

Tree Automata and Applications: TD6 bonus *DTDs*

Emile Contal

<http://econtal.perso.math.cnrs.fr/teaching>

October 25, 2013

Exercice 1. Donner un langage d'arbres reconnaissable, qui n'est accepté par aucun DTD

Si e est une expression régulière, on marque les occurrences des lettres dans l'expression de manière à ce que chaque lettre n'apparaisse qu'une fois. Formellement, si f est une fonction de l'alphabet A dans \mathbb{N} , 0 est la fonction nulle, et $f + a$ est la fonction $f + a(a) = f(a) + 1$ et $f + a(b) = f(b)$ si $b \neq a$.

à une expression régulière e et à une fonction f on associe la paire $m(e, f) = (e', f')$ définie par récurrence par:

- $m(\epsilon, f) = (\epsilon, f)$
- $m(e^*, f) = ((e')^*, f')$ si $m(e, f) = (e', f')$
- $m(e_1 + e_2, f) = (e'_1 + e'_2, f'')$ si $m(e_1, f) = (e'_1, f')$ et $m(e_2, f) = (e'_2, f'')$
- $m(e_1 \cdot e_2, f) = (e'_1 \cdot e'_2, f'')$ si $m(e_1, f) = (e'_1, f')$ et $m(e_2, f) = (e'_2, f'')$
- $m(a, f) = (a_{f(a)}, f + a)$

Enfin, on note e' l'expression $m(e, 0)$, sur l'alphabet $A \times \mathbb{N}$ (la deuxième composante étant notée en indice). Par exemple

$$(((a + b)^* a (ab)^*)')' = ((a_0 + b_0)^* a_1 (a_2 b_1)^*)'$$

Une expression régulière e est *déterministe* si, pour tous $ua_iw, ub_jx \in L(e')$ où $a_i, b_j \in A \times \mathbb{N}$, $a_i \neq b_j$ entraîne $a \neq b$.

Exercice 2. Un DTD est dit *déterministe* si ses contenus sémantiques sont définis par des expressions déterministes.

1. Montrer que pour toute expression régulière déterministe on peut construire en temps polynomial un automate déterministe qui accepte le même langage
2. Montrer que les problèmes d'appartenance, du vide et de l'inclusion sont décidables en temps polynomial pour les DTD déterministes.
3. Donner un exemple de langage régulier qui ne peut pas être défini par une expression régulière déterministe.

Un alphabet *étendu* est un alphabet fini A_e avec une application surjective σ de A_e dans A . (L'image réciproque d'une étiquette de A sont les différents *types* d'étiquettes).

Un DTD *étendu* (EDTD) est un DTD sur l'alphabet étendu A_e . Un arbre t étiqueté par A est accepté par un EDTD s'il est l'image homomorphe d'un arbre accepté par l'EDTD sur l'alphabet A_e . Autrement dit, il existe un étiquetage des noeuds de t par des types, de telle manière à ce que l'arbre obtenu soit accepté par l'EDTD.

Exercice 3. Montrer qu'un langage d'arbres est reconnaissable ssi c'est le langage engendré par un EDTD.

Montrer que le problème d'appartenance à un EDTD se résoud en temps polynomial.

Un EDTD est *uniquement typé* si, pour chacune des règles de grammaire, la restriction de σ aux symboles apparaissant dans le membre droit de la règle est une injection.

Exercice 4. Donner un exemple d'EDTD uniquement typé et un exemple d'EDTD qui n'est pas uniquement typé.

Montrer que le complémentaire d'un EDTD déterministe et uniquement typé peut être calculé en temps polynomial.