TECHNISCHE UNIVERSITÄT DRESDEN

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

Automata and Logic

Exercise Sheet 10

Dr. rer. nat. Rafael Peñaloza
Summer Semester 2013

Exercise 49

For each of the following languages L_i, give an S1S-formula ϕ_i such that $L_\omega(\phi_i) = L_i$:

a) $L_1 = (abb^*)^\omega$,

b) $L_2 = ((aa)^+(bb)^+)^\omega$, and

c) $L_3 = (aaa)^+b(a \cup b)^\omega$.

Exercise 50

Transform the S1S-formula $P(\underline{0})$ into an equivalent $\mathrm{S1S_0}$-formula.

Exercise 51

Let $L := (a^+b)^\omega \cup (b^+a)^\omega$. Use the proof of Thm. 5.4 to construct a closed S1S-formula ϕ with $L_\omega(\phi) = L$.

Exercise 52

A *Rabin automaton* is a tuple $\mathcal{A} := (Q, \Sigma, I, \Delta, \Omega)$ where Q, Σ, I, and Δ are defined as for non-deterministic Büchi automata, and $\Omega := \{(F_1, G_1), \ldots, (F_n, G_n)\}$ is a finite set of pairs (F_i, G_i) such that $F_i, G_i \subseteq Q$. For a word α, let $\mathsf{path}_\mathcal{A}(\alpha)$ denote the set of all paths in $\mathcal{A}$ labelled with α. For a path $p \in \mathsf{path}_\mathcal{A}(\alpha)$, let $\mathsf{inf}(p)$ denote the set of all states that are visited infinitely often. The ω-language $L_\omega(\mathcal{A})$ recognised by a Rabin automaton is defined as

$$L_\omega(\mathcal{A}) := \{\alpha \in \Sigma^\omega \mid \exists i \in \{1, \ldots, n\}\ \exists p \in \mathsf{path}_\mathcal{A}(\alpha)\colon \mathsf{inf}(p) \cap F_i \neq \emptyset \wedge \mathsf{inf}(p) \cap G_i = \emptyset\}.$$

Show that every language recognised by a Rabin automaton is also recognised by a Büchi automaton by constructing for a given Rabin automaton $\mathcal{A}$, an S1S-formula $\phi_\mathcal{A}$ defining the language $L_\omega(\mathcal{A})$.

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.