Prof. Bernd Finkbeiner, Ph.D.
Felix Klein, M.Sc.

Problem Set 1
Discussions: April 29th, 2015

Automata, Games, and Verification

Please send a mail to `agv15@react.uni-saarland.de` if you can't make it to the discussion session.

We will use the first tutorial to review some basics about automata on finite words. If you want to brush up some more on this, we recommend the book "*Introduction to Automata Theory, Languages, and Computation*" by Hopcroft, Motwani and Ullman.

An *automaton on finite words* $\mathcal{A}$ is a tuple $(\Sigma, Q, q_0, \Delta, F)$, where Σ is an input alphabet, Q is a nonempty finite set of states, $q_0 \in Q$ is the initial state, $\Delta \subseteq Q \times \Sigma \times Q$ is a set of transitions, and $F \subseteq Q$ are the final states. An automaton $\mathcal{A}$ accepts a finite word $w \in \Sigma^*$ if there is a finite sequence of states $q_0 q_1 \dots q_{|w|}$ such that $(q_i, w_i, q_{i+1}) \in \Delta$ for all $i \leq |w|$ and with $q_{|w|} \in F$. The set of all words accepted by $\mathcal{A}$ is called the *language* of $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$.

1. **Recap: Automata on finite words** (presented by Group 14)

   For each the following languages over $\Sigma = \{a, b\}$ either give an automaton on finite words recognizing the language or argue why no such automaton exists.

   a) $L_1 = \{w \in \Sigma^* \mid w \text{ contains no more than two } b\text{'s}\}$
   b) $L_2 = \{w \in \Sigma^* \mid \text{every } a \text{ in } w \text{ is eventually followed by two } b\text{'s and } w \text{ ends with a } b\}$
   c) $L_3 = \{w \in \Sigma^* \mid w \text{ contains more } a\text{'s than } b\text{'s}\}$
   d) $L_4 = \{w \in \Sigma^* \mid \text{between every pair of } b\text{'s in } w \text{ there are either at least two } a\text{'s or a } b\}$

2. **Recap: Operations on Automata on finite words** (presented by Group 01)

   Show that automata on finite words are closed under union, intersection and complement. I.e., show that for arbitrary automata on finite words $\mathcal{A} = (\Sigma, Q, q_0, \Delta, F)$ and $\mathcal{A}' = (\Sigma', Q', q_0', \Delta', F')$ with $\Sigma = \Sigma'$ there exist automata $\mathcal{A}_\cup$, $\mathcal{A}_\cap$ and $\mathcal{A}^C$ such that

   a) $\mathcal{L}(\mathcal{A}_\cup) = \mathcal{L}(\mathcal{A}) \cup \mathcal{L}(\mathcal{A}')$,
   b) $\mathcal{L}(\mathcal{A}_\cap) = \mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{A}')$ and
   c) $\mathcal{L}(\mathcal{A}^C) = \Sigma^* \setminus \mathcal{L}(\mathcal{A})$.

3. **Model Checking** (presented by Group 15)

   Consider the following program TURN':

$$\text{local } t_0, t_1 \text{: boolean where initially } t_0 = 0, t_1 = 0$$

$$P_0 :: \left[\begin{array}{l} \text{loop forever do} \\ \quad \left[\begin{array}{ll} 00: & t_1 := 1; \\ 01: & \text{await } t_0 = 1; \\ 10: & t_0 := 0; \\ 11: & \text{critical}; \end{array}\right] \end{array}\right] \;||\; P_1 :: \left[\begin{array}{l} \text{loop forever do} \\ \quad \left[\begin{array}{ll} 00: & \text{await } t_1 = 1; \\ 01: & \text{critical}; \\ 10: & t_1 := 0; \\ 11: & t_0 := 1; \end{array}\right] \end{array}\right]$$

   a) Present the behaviour of the program TURN' as an automaton using bitvectors of a fixed length as the alphabet. Explain which information is represented by each bit of the bitvectors.
   b) Does TURN' satisfy the mutual exclusion property? Argue using your solution of a).

4. **Synthesis** (presented by Group 09)

   Consider the following set of processes, where the Arbiter process is left open.

$$\text{local } t_0, t_1 \text{: boolean where initially } t_0 = t_1 = 0$$

$$P_0 :: \left[\begin{array}{l} \text{loop forever do} \\ \quad \left[\begin{array}{l} 00 : \text{await } t_0 = t_1; \\ 01 : \text{critical;} \\ 10 : \text{await } t_0 = 1; \\ 11 : \text{critical;} \end{array} \right] \end{array} \right] \; || \; P_1 :: \left[\begin{array}{l} \text{loop forever do} \\ \quad \left[\begin{array}{l} 00 : \text{await } t_0 \neq t_1; \\ 01 : \text{critical;} \\ 10 : \text{await } t_1 \neq 1; \\ 11 : \text{critical;} \end{array} \right] \end{array} \right] \; || \; \text{Arbiter} :: \textbf{?}$$

   a) Construct a game arena where the system player fixes the output of t_0 and t_1 and the environment player makes moves in P_0 and P_1.

   b) Is there an arbiter implementation which satisfies the mutual exclusion property and which ensures that each process P_j for $j \in \{0, 1\}$ can enter all its critical sections infinitely often? In case of a positive answer, give the corresponding system player strategy of your constructed game.

5. **Challenge: Countable**

   A set S is *countable* if there is an onto function $\tau \colon \mathbb{N} \to S$ from the natural numbers to that set, i.e, $\tau(\omega) \supseteq S$. Let X be the set of all infinite sequences $v_0 v_1 v_2 \ldots$ with $v_n \in \{a, b\}$ for all $n \in \mathbb{N}$. Prove or disprove that X is countable.