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

Winter term 2008/2009
February 4, 2009

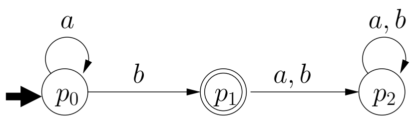
Exam "Automata and Formal Languages"

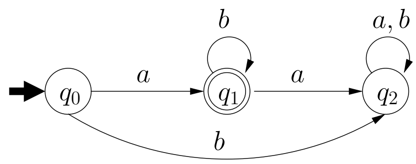
Exercise 1 **(3+4+3+4 points)**

Let $\Sigma = \{a, b\}$ be an alphabet and let $L_1, L_2 \subseteq \Sigma^*$ be languages. We define the *symmetric difference*, $L_1 \ominus L_2$, as follows:

$$L_1 \ominus L_2 = \{w \in \Sigma^* \mid w \text{ is in } L_1 \text{ or in } L_2 \text{ but not in both}\}$$

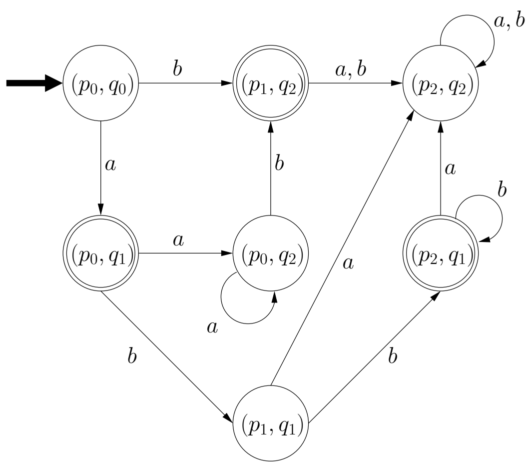
(a) Find two DFAs accepting $\mathcal{L}(a^*b)$ and $\mathcal{L}(ab^*)$, respectively. Make sure your DFAs are complete.





(b) Construct a DFA accepting $\mathcal{L}(a^*b) \ominus \mathcal{L}(ab^*)$.

- The systematic way to solve this exercise is to construct a product automaton with final states $F_1 \times (Q_2 \setminus F_2) \cup (Q_1 \setminus F_1) \times F_2$, where Q_i and F_i are the states and final states of the automata of (a). Since this construction effectively uses a complementation, it is crucial that the input automata are complete.



(c) Find a regular expression r such that $\mathcal{L}(r) = \mathcal{L}(a^*b) \ominus \mathcal{L}(ab^*)$.

- This can be read off the automaton of (b) – or directly found by using the definition of symmetric difference: $a + b + aaa^*b + abbb^*$

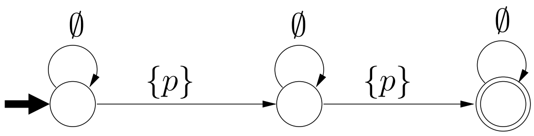
(d) Find an MSO(Σ) formula φ such that $L(\varphi) = \mathcal{L}(a^*b) \ominus \mathcal{L}(ab^*)$. *Hint:* You can solve (d) without having solved (a), (b), and (c).

- It is easiest, if we write formulas ψ_1 and ψ_2 defining $\mathcal{L}(a^*b)$ and $\mathcal{L}(ab^*)$ and then take the xor of these two: $(\psi_1 \wedge \neg\psi_2) \vee (\neg\psi_1 \wedge \psi_2)$, where

$$\begin{array}{rcl} \psi_1 & = & \exists x. \mathit{last}(x) \wedge Q_b(x) \wedge \forall y. y < x \rightarrow Q_a(x) \\ \psi_2 & = & \exists x. \mathit{zero}(x) \wedge Q_a(x) \wedge \forall y. x < y \rightarrow Q_b(x) \end{array}$$

Exercise 2 **(4 points)**

Let $\Sigma = 2^{\{p\}}$ and consider the following Büchi automaton $(\mathcal{A}, \mathcal{G})$.



Give an LTL formula ψ such that $L_\psi = \mathcal{L}(\mathcal{A}, \mathcal{G})$.

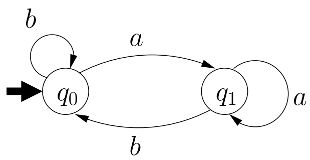
- $\neg p \,\mathsf{U}\, (p \wedge \mathsf{X}(\neg p \,\mathsf{U}\, (p \wedge \mathsf{XG}\neg p)))$

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

Let Σ be an alphabet. For $\alpha \in \Sigma^\omega$ and $\sigma \in \Sigma$ we write $\mathsf{fin}(\sigma, \alpha)$ if σ occurs finitely often in α. Let $L(\Sigma) \subseteq \Sigma^\omega$ be the language $L(\Sigma) = \{\alpha \in \Sigma^\omega \mid \bigvee_{\sigma \in \Sigma} \mathsf{fin}(\sigma, \alpha)\}$

(a) Give a deterministic Muller automaton accepting $L(\{a, b\})$.

- A solution is the Muller automaton $(\mathcal{A}, (\{q_0\}, \{q_1\}))$, where $\mathcal{A}$ is as follows:

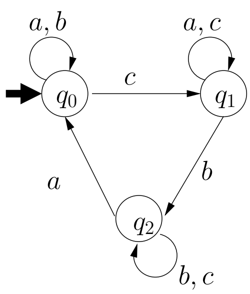


(b) Give an S1S formula $\psi(X_a, X_b, X_c)$ such that $L_\psi = L(\{a, b, c\})_{\{0,1\}}$.

- $\mathit{Enc}(X_a, X_b, X_c) \wedge \left(\begin{array}{ll} & \exists x. \forall y. x < y \rightarrow x \notin X_a \\ \vee & \exists x. \forall y. x < y \rightarrow x \notin X_b \\ \vee & \exists x. \forall y. x < y \rightarrow x \notin X_c \end{array} \right)$

(c) Give a deterministic Muller automaton accepting $L(\{a, b, c\})$ and argue its correctness.

- A solution is the Muller automaton $(\mathcal{A}, (\{q_0\}, \{q_1\}, \{q_2\}))$, where $\mathcal{A}$ is given below. As for why it works, observe that $\mathcal{L}(\{a, b, c\})$ is the set of all words that will *not* see all of a, b, and c infinitely often. The Muller condition ensures that a word is accepted if and only if it eventually only contains at most two of the three letters. Before that point, in any state each of the three letters can be taken.



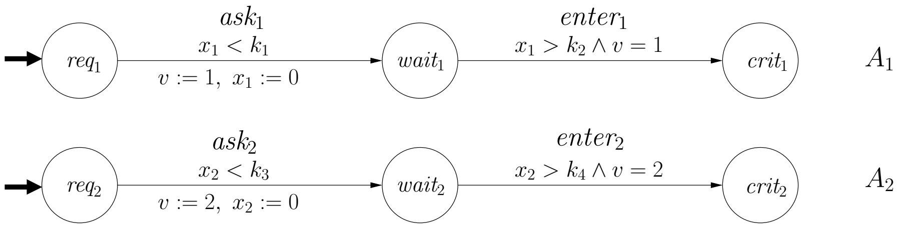
Exercise 4 **(5 points)**

A *strict* Büchi automaton is syntactically like a Büchi automaton $(\mathcal{A}, \mathcal{G})$. It accepts a word $\alpha \in \Sigma^\omega$ if there exists a run ρ of $\mathcal{A}$ on α such that $\mathsf{inf}(\rho) = \mathcal{G}$. Show that the language $\{a^\omega, b^\omega\} \subset \{a, b\}^\omega$ is not accepted by any strict Büchi automaton.

- Suppose there exists a strict Büchi automaton accepting $\{a^\omega, b^\omega\}$. Then there are runs ρ and ρ' accepting a^ω and b^ω, respectively. Let $q \in \mathcal{G}$ be arbitrary (note that $\mathcal{G} \neq \emptyset$). Then ρ visits q infinitely often and there must be a path from q to itself reading only a's and visiting all states in $\mathcal{G}$. Since the same argument holds for q, ρ', and b's, the automaton will also accept words containing both a's and b's. Contradiction.

Exercise 5 **(4+4 points)**

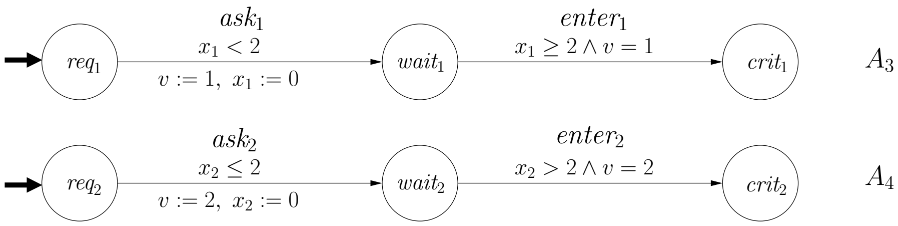
(a) Consider the timed automata A_1, A_2, which operate on the set $\{x_1, x_2\}$ of clocks and the shared integer variable v. A_i has actions $\{ask_i, enter_i\}$ and locations $\{req_i, wait_i, crit_i\}$ for $i = 1, 2$. Guards are given on top of edges, reset and variable updates below edges. Initially, v has value 1. Moreover, $k_i \in \mathbb{R}^+$. Determine one value for each k_i for $i \in \{1, 2, 3, 4\}$ such that a state of the form $(crit_1, crit_2, v =?, x_1 =?, x_2 =?)$ is reachable in $A_1 || A_2$. Give a witnessing sequence of transitions.



- Choose $k_1 = k_2 = k_4 = 1$ and $k_3 = 2$ and the following sequence of transitions.

$$
\begin{aligned}
(req_1, req_2, 1, 0, 0) &\xrightarrow{ask_1} (wait_1, req_2, 1, 0, 0) \\
&\xrightarrow{1.1} (wait_1, req_2, 1, 1.1, 1.1) \\
&\xrightarrow{enter_1} (crit_1, req_2, 1, 1.1, 1.1) \\
&\xrightarrow{ask_2} (crit_1, wait_2, 2, 1.1, 0) \\
&\xrightarrow{1.1} (crit_1, wait_2, 2, 2.2, 1.1) \\
&\xrightarrow{enter_2} (crit_1, crit_2, 2, 2.2, 1.1)
\end{aligned}
$$

(b) Now consider the timed automata A_3 and A_4 below. They are similar to A_1 and A_2 with $k_i = 2$, but have $>$ replaced with $\geq$ in the guard on $enter_1$ and $<$ replaced with $\leq$ in the guard on ask_2. Give a timed trace of $A_3 || A_4$ such that both action $enter_1$ and action $enter_2$ occur.



- $(0, ask_1)(2, enter_1)(2, ask_2)(4.1, enter_2)$