Applied Automata Theory (WS 2014/2015) Technische Universität Kaiserslautern

Exercise Sheet 7

Prof. Dr. Roland Meyer, Reiner Hüchting Due: Tue, Dec 16

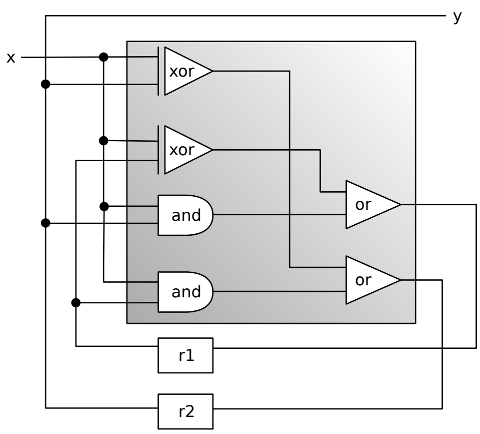
Exercise 7.1 NBA Languages = ω-regular Languages

(a) Show that if there is an NBA that accepts $L \subseteq \Sigma^\omega$ then L is ω-regular.

(b) Construct an NBA that accepts $L = (ab + c)^*((aa + b)c)^\omega + (a^*c)^\omega$

Exercise 7.2 Circuit Verification

Consider a circuit[1] that continuously receives inputs x and generates outputs y:



The circuit uses registers r_1 and r_2, which are initially $r_1 = 0$ and $r_2 = 1$.

(a) Construct a Büchi automaton over the alphabet $\{0,1\}^2$ that accepts all sequences of input/output pairs which describe the possible runs of the circuit.

Hint: The states are determined by r_1 and r_2 and the transitions only depend on x.

(b) Use the automaton to determine whether the circuit satisfies the properties ...

P_{fair}: whenever x is infinitely often high, then y is infinitely often high.

P_{safe}: always $x = y = 1$ or $x = y = 0$.

$P_{\text{persistent}}$: starting from some point, y will always be high.

(c) Give words (finite if possible) that satisfy P_i and $\neg P_i$ for each $i \in \{\text{fair}, \text{safe}, \text{persistent}\}$.

---

[1]Inspired by C. Baier & J.P. Katoen: Principles of Model Checking

Exercise 7.3 Disjunctive Well-Foundedness (optional)

A partially ordered set $(A, \leq)$ is said to be *well-founded* if for every sequence

$$a_1 \geq a_2 \geq a_3 \geq \cdots ,$$

$a_i \in A$, $i \in \mathbb{N}$, there is an $n \in \mathbb{N}$ such that $a_m = a_n$ for any $m \geq n$.

Let $T_1, \ldots, T_n \subseteq A \times A$ be well-founded partial orders and $R \subseteq A \times A$ be a partial order such that $R \subseteq T_1 \cup \cdots \cup T_n$. Show that R is well-founded, too.

Hint: Use Ramsey's Theorem.

Exercise 7.4 Disjunctive Well-Foundedness (optional)

Consider the following program over integer variables and the corresponding automaton:

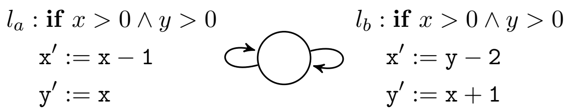
while $x > 0 \wedge y > 0$ **do**

$\mathtt{l_a}$: $(\mathtt{x}, \mathtt{y}) := (\mathtt{x} - 1, \mathtt{x})$

or

$\mathtt{l_b}$: $(\mathtt{x}, \mathtt{y}) := (\mathtt{y} - 2, \mathtt{x} + 1)$

endwhile



A state S of this program is a vector giving a value to each variable. The execution of a command l_a or l_b leads to a labelled transition between states. For example:

$$S = (x = 2, y = 1) \xrightarrow{l_a} (1, 2) = S'.$$

One can show that between every pair of states $S \xrightarrow{w} S'$, where $w \in \{l_a, l_b\}^+$, one of the following relations holds:

T_1	$x > 0 \wedge x > x'$
T_2	$x + y > 0 \wedge x + y > x' + y'$
T_3	$y > 0 \wedge y > y'$

Show that this implies termination (from any starting state).