Automata and Formal Languages — Homework 14

Due 10.02.2017

Exercise 14.1
Prove (via a proof) or disprove (via a counterexample) the following LTL equivalence.

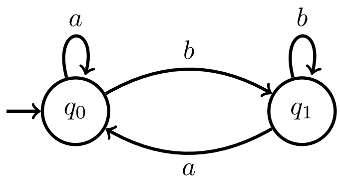
$$(\mathbf{G}p)\ \mathbf{U}\ (\mathbf{G}q) \equiv \mathbf{G}(p\ \mathbf{U}\ q)$$

Exercise 14.2
Let $\Sigma = \{a, b\}$. Give a formula of MSO(Σ) for $(ab + ba)^*$. You may use the following macros from the lecture notes:

$$\begin{aligned} \text{first}(x) &:= \neg\exists y\ y < x \\ y = x + 1 &:= x < y \wedge \neg\exists z(x < z \wedge z < y) \\ y = x + 2 &:= \exists z(z = x + 1 \wedge y = z + 1) \\ X \subseteq Y &:= \forall x(x \in X \rightarrow x \in Y) \\ \text{Odd}(X) &:= \forall x(x \in X \leftrightarrow (\text{first}(x) \vee \exists z(x = z + 2 \wedge z \in X))) \end{aligned}$$

Exercise 14.3
Let A be a DFA recognizing a language $L \subseteq \Sigma^n$ of a fixed length n. What is the language recognized by A seen as a co-Büchi automaton? (Recall that transition function of a DFA is total.)

Exercise 14.4
Describe the language recognized by the following Muller automaton with $F = \{\{q_0, q_1\}, \{q_1\}\}$:



Exercise 14.5

(a) Draw a DFA recognising $(a + bb)^*$.

(b) Given a language $L \subseteq \Sigma^*$ and $w \in \Sigma^*$, give a formal definition of the residual L^w.

(c) Give regular expressions for every residual of the language $(a + bb)^*$.

(d) Prove that the language $L' = \{a^{2^n} : n \in \mathbb{N}_0\}$ has infinitely many residuals. Is L' regular? Justify your answer.

Exercise 14.6

Consider the following program with two parallel processes. The domain of x is $\{0, 1\}$ and the initial value is 0.

Process 1:	Process 2:
while *true* do	**while** *true* **do**
if $x = 0$ **then**	$x \leftarrow 0$
$x \leftarrow 1$	

Model the program by constructing a network of three automata (one for each process and one for the variable x) and then drawing their asynchronous product.

Exercise 14.7

Given a finite automaton $A = (Q, \Sigma, \delta, q_0, F)$ recognizing a language $L \subseteq \Sigma^*$, construct a transducer $T = (Q', \Sigma', \delta', q_0', F')$ recognizing $\{(a_1 \cdots a_n, b_1 \cdots b_n) \in \Sigma^* \times \Sigma^* : a_1 a_2 \cdots a_n \in L$ and $a_1 b_1 a_2 b_2 \cdots a_n b_n \in L\}$.

Exercise 14.8

Construct a LazyDFA recognizing words over the alphabet $\{a, b, c\}$ that contain the pattern *abacab*.

Solution 14.1
The equivalence is false. Let $\sigma = (\{p\}\{q\})^\omega$. We have $\sigma \models \mathbf{G}(p \ \mathbf{U}\ q)$, but $\sigma \not\models (\mathbf{G}p)\ \mathbf{U}\ (\mathbf{G}q)$ since $\mathbf{G}q$ never becomes true.

Solution 14.2

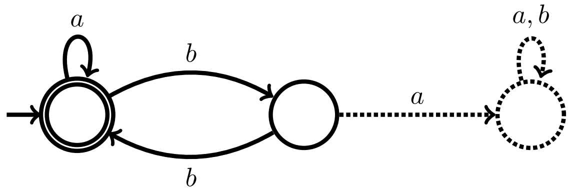
$$\underbrace{\forall x\ \neg\text{first}(x)}_{\varepsilon} \vee \exists X\ \text{Odd}(X) \wedge \underbrace{\forall x \in X\ [\exists y\ (y = x+1)}_{|w| \text{ is even}} \wedge \underbrace{(Q_a(x) \leftrightarrow Q_b(y))}_{w_x w_{x+1} \in \{ab, ba\}}].$$

Solution 14.3
Σ^ω. Every word can be read by A as it is total, and every word is accepted by A since A does not contain any lasso containing an accepting state (otherwise it would accept an infinite language when interpreted as a DFA).

Solution 14.4

$$L = \{w \in \{a,b\}^\omega : w \text{ has infinitely many } b\text{'s}\}.$$

Solution 14.5
(a)



(b) $L^w = \{u \in \Sigma^* : wu \in L\}$.

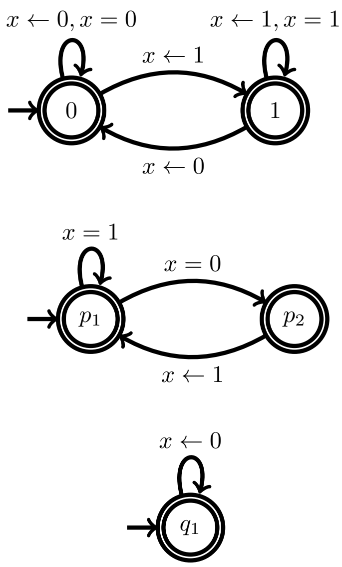
(c) $(a + bb)^*$, $b(a + bb)^*$ and $\emptyset$

(d) Let $i, j \in \mathbb{N}$ be such that $i < j$. We have

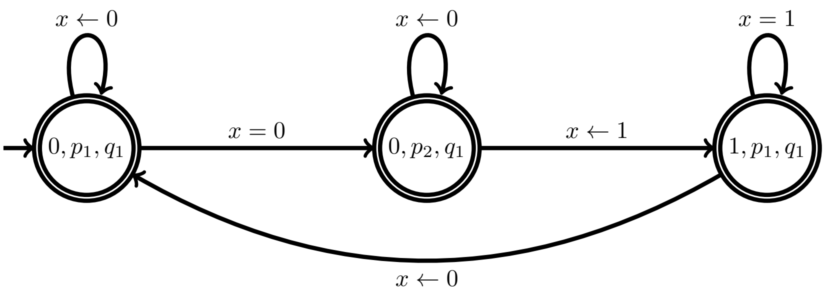
$$\begin{aligned} a^{2^i} a^{2^i} &= a^{2^{i+1}} \in L', \\ a^{2^j} a^{2^i} &= a^{2^i + 2^j} \notin L', \end{aligned} \tag{1}$$

where (1) follows from $2^j < 2^i + 2^j < 2^j + 2^j = 2^{j+1}$. Therefore, L' has infinitely many residuals, and hence L' is not regular, since every regular language has finitely many residuals.

Solution 14.6
Network of automata:



Asynchronous product:



Solution 14.7
T is defined by

$$\begin{aligned} Q' &= Q \times Q, \\ \Sigma' &= \Sigma \times \Sigma, \\ q_0' &= (q_0, q_0), \\ F' &= F \times F, \end{aligned}$$

and for every $p, q \in Q$, $a, b \in \Sigma$,

$$\delta'((p, q), (a, b)) = (\delta(p, a), \delta(\delta(q, a), b)).$$

Solution 14.8