Solution

Automata and Formal Languages – Homework 4

Due 19.11.2009.

Exercise 4.1

Show that for any natural number $n \geq 2$ there exist *minimal* DFAs $\mathcal{A}_1$ and $\mathcal{A}_2$, both having at most $n+1$ states, such that the *minimal* DFA accepting $\mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$ has at least n^2 states.

Solution: Let $\mathcal{A}_n$ denote the minimal DFA with $\mathcal{L}(\mathcal{A}_n) = \mathcal{L}((a^n)^*)$ over the alphabet $\Sigma = \{a\}$. Then $\mathcal{A}_n$ is basically a modulo-n counter and has exactly n states $\{0, 1, \ldots, n-1\}$ with $i \xrightarrow{a} (i+1) \pmod n$.

We claim that the minimal DFA for $(a^n)^* + (a^{n+1})^*$ has $n(n+1)$ states (for $n \geq 2$). For this to show, it suffices to prove for every pair $(i,j),(k,l) \in \{0,\ldots,n-1\} \times \{0,\ldots,n\}$ of states of the product automaton that $\mathcal{L}((i,j) \neq \mathcal{L}((k,l))$ for $(i,j) \neq (k,l)$.

Assume $0 \leq i < k < n$ and set $d := n-k$. If $j + d \equiv 0 \pmod n + 1$, then increase d by n, i.e., $d := 2n - k$. We then have that $i + d \not\equiv 0 \pmod n$ and $k + d \equiv 0 \pmod n$. Further, we also have $j + d \not\equiv 0 \pmod n + 1$ by choice of d. Hence, $(i+d, j+d)$ is not an accepting state, while $(k+d, l+d)$ is, i.e., the two state cannot be equivalent.

Exercise 4.2

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be an NFA.

We introduce a "game" on $\mathcal{A}$: We have two players, called "refuter" (short: $\bot$) and "prover" (short: $\top$). A play $\pi = \{s_0,t_0\}\{s_1,t_1\}\ldots$ of the two players on $\mathcal{A}$ is a sequence of (unordered) pairs of states. The initial pair $\{s_0,t_0\}$ can be chosen arbitrarily, but the choice of a successor pair $\{s_{i+1},t_{i+1}\}$ is limited by the current pair $\{s_i,t_i\}$ as follows:

- If $s_i \in F \Leftrightarrow t_i \notin F$, then the play terminates immediately, and player $\bot$ is declared the winner of the play.
- If the pair $\{s_i,t_i\}$ has been visited before in the play, i.e., there is some $j < i$ s.t. $\{s_i,t_i\} = \{s_j,t_j\}$, then the play terminates and player $\top$ wins; otherwise:
- Two tokens are put on the states $\{s_i,t_i\}$. If $s_i = t_i$, then both tokens lie on the same state.

  Player $\bot$ then moves exactly one of the two tokens along an outgoing transition of the state the chosen token is located on.

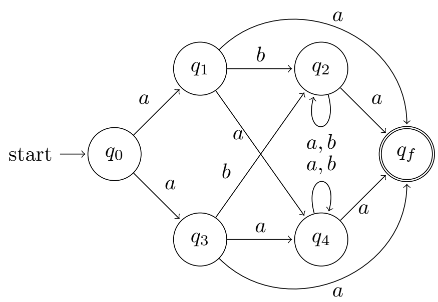
  If both states do not have outgoing transitions, i.e., if player $\bot$ cannot move, player $\top$ wins; otherwise:
- Let a be the label of the transition along which player $\bot$ has moved.

  Player $\top$ now has to try to match this move by moving the other token, i.e., the one that hasn't been moved in this round yet, along an outgoing transition also labeled by a.

  If player $\top$ cannot move in such a way, he immediately loses; otherwise:
- The new pair $\{s_{i+1},t_{i+1}\}$ is determined by the states on which the two tokens are located.

A player wins a pair $\{s_0,t_0\}$ of states if he can choose his moves in such a way that he wins any resulting play starting in $\{s_0,t_0\}$.

(a) Consider the NFA of Ex3.4:



Determine which player wins the pair $\{q_1, q_2\}$, resp. the pair $\{q_4, q_f\}$, if there is a winner.

(b) Show for arbitrary NFAs that if $s_0 \sim t_0$, then player wins the pair $\{s_0, t_0\}$, otherwise player $\bot$ wins.

Hint: For the case that $s_0 \not\sim t_0$, consider the sequence of partitions $\sim_0, \sim_1, \ldots, \sim_l$ constructed by the algorithm "Bisim" (with $\sim = \sim_l$).

Solution:

(a) -

(b)
- Assume $s_0 \sim t_0$. We then have $s_0 \in F \Leftrightarrow t_0 \in F$, and as we are at the start of the play, we haven't visited any other pair previously, so player $\bot$ has to move one token.

  If player $\bot$ cannot move, i.e., if both s_0 and t_0 have no outgoing transitions, then player $\top$ wins by definition, and we are done.

  Hence, assume player $\bot$ moves one token along an transition labeled by a. As $s_0 \sim t_0$, player $\top$ finds also an outgoing a-transition for the other token such that the resulting pair of states is bisimilar again.

  Player $\top$ can react on the moves of player $\bot$ until either $\{s_i, t_i\}$ is a dead end (both states have no outgoing transitions) or the players reach a previously visited pair of states. One of the two cases has eventually to hold as there are only finitely many pairs of states. In both cases player $\top$ wins the play, no matter how player $\bot$ chooses his moves. Hence, player $\top$ wins the pair $\{s_0, t_0\}$.

- For $s \not\sim t$ let $r(\{s,t\})$ be the least number such that $s \not\sim_l t$ for all $l \geq r(\{s,t\})$.

  We show by induction on $r := r(\{s,t\})$ that player $\bot$ wins the pair $\{s,t\}$. (As $\sim_l = \sim$, r is defined.)

  More precisely, we show that player $\bot$ can choose his moves in such a way that $r(\cdot)$ decreases along the play:

  - $r = 0$:

    We have $s \not\sim_0 t$, i.e., $s \in F \Leftrightarrow t \notin F$. Hence, player $\bot$ trivially wins any play starting from $\{s,t\}$, as there is only one play which has length 0.

  - $r \to r+1$:

    Assume $s \sim_r t$ and $s \not\sim_{r+1} t$. Then there is some letter a and some class K of $\sim_r$ such that either s has an a-transition leading to some state of K while no a-transition of t leads to a state of K; or the symmetrical case with s and t exchanged holds. We only consider the first case.

    Hence, if $\bot$ moves the token located on s along any a-transition ending in K, no matter how player $\top$ chooses his a-transition (if there is one at all), the resulting new pair $\{s', t'\}$ of states will be non-bisimilar again, in particular, $s' \not\sim_r t'$ as $s' \in K$ and $t' \notin K$, i.e., $r(\{s', t'\}) < r(\{s,t\})$.