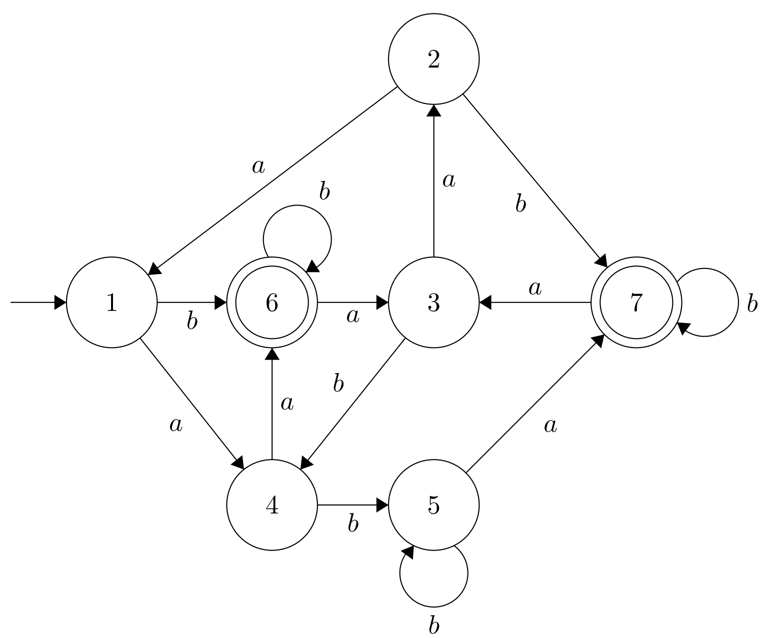
Regularitet og Automater: Assignment 4

Rasmus Dalsgaard (201605295)

May 2017

Introduction

In this assigment we're tasked to minimize the following finite automaton by hand. This procedure is done by eliminating empty language states and using marking tables.



Step 1

The very first step in reducing a finite automaton is a series of attempts to manipulate the states in order to single out and exterminate redundancies.
Given an automaton M, it is pre-defined as $(Q, \Sigma, q_0, A, \delta)$. M accepts $L \subseteq \Sigma *$.
For a state $q \in Q$ of M, let L_q denote the set of strings that causes M to reside in the bowels of q:

$$Ł_q = \{x \in \Sigma^* | \delta^*(q_0, x) = q$$

Now, to remove the aforementioned redundant states, we start off by eliminating every q for which $L_q = \emptyset$, including the transitions of these.

We do this recursively with

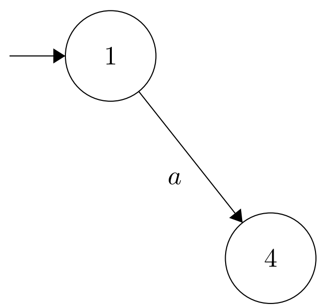
$$\forall q \in R, a \in \Sigma : \delta(q, a) \in R$$

R, of course, is the set of reachable states from q_0; the initial state of the FA. This procedure will populate R fully.

Applying this function by hand to $M = (Q, \Sigma, q_0, A, \delta)$ where $Q = \{1, 2, 3, 4, 5, 6, 7\}$, $\Sigma = \{a, b\}$, $q_0 = \{1\}$ and $A = \{6, 7\}$, it's all about plugging in those numbers and keeping track of R. Let's go ahead with $\sigma_0 \in \Sigma$

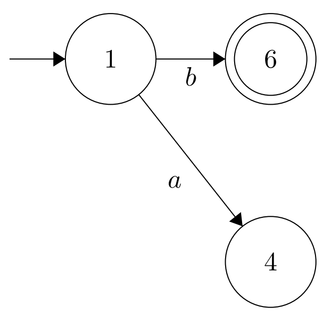
$$q_0 \in R \rightarrow \delta(q_0, a) = \{4\} \in R$$

This means that $R = \{1\}$ gets expanded with the newly discovered state, 4, such that $R = \{1, 4\}$.
A visual representation of this subcomputation:



Let's do that again with the remaining alphabet symbol $\sigma_1 = b$

$$q_0 \in R \rightarrow \delta(q_0, b) = \{6\} \in R, A$$

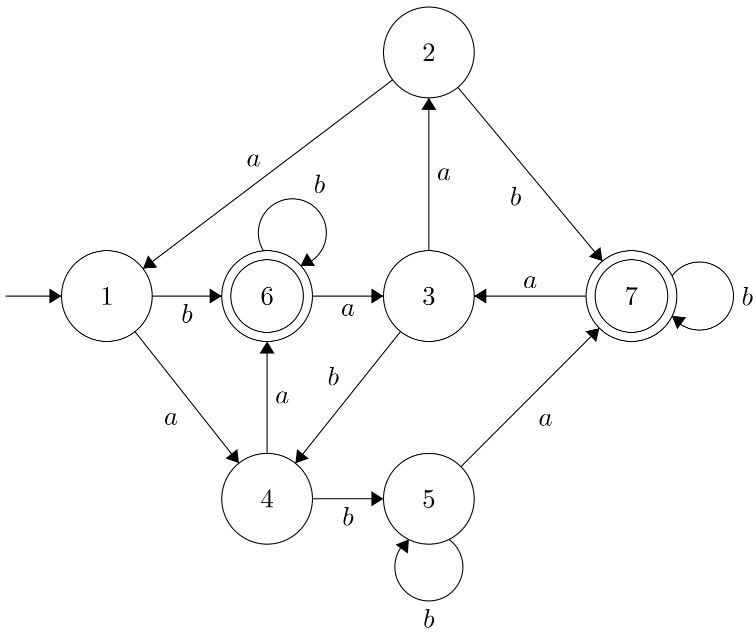


This procedure is kept running for as long as R gets appended new states per iteration. Once R remains static, the procedure is terminated and we end up with the full list of reachable states

$$R = \{1, 4, 6, 5, 3, 7, 2\}$$

$$A = \{6, 7\}$$

If the final result of removing $L_q = \emptyset$ seems fairly familiar, it's because this FA is an exact replica of the one given; not a single state could be excluded by this procedure. A fully-explored FA drawn from M:



Step 2

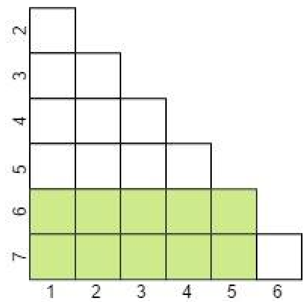
The second step of minimzation of a finite automaton is the nifty trick of drawing tables. Who would've guessed. [1] Listing all unordered pairs of distinct states (p, q), we make a sequence of picaresque pass-throughs:

On the first pass, mark each pair such that

$$\forall(p, q) : (p \in A \wedge q \notin A) \vee (p \notin A \wedge q \in A)$$

Note that this computation is based on general, unordered pairs; not on δ. This is possible because we earlier on excluded all unreachable nullstates.
Applying this onto the empty grid, marking any pairs that are both an accept state and not an accept state:

[1] [Martin], p. 75: "Algorithm 2.40"



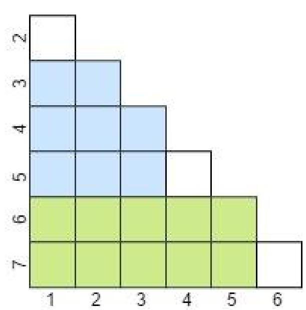
Now that the first iteration is done, we use a different technique to progress: examination of unmarked pairs, (r, s). If there is a symbol $\sigma \in \Sigma$ such that $\delta(r, \sigma) = p$ and $\delta(s, \sigma) = q$, and the pair (p, q) is already marked, then mark (r, s). More fomally, this can be expressed as

$$\exists a \in \Sigma : (\delta(p, a), \delta(q, a)) \in \delta \ \rightarrow (p, q) \in \delta$$

First passthrough of this new approach will rattle the states even harder and end up marking even more pairs. Let's try it out! Let's pick the unmarked pair $(1, 5)$ from the table; these are now our (r, s).

$$(\delta(1, a), \delta(5, a)) \rightarrow (4, 7)$$

Inspecting our table, $(4, 7)$ is a marked pair; and if (p, q) was already marked then we mark (r, s), therefore we mark $(1, 5)$. Now we can skip $b \in \Sigma$ because the pair is already marked. Following this rule of thumb all through the table in a single iteration gives us the following table



The most recent passthrough marked up new pairs, and so we must traverse the table again to check if any new pairs can be marked. It turns out that $(1, 2)$ is the markable pair left

$$(\delta(1, a), \delta(2, a)) \rightarrow (4, 1)$$

For completeness sake let's rule out the other two pairs:

$$(\delta(4, a), \delta(5, a)) \rightarrow (6, 7)$$

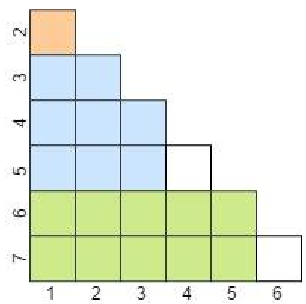
$$(\delta(4, b), \delta(5, b)) \rightarrow (5, 5)$$

The pair $(6, 7)$ is unmarked and the pair $(5, 5)$ is invalid, and so we do nothing.

$$(\delta(6, a), \delta(7, a)) \rightarrow (3, 3)$$

$$(\delta(6, b), \delta(7, b)) \rightarrow (6, 7$$

The pair $(3, 3)$ is invalid and the pair $(6, 7)$ is unmarked, and so we do nothing.



The unmarked spots reveals the redundancy of pair (4,5) and pair (6,7), and these may be merged into single states, following the inlaid structures

- State 4
  - one inbound edge of a from $q \notin A$
  - one inbound edge of b from $q \notin A$
  - one outbound edge of a to $q \in A$
  - one outbound edge of b to $p \notin A$
- State 5
  - one inbound edge of b from $q \notin A$
  - one outbound edge of a to $q \notin A$
  - one outbound edge of b to $p \notin A$

into a single state, $\{4, 5\}$ maintaining all the properties of both former states:

- (4,5)
  - one inbound edge of a from $q \notin A$
  - one inbound edge of b from $q \notin A$
  - one outbound edge of a to $q \in A$
  - one outbound edge of b to $p \notin A$

Minimized FA

Drawing the finite automaton from scratch with the merged $(4, 5)$ state and $(6, 7)$ state, gets us something like this: