TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Theoretical Computer Science, Chair of Automata Theory

Automata and Logic

Exercise Sheet 11

Prof. Dr.-Ing. Franz Baader
Summer Semester 2012

Exercise 53

Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ be an alphabet with arity function, where $\Sigma_0 = \{x, y, z\}$, $\Sigma_1 = \{\neg\}$, and $\Sigma_2 = \{\wedge, \vee\}$. Define tree automata (either LR or RL) recognising the tree languages consisting of the following trees:

a) trees containing the symbol $\vee$ exactly once;

b) trees containing the symbol $\neg$ at least once on every path of the tree; and

c) trees describing *satisfiable* propositional formulae.

Exercise 54

Let $\mathcal{A} = (Q, \Sigma, I, \Delta, F)$ be the non-deterministic LR-tree automaton given by:

- $Q = \{0, 1\}$;
- $\Sigma = \{f, x, y\}$ with $\nu(f) = 2$ and $\nu(x) = \nu(y) = 0$;
- $I(x) = \{0, 1\}$ and $I(y) = \{0\}$;
- $\Delta_f(0, 0) = \{0\}$, $\Delta_f(0, 1) = \{1, 0\}$, $\Delta_f(1, 0) = \{1, 0\}$, and $\Delta_f(1, 1) = \{1\}$; and
- $F = \{1\}$.

Do the following:

a) Adapt the standard powerset construction from finite automata on words to LR-tree automata and use it to construct a deterministic LR-tree automaton $\mathcal{A}'$ such that $L(\mathcal{A}) = L(\mathcal{A}')$.

b) Try to apply a similar construction to the RL-tree automaton from Example 6.10 from the lecture. Explain why this method fails for RL-tree automata.

Exercise 55

Example 6.10 from the lecture shows that deterministic RL-tree automata recognise a smaller class of languages than non-deterministic ones. We call an RL-tree automaton $\mathcal{A} = (Q, \Sigma, I, \Delta, F)$ *quasi-deterministic* if

- Δ is a deterministic transition assignment, and
- $I \subseteq Q$ is a *set* of initial states.

Prove or refute:

a) If $L \subseteq \mathbf{T}_\Sigma$ is a *finite* tree language, then there exists a quasi-deterministic tree automaton recognising L.

b) If $L \subseteq \mathbf{T}_\Sigma$ is a *recognisable* tree language, then there exists a quasi-deterministic tree automaton recognising L.

Exercise 56

Devise a quadratic-time algorithm that decides the emptiness problem for LR-tree automata.

Exercise 57

Let $\mathcal{A} = (Q, \Sigma, I, \Delta, F)$ be an RL-tree automaton given by:

- $Q = \{1, \dots, 4\}$;
- $\Sigma = \{g, n, a, b\}$ with $\nu(g) = 2$, $\nu(n) = 1$, and $\nu(a) = \nu(b) = 0$;
- $I = \{1\}$;
- $\Delta_g(1) = \{(1, 1), (1, 2), (3, 4), (4, 1)\}$, $\Delta_g(2) = \emptyset$, $\Delta_g(3) = Q \times Q$, $\Delta_g(4) = \{(1, 2), (1, 4), (2, 4), (2, 2)\}$;
- $\Delta_n(1) = \{1\}$, $\Delta_n(2) = \{3\}$, $\Delta_n(3) = \{1, 2\}$, $\Delta_n(4) = \{1, 3\}$; and
- $F(a) = \{2\}$, $F(b) = \{2, 3\}$.

Decide whether $L(\mathcal{A}) = \emptyset$ or not.