Lecture L&M 3

Pumping Lemma, Regular Languages, DFA Minimisation

3.1 Pumping lemma and non-regular languages

For each of the following languages L, use the pumping lemma to prove that it is non-regular:

A. $L = \{w \in \{a,b\}^* \mid \#_a(w) = \#_b(w)\}$, i.e. the language of all words over $\{a,b\}$ with the same number of a and b.

B. Properly matched parentheses: L is the language over $\{(,)\}$ in which each (is paired with a), and each) is preceded by its matching (.

   Note: Words in this language are for instance λ, $()$, $(())$ and $()(())$, but neither $)($ nor $((())()$.

C. The language of palindromes over $\{a,b\}$, i.e. $L = \{w \in \{a,b\}^* \mid w = w^R\}$.

D. The language of "double words" over $\{a,b\}$, i.e. $L = \{ww \mid w \in \{a,b\}^*\}$.

E. *(extra)* $L = \{a^n b^m \mid n < m\}$.

F. *(extra)* $L = \{a^i b^j c^{2j} \mid i \geq 0, j \geq 0\}$.

Hint: The idea of the proofs for exercises A and B is rather similar. Do only one of them during the tutorial and save the other one for the exam preparation. The same applies to exercises C and D.

Now prove that the following language L is not regular. You do not *need* to use the pumping lemma.

G. The language of non-palindromes over $\{a,b\}$, i.e. $L = \{w \in \{a,b\}^* \mid w \neq w^R\}$.

3.2 The class of regular languages

Let L be a regular language over the alphabet $\Sigma = \{a,b,c\}$. Show that the following languages L' are regular as well:

H. The words from L that end with aa, i.e. $L' = \{w \in L \mid w = vaa \text{ for some } v \in \Sigma^*\}$.

I. The words *not* in L that contain no a, i.e. $L' = \{w \in \Sigma^* \mid w \notin L \wedge \#_a(w) = 0\}$.

Now let L be a regular language over an arbitrary alphabet Σ. Show that the following languages are also regular:

J. The prefixes of words from L, i.e. the language $P = \{u \mid \exists\, v \in \Sigma^*\colon uv \in L\}$.

K. *(extra)* The reversals of words from L, i.e. the language $L^R = \{w^R \mid w \in L\}$.

L. *(extra)* The words which have a suffix in L, i.e. the language $E = \{uv \mid u \in \Sigma^* \wedge v \in L\}$.

Give examples of languages L_1 and L_2 over $\{a, b\}$ that satisfy the following descriptions:

M. L_1 is regular, L_2 is non-regular, and $L_1 \cup L_2$ is regular.

N. L_1 is regular, L_2 is non-regular, and $L_1 \cup L_2$ is non-regular.

O. L_1 is regular, L_2 is non-regular, and $L_1 \cap L_2$ is regular.

P. L_1 is non-regular, L_2 is non-regular, and $L_1 \cup L_2$ is regular.

Q. *(extra)* L_1 is non-regular and L_1^* is regular.

3.3 DFA minimisation

R. Assume that q_i and q_j are equivalent states in some DFA M. Prove that, if $\hat{\delta}(q_i, u) = q_m$ and $\hat{\delta}(q_j, u) = q_n$ for some $u \in \Sigma^*$, then q_m and q_n are also equivalent.

S. For each of the three DFA below:

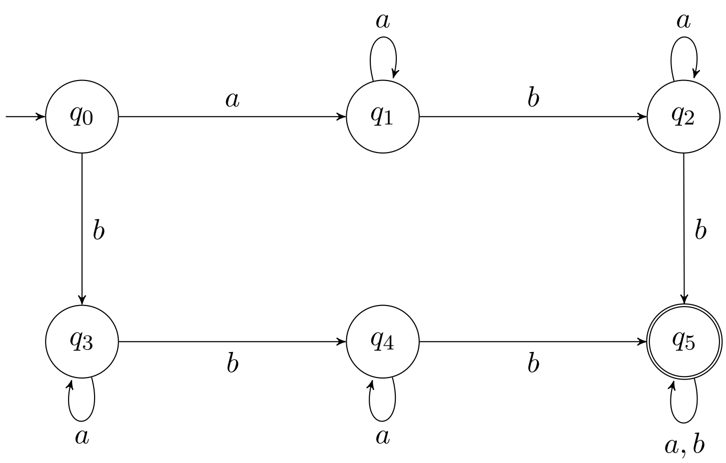
   (a) Trace the actions of Algorithm 5.7.2 in the book (or see below step-by-step, showing the values of $D[i, j]$ and the sets of $S[i, j]$.

   Note: Recall only $i < j$ needs to be considered. Do this for one of the DFA of your choice, and then only for the other DFA if you feel that you need more practice.

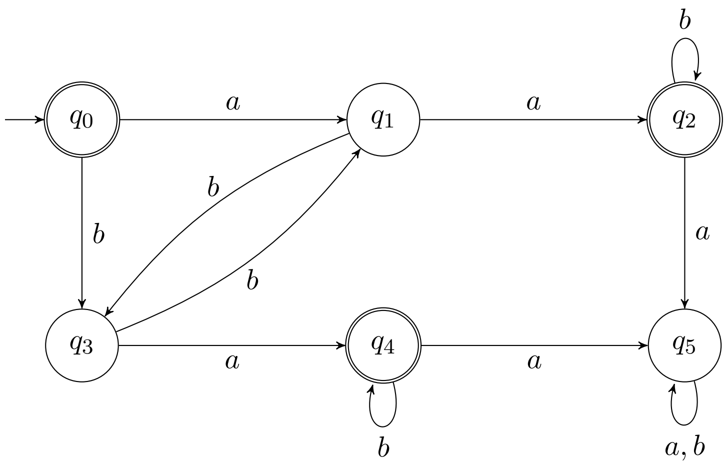
   (b) Give the equivalence classes of states.

   (c) Draw a state diagram for the minimised DFA that accepts the same language.

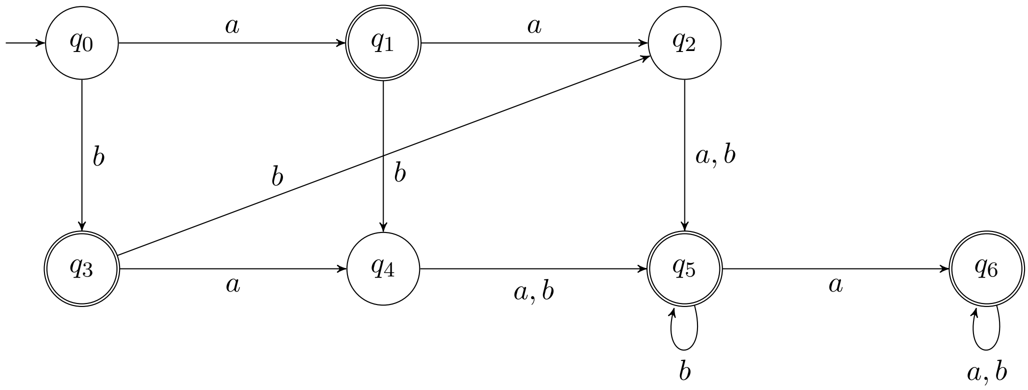
   1.



2.



3.



Algorithm 5.7.2
Determination of Equivalent States of DFA

input: DFA M = (Q, Σ, δ, q_0, F)

1. (Initialization)
   for every pair of states q_i and q_j, $i < j$, **do**
   - 1.1. $D[i, j] := 0$
   - 1.2. $S[i, j] := \emptyset$

   end for
2. **for** every pair $i, j, i < j$, **if** one of q_i or q_j is an accepting state and the other is not an accepting state, **then** set $D[i, j] := 1$
3. **for** every pair $i, j, i < j$, with $D[i, j] = 0$, **do**
   - 3.1. **if** there exists an $a \in \Sigma$ such that $\delta(q_i, a) = q_m$, $\delta(q_j, a) = q_n$ and $D[m, n] = 1$ or $D[n, m] = 1$, **then** $DIST(i, j)$
   - 3.2. **else** for each $a \in \Sigma$, do: Let $\delta(q_i, a) = q_m$ and $\delta(q_j, a) = q_n$
     if $m < n$ and $[i, j] \neq [m, n]$, **then** add $[i, j]$ to $S[m, n]$
     else if $m > n$ **and** $[i, j] \neq [n, m]$, **then** add $[i, j]$ to $S[n, m]$

   end for

$DIST(i, j)$;
begin
  $D[i, j] := 1$
  for all $[m, n] \in S[i, j]$, $DIST(m, n)$
end