Technische Universität München
Theoretische Informatik
Prof. J. Esparza / Dr.-Ing. J. Kreiker

Winter term 2008/2009
Exercise Sheet 13
January 27, 2009

Test Exam "Automata and Formal Languages"

- This problem set serves as a preparation for the upcoming exam. It will not be corrected, but solutions will be discussed on Monday, February 2.
- Each problem is supposed to be solvable within 15-20 minutes time.
- You may use all your notes and other written material to solve the problems, since the exam will be *open book*.
- For MSO, S1S, and LTL, you may use all abbreviations introduced in class.
- Generally, final states are drawn with double circles, while initial states are marked by an incoming arrow.
- This problem set together with problem set 12, gives a good overview of possible exam exercises. It is not an exhaustive list of all possible kinds of exercises.

Exercise 1 **(4+2 points)**

Let $L_1, L_2 \subseteq \Sigma^*$ be languages.

(a) Suppose L_1 and L_2 are accepted by DFAs $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$, respectively. Construct a DFA accepting $L_1 \setminus L_2$.

(b) Prove: If L_1 is regular, L_2 is not regular, and $L_1 \cap L_2 = \emptyset$, then $L_1 \cup L_2$ is not regular.

Exercise 2 **(3+3 points)**

Let $\Sigma = \{0,1\}$. For $w \in \Sigma^*$, where $w = a_0a_1a_2\ldots a_k$, we define $bin(w) := \Sigma_{i=0}^{k} a_i 2^i$. We define $R \subseteq \Sigma^* \times \Sigma^*$ as follows:

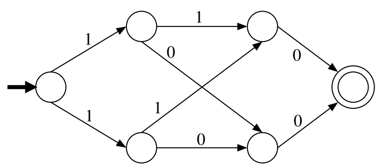
$$R = \{(w, w') \mid bin(w) > bin(w'),\ |w| = |w'|\}$$

(a) Show that R is a regular relation by finding an accepting transducer.

(b) Let $L = (\{0,1\}\{0,1\})^*$ be a language. Construct an automaton accepting $post_R(L)$.

Exercise 3 **(4+4 points)**

Consider the following NFA, A, over $\Sigma = \{0,1\}$.

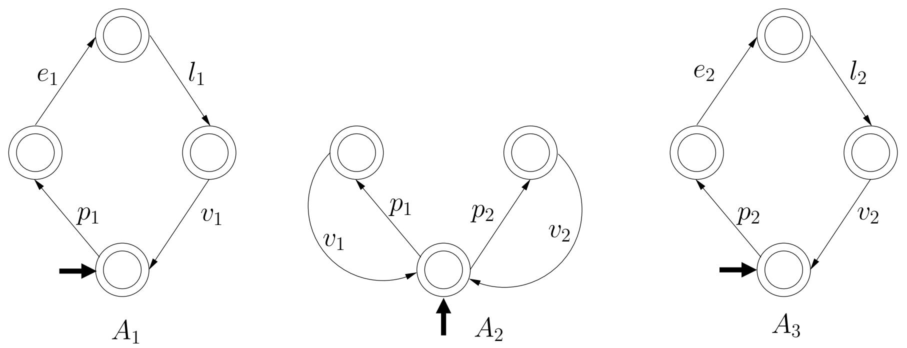


(a) Minimize A with respect to bisimilarity. Justify your solution.

(b) Construct a minimal DFA accepting $\mathcal{L}(A) \cap \{010, 011, 110, 111\}$.

Exercise 4 **(3+3 points)**

Consider the following three automata, A_1, A_2, and A_3 over the alphabets $\Sigma_1 = \{p_1, v_1, e_1, l_1\}$, $\Sigma_2 = \{p_1, v_1, p_2, v_2\}$, and $\Sigma_3 = \{p_2, v_2, e_2, l_2\}$, respectively.



(a) Construct the automaton B with $B = A_1||A_2||A_3$.

(b) Find an MSO($\Sigma_1 \cup \Sigma_2 \cup \Sigma_3$) formula, φ, such that $L(\varphi) = \mathcal{L}(B)$.

Exercise 5 **(3+3 points)**

Consider the following 2DFA, A, over $\{a, b\}$ with states $\{s_0, s_1, s_2\}$, all of which are final, and with initial state s_0. The transition table is given as follows:

	a	b
s_0	(s_0, R)	(s_1, R)
s_1	(s_1, R)	(s_2, L)
s_2	(s_0, R)	(s_2, L)

(a) Give a regular expression, r, such that $\mathcal{L}(r) = \mathcal{L}(A)$.

(b) Give a formula, ψ of MSO($\{a, b\}$) such that $L(\psi) = \mathcal{L}(A)$.

Exercise 6 **(3+3+2 points)**

Let $\Sigma = \{a, b\}$ and let $L_1, L_2 \subseteq \Sigma^\omega$ be an alphabet and two languages, such that

- $L_1 = \{\alpha \mid a \text{ occurs infinitely often in } \alpha\}$,
- $L_2 = \{\alpha \mid b \text{ occurs infinitely often in } \alpha\}$, and
- $L = L_1 \cap L_2$

(a) Find a Büchi automaton accepting L.

(b) Find a Muller automaton accepting L.

(c) Find an S1S formula, $\psi(A_a, A_b)$, such that $L_\psi = L_{\{0,1\}}$.

Exercise 7 **(2+4 points)**

Let Σ be an alphabet Let $\mathcal{A} = (S, \rightarrow, S_{in})$ be an automaton over Σ and let $\mathcal{G} \subseteq S$ be a set of states. A universal Co-Büchi automaton $(\mathcal{A}, \mathcal{G})$ accepts an infinite word, $\alpha \in \Sigma^\omega$, if for *all* runs ρ of $\mathcal{A}$ on α holds that $\mathsf{inf}(\rho) \cap F = \emptyset$. The set of all words accepted by a universal Co-Büchi automaton $(\mathcal{A}, \mathcal{G})$ is written $L^{uc}(\mathcal{A}, \mathcal{G})$.

(a) Let $\Sigma = \{a, b\}$ and assume

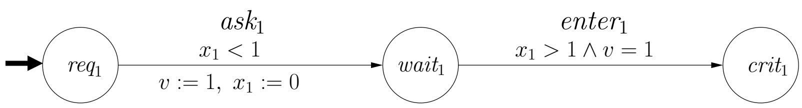
$$\mathcal{A}' = (\{q_0, q_1\}, \{(q_0, a, q_0), (q_0, b, q_0), (q_0, b, q_1), (q_1, b, q_1)\}, \{q_0\})$$
$$\mathcal{G}' = \{q_1\}$$

Give an ω-regular expression describing $\mathcal{L}^{uc}(\mathcal{A}', \mathcal{G}')$.

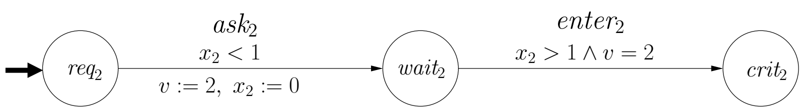
(b) Prove: $L \subseteq \Sigma^\omega$ is ω-regular if and only if it is accepted by a universal Co-Büchi automaton.

Exercise 8 **(4+2+2+2 points)**

Consider the timed automata A_1 and A_2 given below. They operate on the set $\{x_1, x_2\}$ of clocks and the shared integer variable v. A_i has actions $\{ask_i, enter_i\}$ and locations $\{req_i, wait_i, crit_i\}$ for $i = 1, 2$. Guards are given on top of edges, reset and variable updates below edges. Initially, v has value 1.



A_1



A_2

(a) Construct $A_1||A_2$.

(b) Find a timed trace of the system $A_1||A_2$ that ends with an action $enter_2$ and no more action is possible after that.

(c) Give a sequence of transitions in $A_1||A_2$ starting from the initial state, such that the actions ask_1, ask_2, and $enter_2$ occur in that order.

(d) Argue that no state of the form $(crit_1, crit_2, \ldots)$ is reachable in $A_1||A_2$.

Solutions

Exercise 1

Part (a)

Use the product construction, where final states contain F_1 states but not F_2 states. Obviously, the product construction preserves determinism. Formally, let $A = (Q_1 \times Q_2, \Sigma, \delta, (q_{01}, q02), F_1 \times (Q_2 \setminus F_2))$, where $\delta(q_1, q_2) = (\delta_1(q_1), \delta_2(q_2))$. Now assume $w = a_1 a_2 \ldots a_k$ is accepted by A. This is the case *iff* there exists a (unique) run

$$(q_{01}, q_{02}) \xrightarrow{a_1} (q_1, q_1') \xrightarrow{a_2} \ldots \xrightarrow{a_k} (q_k, q_k')$$

of A on w, where $q_k \in F_1$ and $q_k' \notin F_2$. This is in turn *equivalent* to the existence of two runs

$$\begin{array}{l} q_{01} \xrightarrow{a_1} q_1 \xrightarrow{a_2} \ldots \xrightarrow{a_k} q_k \\ q_{02} \xrightarrow{a_1} q_1' \xrightarrow{a_2} \ldots \xrightarrow{a_k} q_k' \end{array}$$

of w on A_1 and A_2. Since A_1 and A_2 are both deterministic, this is equivalent to $w \in L_1$ and $w \notin L_2$ proving the correctness of the construction.

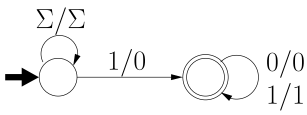
Part (b)

Suppose $L_1 \cup L_2$ *is* regular. Then $(L_1 \cup L_2) \setminus L_1$ is regular because of (a). Because of L_1 and L_2 being disjoint this means that $(L_1 \cup L_2) \setminus L_1 = L_2$ is regular. Contradiction.

Exercise 2

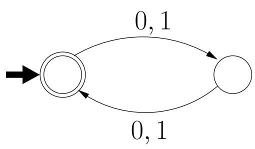
Part (a)

An accepting transducer T:

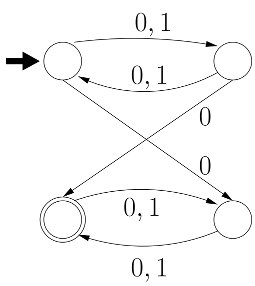


Part (b)

To construct $post_R(L)$, we construct the product of T with the following automaton accepting L:



As a result, we obtain:

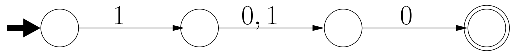


Exercise 3

Part (a)

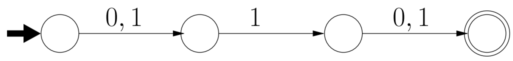
Use the partition algorithm:

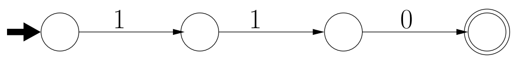
- the final state gets into a partition of its own
- initial state gets into a partition of its own, since it is the only non-final state with only a 1 to other non-final states.
- finally, the second and third columns can be distinguished, because the third is connected to final states and the second is not.
- We obtain:



Part (b)

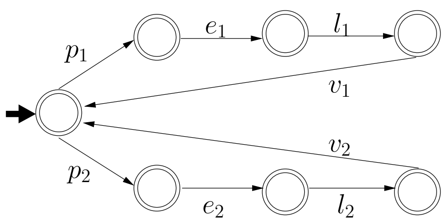
Observe that $\mathcal{L}(A)$ is a bounded language, as well as the one explicitly given. We can thus apply the layer-wise, combined minimization and product construction for bounded languages. The automaton accepting $\{010, 011, 110, 111\}$ and the final result are as follows:





Exercise 4

Part (a)



Part (b)

The second-order variable X contains the positions divisible by 4:

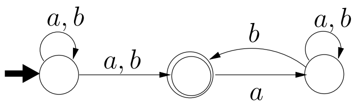
$$\begin{aligned}\exists X. \quad & \forall x.(x \in X \leftrightarrow (zero(x) \vee \exists y \in X. x = y + 4)) \\ \wedge \quad & \forall x_0 \in X. \exists x_1. \exists x_2. \exists x_3. succ(x_0, x_1) \wedge succ(x_1, x_2) \wedge succ(x_2, x_3) \wedge \\ & (\ (Q_{p_1}(x_0) \wedge Q_{e_1}(x_1) \wedge Q_{l_1}(x_2) \wedge Q_{v_1}(x_3)) \quad \vee \\ & \quad (Q_{p_2}(x_0) \wedge Q_{e_2}(x_1) \wedge Q_{l_2}(x_2) \wedge Q_{v_2}(x_3)) \ )\end{aligned}$$

Exercise 5

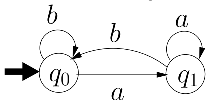
(a) The accepted words are those without consecutive 1's. A regular expression for that language is $(\epsilon + 1)(0 + 01)^*$

(b) $\forall x. \forall y. (Q_1(x) \wedge succ(x, y)) \rightarrow (\neg Q_1(y))$

Exercise 6

(a) 

(b) The following Muller automaton is a solution with acceptance condition $(\{q_0, q_1\})$:



(c) $\forall x. \exists y. \exists z. y > x \wedge z > x \wedge y \in A_a \wedge z \in A_b \wedge Enc(A_a, A_b)$, where *Enc* is the formula in the script guaranteeing a correct encoding of a as $\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and of b as $\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$

Exercise 7

(a) The given universal Co-Büchi automaton accepts the language with infinitely many a's: $(b^*a)^\omega$.

(b) The claim is proven by the following chain of equivalences:

$$\begin{aligned} & L \text{ is an } \omega\text{-regular language} \\ \Leftrightarrow\ & \bar{L} = \Sigma^\omega \setminus L \text{ is an } \omega\text{-regular language, because these are closed under complement.} \\ \Leftrightarrow\ & \text{there exists a non-deterministic Büchi automaton, } (\mathcal{A}, \mathcal{G}) \text{ such that } \mathcal{L}(\mathcal{A}, \mathcal{G}) = \bar{L} \\ \Leftrightarrow\ & \mathcal{L}^{uc}(\mathcal{A}, \mathcal{G}) = L \\ \Leftrightarrow\ & L \text{ is accepted by a universal Co-Büchi automaton.} \end{aligned}$$

In order to understand the penultimate equivalence, observe the following: A word $\alpha \in \Sigma^\omega$ is accepted by the Büchi automaton $(\mathcal{A}, \mathcal{G})$ if and only if there exists a run ρ such that $\mathsf{inf}(\rho) \cap \mathcal{G} \neq \emptyset$. This means that α is *not* accepted by $(\mathcal{A}, \mathcal{G})$ if and only if *for all* runs ρ holds that $\mathsf{inf}(\rho) \cap \mathcal{G} = \emptyset$. And this means that α is accepted by the universal Co-Büchi automaton $(\mathcal{A}, \mathcal{G})$.

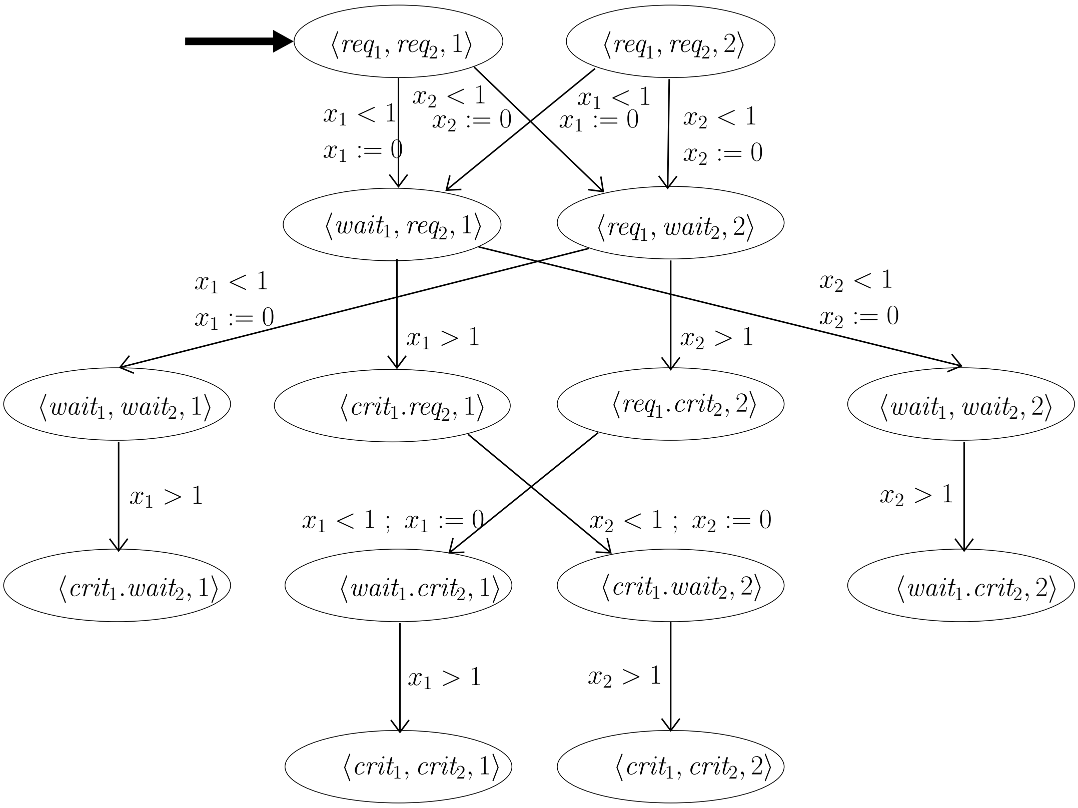
Exercise 8

(a) see next page

(b) $(.5, ask_2)(1.6, enter_2)$

(c)
$$\begin{aligned} (req_1, req_2, v = 1, 0, 0) & \xrightarrow{ask_1} (wait_1, req_2, 1, 0, 0) \\ & \xrightarrow{ask_2} (wait_1, wait_2, 2, 0, 0) \\ & \xrightarrow{1.1} (wait_1, wait_2, 2, 1.1, 1.1) \\ & \xrightarrow{enter_2} (wait_1, crit_2, 2, 1.1, 1.1) \end{aligned}$$

(d) If one automaton moves to the wait location, it must do so within 1 time unit. Now it has to wait for *more* than 1 time unit in order to proceed. If the other automaton wants to move at all, it must do so before that one unit has elapsed. If it does, it effectively prevents the first automaton to move to the critical location because of the constraint on v. If it does not, then it won't be able to do anything at all. In any case, not both automata can be at the critical location at the same time.



Action annotations are left our for clarity.