Magistère d'Informatique et de Mathématiques
ENS de Cachan *Mercredi 29 Mars 2006*
TD de Théorie des Langages

Automates d'arbres (3)

Propriétés de clôture.

Exercice 1 (Opérations booléennes)
Montrer que la classe des langages d'arbres reconnaissables est close par les opérations booléennes (union, intersection et complémentaire). Proposer des constructions préservant le déterminisme ascendant.

Exercice 2 (Homomorphismes d'arbres)
Soit $\mathcal{F}$ et $\mathcal{F}'$ deux ensembles de symboles de fonctions, éventuellement non disjoints. Pour tout $n > 0$ tel que $\mathcal{F}$ contient un symbole d'arité n, nous définissons un ensemble de variables $\mathcal{X}_n = \{x_1, \ldots, x_n\}$.
Soit $h_\mathcal{F}$ une application qui, à tout élément $f \in \mathcal{F}$ d'arité n, associe un terme $t_f \in T(\mathcal{F}', \mathcal{X}_n)$.
L'homomorphisme d'arbres $h : T(\mathcal{F}) \to T(\mathcal{F}')$ déterminé par $h_\mathcal{F}$ est défini inductivement par :
- $h(a) = t_a \in T(\mathcal{F}')$ pour tout $a \in \mathcal{F}$ d'arité 0,
- $h(f(t_1, \ldots, t_n)) = t_f\{x_1 \leftarrow h(t_1), \ldots, x_n \leftarrow h(t_n)\}$.

Un homomorphisme est dit linéaire si toutes les images de l'application associée sont des termes linéaires.
Exemple : On considère les deux ensembles $\mathcal{F} = \{g(3), a(0), b(0)\}$ et $\mathcal{F}' = \{f(2), a(0), b(0)\}$.
On définit l'application suivante :

$$h : \begin{cases} \mathcal{F} & \to & T(\mathcal{F}', \mathcal{X}_3) \\ g & \mapsto & f(x_1, f(x_2, x_3)) \\ a & \mapsto & a \\ b & \mapsto & b \end{cases}$$

Calculer l'image par l'homomorphisme d'arbres associé à h du terme $t = g(a, g(b, b, b), a)$.
Démontrer les propriétés suivantes :

1. les homomorphismes préservent par image réciproque la reconnaissabilité.
2. les homomorphismes linéaires préservent (par image directe) la reconnaissabilité,

Donner un exemple d'homomorphisme non linéaire ne préservant pas la reconnaissabilité par image directe.

Algorithmique.

Dans cette partie nous nous intéressons à des questions de complexité. Définissons la taille $||t||$ d'un arbre t par induction :

$$\begin{cases} ||t|| = 1 \text{ si } t \text{ est une feuille} \\ ||t|| = 1 + \sum_{i \in \{1,\ldots,n\}} ||t_i|| \text{ sinon} \end{cases}$$

Définissons également la taille d'un automate d'arbre $\mathcal{A} = (Q, \Sigma, Q_f, \Delta)$:

$$||\mathcal{A}|| = |Q| + \sum_{(q_1,\ldots,q_n)\xrightarrow{\sigma} q \in \Delta} n + 2$$

Exercice 3 (Automates d'arbres émondés)
Comment définir le caractère "émondé" pour un automate d'arbre ?
Proposer un algorithme pour émonder un automate d'arbre.
Quel est le coût de votre algorithme ?

Exercice 4 (Déterminisation)
Étudier le coût de la déterminisation des automates d'arbres.

Exercice 5 (Problèmes de décision)
Nous nous intéressons dans cet exercice à des problèmes de décision classique.
- Problème du mot : Un automate d'arbre $\mathcal{A}$ est fixé. On souhaite décider, étant donné un arbre t en entrée, si t est reconnu par $\mathcal{A}$. Montrer que l'on peut résoudre ce problème en temps linéaire.
- Problème de l'appartenance : On considère en entrée un automate d'arbre $\mathcal{A}$ et un arbre t. On souhaite décider si t est reconnu par $\mathcal{A}$. Montrer que ce problème peut être résolu en temps linéaire si $\mathcal{A}$ est déterministe, et quadratique si $\mathcal{A}$ est non déterministe.
- Problème du vide : Étant donné un automate d'arbre $\mathcal{A}$, on souhaite décider si le langage reconnu par $\mathcal{A}$ est vide. Montrer que ce problème peut être résolu en temps quadratique.
- Problème de la finitude : Étant donné un automate d'arbre $\mathcal{A}$, on souhaite décider si le langage reconnu par $\mathcal{A}$ est fini. Montrer que ce problème peut être résolu en temps quadratique.

Résiduels.

Exercice 6 (Résiduels d'automates d'arbres)
Nous définissons deux classes de résiduels pour les langages d'arbres. Soit $\mathcal{L}$ un langage d'arbres sur Q.
- Soit t un Q-arbre à trou. Le résiduel à gauche de $\mathcal{L}$ par t est défini par :

$$t^{-1}\mathcal{L} = \{u \in \{Q - \text{arbre}\} \mid t.u \in \mathcal{L}\}$$

- Soit t un Q-arbre. Le résiduel à droite de $\mathcal{L}$ par t est défini par :

$$\mathcal{L}t^{-1} = \{u \in \{Q - \text{arbre à trou}\} \mid u.t \in \mathcal{L}\}$$

Montrer qu'un langage d'arbres reconnaissables possède un nombre fini de résiduels à gauche et un nombre fini de résiduels à droite.
Étudier la réciproque.