Σ est un arbre de $T(\Sigma \cup \{\bullet\})$, qui contient \bullet une seule fois, en tant qu'étiquette d'une feuille. Si $t \in T(\Sigma)$, alors $C[t]$ désigne l'arbre de $T(\Sigma)$ obtenu en remplaçant \bullet par t .
Soit \mathcal{A} un NTA à n états, et $t \in \mathcal{L}(\mathcal{A})$ de hauteur supérieure à n . Montrez qu'on peut trouver des contextes C, D ainsi qu'un arbre t' tels que $t = C[D[t']]$ et $C[\underbrace{D[D[\dots D[t'] \dots]]}_m] \in \mathcal{L}(\mathcal{A})$
pour tout $m \geq 0$.
4. Donnez un algorithme (de complexité la meilleure possible) qui décide si le langage reconnu par un NTA est infini.

Exo 5

1. Montrez que l'ensemble des arbres binaires dont toutes les feuilles ont la même profondeur, n'est pas reconnaissable.

2. Montrez que l'ensemble des arbres binaires dont toutes les feuilles ont une profondeur paire, est reconnaissable (*Remarque* : Vous pouvez soit construire un automate d'arbre, ou une formule MSOL pour ce langage).
3. Montrez que l'ensemble des arbres binaires de hauteur paire, n'est pas reconnaissable.

Exo 6

Soit $\Sigma = \Sigma_0 = \Sigma_2 = \{a, b\}$. Montrez qu'il n'existe pas d'automate d'arbre descendant *déterministe*, qui reconnaît l'ensemble des arbres binaires qui contiennent au moins une feuille a .

Exo 7

Un arbre ordonné d'arité quelconque peut être décrit par un ensemble V (fini) de nœuds, ainsi que les 3 fonctions suivantes :

- racine : $V \rightarrow \{true, false\}$: racine(v) = $true$ ssi v est la racine de l'arbre,
- premier : $V \rightarrow V$: premier(v) = w ssi w est le premier enfant de v ,
- suivant : $V \rightarrow V$: suivant(v) = w ssi v, w ont le même parent, et w est le successeur de v dans la fratrie.

L'arbre binaire associé à $t = (V, \text{racine}, \text{premier}, \text{suivant})$ est l'arbre $\text{bin}(t)$ dont l'ensemble de nœuds et la racine sont les mêmes que pour t ; pour chaque $v \in V$, le fils gauche de v est premier(v), et le fils droit est suivant(v) (s'ils existent).

1. Soit $v \in V$ et w le n -ème enfant de v dans t . Quel nœud dans $\text{bin}(t)$ correspond à w ?
2. Montrez que $\text{bin}(\cdot)$ est un codage d'arbre ordonnés par des arbres binaires (c'est-à-dire, que $\text{bin}(t) = \text{bin}(t')$ implique $t = t'$).

Exo 8

Soit $\Sigma = \Sigma_0 \cup \Sigma_2$. Une *congruence* sur $T(\Sigma)$ est une relation d'équivalence \equiv qui satisfait la propriété suivante : pour tout $a \in \Sigma_2$ et $t_1 \equiv t'_1, t_2 \equiv t'_2$, on a aussi $a(t_1, t_2) \equiv a(t'_1, t'_2)$.

1. Montrez qu'une relation d'équivalence \equiv sur $T(\Sigma)$ est une congruence ssi pour tout contexte $C, t \equiv t'$ implique $C[t] \equiv C[t']$.
2. Soit \mathcal{A} un NTA, et Q l'ensemble des états. On définit la relation $\sim_{\mathcal{A}}$ sur $T(\Sigma)$ par $t \sim_{\mathcal{A}} t'$ si $t \xrightarrow{*} q \Leftrightarrow t' \xrightarrow{*} q$, pour tout $q \in Q$.

Montrez que $\sim_{\mathcal{A}}$ est une congruence (d'indice fini) sur $T(\Sigma)$.

Exo 9

Soit $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$. On définit le codage d'un arbre $t \in T(\Sigma)$ par le mot $c(t)$ sur l'alphabet $\Gamma = \Sigma \cup \{[,]\}$ inductivement :

- Si $t = a$ alors $c(t) = a$.
- Si $t = a(t_1, t_2)$, alors $c(t) = a [c(t_1)] [c(t_2)]$.
- Si $t = a(t_1)$, alors $c(t) = a [c(t_1)]$.

1. Montrez que $c(\cdot)$ est un codage.
2. Montrez que $\{c(t) \mid t \in T(\Sigma)\}$ est un langage hors-contexte.