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.

Note: These exercises highlight the proof scheme of proving non-regularity by means of the Pumping Lemma. The core concept is to identify the regular part of the expression that can be "pumped-up," bloating some (extract of a) word taken from the language until we can prove it no longer follows the scheme.

Let L be the set of words over $\{a,b\}$ with the same number of a and b.
To prove: L is non-regular.

- Let $k > 0$ be given arbitrarily.
- Choose $z = a^k b^k$. Then $z \in L$ and $|z| = 2k \geq k$.
- Let u, v, w be given arbitrarily, such that $z = uvw$ and $|uv| \leq k$ and $|v| > 0$.
  - Then $u = a^p$ and $v = a^q$ and $w = a^{k-p-q} b^k$, for certain p, q with $p + q \leq k$ and $q > 0$.
- Choose $i = 2$.
  - $uv^2w = a^{k+q}b^k$. Since $q > 0$, there must be more a than b in the word uv^2w, thus $uv^iw \notin L$.
- It then follows from the pumping lemma that L cannot be regular.

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 ((())().

Let L be the set of words over $\{(,)\}$ where the parentheses form pairs.
To prove: L is non-regular.

- Let $k > 0$ be given arbitrarily.
- Choose $z = (^k)^k$, then $z \in L$ and $|z| = 2k \geq k$.
- Let u, v, w be given arbitrarily, such that $z = uvw$ and $|uv| \leq k$ and $|v| > 0$
  - Then $u = (^p$ and $v = (^q$ and $w = (^{k-p-q})^k$, for certain p, q with $p + q \leq k$ and $q > 0$.
- Choose $i = 2$.
  - $uv^2w = (^{k+q})^k$. Since $q > 0$, there must be more (than) in the word uv^2w, thus $uv^iw \notin L$.
- It then follows from the pumping lemma that L cannot be regular.

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

Let L be the set of palindromes over $\{a, b\}$.
To prove: L is non-regular.

- Let $k > 0$ be given arbitrarily.
- Choose $z = a^k b a^k$. This is a palindrome, so $z \in L$ and moreover $|z| = 2k + 1 \geq k$.
- Let u, v, w be given arbitrarily, such that $z = uvw$ and $|uv| \leq k$ and $|v| > 0$
  - Then $u = a^p$ and $v = a^q$ and $w = a^{k-p-q} b a^k$, for certain p, q with $p + q \leq k$ and $q > 0$.
- Choose $i = 2$.
  - $uv^2w = a^{k+q} b a^k$. Since $q > 0$, this cannot be a palindrome, thus $uv^i w \notin L$.
- It then follows from the pumping lemma that L cannot be regular.

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

Let $L := \{xx \mid x \in \{a, b\}^*\}$.
[use x here to avoid confusion with w from the pumping lemma]
To prove: L is non-regular.

- Let $k > 0$ be given arbitrarily.
- Choose $z = a^k b a^k b$, then $z \in L$ and moreover $|z| = 2k + 2 \geq k$.
- Let u, v, w be given arbitrarily, such that $z = uvw$ and $|uv| \leq k$ and $|v| > 0$
  - Then $u = a^p$ and $v = a^q$ and $w = a^{k-p-q} b a^k b$, for certain p, q with $p + q \leq k$ and $q > 0$.
- Choose $i = 2$.
  - $uv^2w = a^{k+q} b a^k b$. Since $q > 0$, this can never be of the form xx:
    Either uv^2w is odd, or the right side contains two b.
- It then follows from the pumping lemma that L cannot be regular.

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

Let $L := \{a^n b^m \mid n < m\}$.
To prove: L is non-regular.

- Let $k > 0$ be given arbitrarily.
- Choose $z = a^k b^{k+1}$, then $z \in L$.
- Let u, v, w be given arbitrarily, such that $z = uvw$ and $|uv| \leq k$ and $|v| > 0$

  - Then $u = a^p$ and $v = a^q$ and $w = a^{k-p-q}b^{k+1}$, for certain p, q with $p + q \leq k$ and $q > 0$.
- Choose $i = 42$.
  - $uv^{42}w = a^p a^{42q} a^{k-p-q} b^{k+1} = a^{k+41q} b^{k+1}$. Since $q > 0$, there cannot be more b than a, thus $uv^i w \notin L$.
- It then follows from the pumping lemma that L cannot be regular.

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

Let $L = \{a^i b^j c^{2j} \mid i \geq 0, j \geq 0\}$.
To prove: L is non-regular.

- Let $k > 0$ be given arbitrarily.
- Choose $z = b^k c^{2k}$, then $z \in L$ where $i = 0$.
- Let u, v, w be given arbitrarily, such that $z = uvw$ and $|uv| \leq k$ and $|v| > 0$
  - Then $u = b^p$ and $v = b^q$ and $w = b^{k-p-q}c^{2k}$, for certain p, q with $p + q \leq k$ and $q > 0$.
- Choose $i = k + 1$.
  - $uv^{k+1}w = b^p b^{kq+q} b^{k-p-q} c^{2k} = b^{k(q+1)} c^{2k}$. Since $q > 0$, there cannot be twice as many c as there are b, thus $uv^i w \notin L$.
- It then follows from the pumping lemma that L cannot be regular.

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. $L = \{w \in \{a, b\}^* \mid w \neq w^R\}$.

Let L be the set of non-palindromes.
To prove: L is non-regular.
Proof by contradiction:
Suppose L is regular, then there exists a DFA M with $\mathcal{L}(M) = L$. DFAs can be complemented, so there exists a DFA M' with $\mathcal{L}(M') = \overline{L}$. This means $\overline{L}$ is regular. Since that is precisely the set of palindromes, this stands in contradiction to Exercise C, where it was shown that the the set of palindromes over $\{a, b\}$ is not a regular language. Conclusion: L' is non-regular.

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. $L' = \{w \in L \mid w = vaa \text{ for some } v \in \Sigma^*\}$.

Let $L' = \{w \in L \mid w = vaa \text{ for some } v \in \Sigma^*\}$ for regular language L.
To prove: L' is regular.
Since the condition "ends with aa" is regular and the class of regular languages is closed under intersection, it follows that L' is regular.
More in detail, let $\hat{L}$ be the language defined by the regular expression $(a \cup b \cup c)^* aa$. By definition $\hat{L}$ is regular, thus $L' = L \cap \hat{L}$ is proof that L' is regular.

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

Let $L' = \{w \in \Sigma^* \mid w \notin L \wedge \#_a(w) = 0\}$ for regular language L.
To prove: L' is regular.
Since the condition "contains no a" is regular and the class of regular languages is closed under intersection and complement, it follows that L' is regular.
More in detail, let $\hat{L}$ be the language defined by the regular expression $(b \cup c)^*$. By definition $\hat{L}$ is regular, thus $L' = \overline{L} \cap \hat{L}$ is proof that L' is regular.

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 $P = \{u \mid \exists\, v \in \Sigma^* \colon uv \in L\}$.

Let $P = \{u \mid uv \in L\}$ for L regular language.
To prove: P is regular.
If L is regular, then there exists a DFA M with $\mathcal{L}(M) = L$. Let $M = (Q, \Sigma, q_0, \delta, F)$. Let $Q' \subseteq Q$ be the states that have a path to F—the states in $Q - Q'$ are *traps* or *sinks*. Then the DFA $M' = (Q, \Sigma, q_0, \delta, F \cup Q')$ accepts all prefixes of L, i.e. $\mathcal{L}(M') = P$, which proves that P is regular.

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

Let $L^R = \{w^R \mid w \in L\}$ for L regular language.
To prove: L^R is regular.
If L is regular, then there exists a Regular Expression E with $\mathcal{L}(E) = L$. Define E^R as follows:

$$
\begin{aligned}
\emptyset^R &= \emptyset \\
\lambda^R &= \lambda \\
a^R &= a \\
(E_1\, E_2)^R &= E_2^R\, E_1^R \\
(E_1 \cup E_2)^R &= E_1^R \cup E_2^R \\
(E^*)^R &= (E^R)^*
\end{aligned}
$$

It can be proven with induction on the structure of E that $\mathcal{L}(E^R) = (\mathcal{L}(\mathcal{E}))^R$.

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

Let $E = \{uv \mid v \in L\}$ for L regular language.
To prove: E is regular.
Assume $M = (Q, \Sigma, q_0, \delta, F)$ is some NFA with $\mathcal{L}(M) = L$. Define a new initial state q that accepts an arbitrary prefix, and a λ-step to q_0. Define $M' = (Q \cup q, \Sigma, q, \delta', F)$, where for each $a \in \Sigma$, $q \to^a q$. Also, $q \to^\lambda q_0$. Then M' accepts E.
Note: There is a simpler proof: $E = \Sigma^* L$, so E is the concatenation of two regular languages and thus a regular language itself.

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.

Let $L_1 = \{a, b\}^*$, $L_2 = \{a^n b^n \mid n \geq 0\}$. Then $L_1 \cup L_2 = \{a, b\}^*$ which is trivially regular.

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

Let $L_1 = \varnothing$, $L_2 = \{a^n b^n \mid n \geq 0\}$. Then $L_1 \cup L_2 = \{a^n b^n \mid n \geq 0\}$, which has been proven to be non-regular in the lecture.

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

Let $L_1 = \varnothing$, $L_2 = \{a^n b^n \mid n \geq 0\}$. Then $L_1 \cap L_2 = \varnothing$ which is trivially regular.

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

Let $L_1 = \{a^n b^n \mid n \geq 0\}$, $L_2 = \{a^m b^n \mid m \neq n\}$. Then $L_1 \cup L_2 = \{a, b\}^*$ which is trivially regular.

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

Let $L_1 = \{a^{m^2} \mid m \geq 0\}$. Then $L_1^* = \{a\}^*$, because every number is the sum of a number of squares.

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.

To prove equivalence we must show that q_m and q_n accept the same words, i.e. $\hat{\delta}(q_m, w) \in F$ iff $\hat{\delta}(q_n, w) \in F$ for every $w \in \Sigma^*$. Since q_i and q_j are assumed equivalent, then for any $w \in \Sigma^*$ it must hold that $\hat{\delta}(q_i, uw) \in F$ iff $\hat{\delta}(q_j, uw) \in F$. But $\hat{\delta}(q_i, uw) = \hat{\delta}(q_m, w)$ and $\hat{\delta}(q_j, uw) = \hat{\delta}(q_n, w)$. Therefore q_m and q_n are equivalent, since w was picked arbitrarily.

S. For each of the three DFA below:

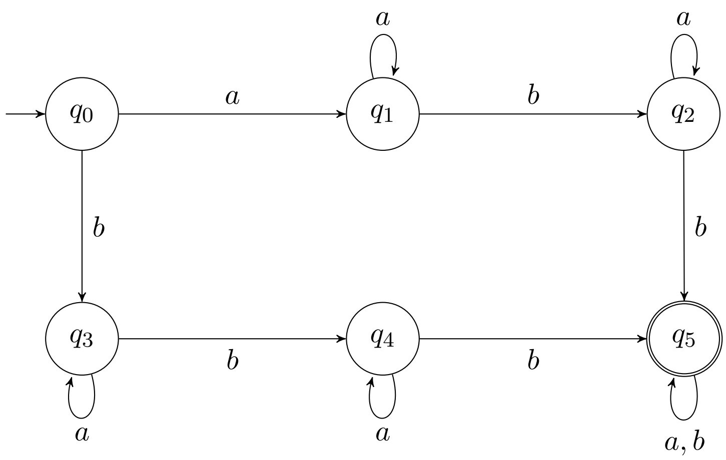
(a) Trace the actions of Algorithm 5.7.2 in the book (or see below step-by-step, showing the values of $D[i, j]$ and the sets of $S[i, j]$.

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.

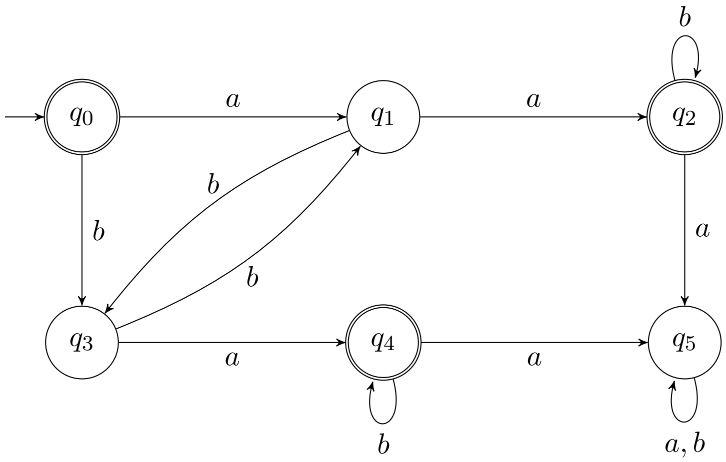
(b) Give the equivalence classes of states.

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

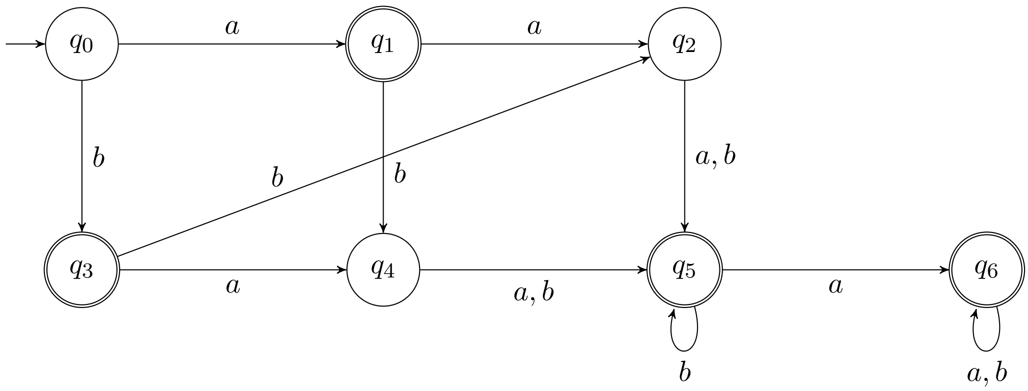
1.



2.



3.

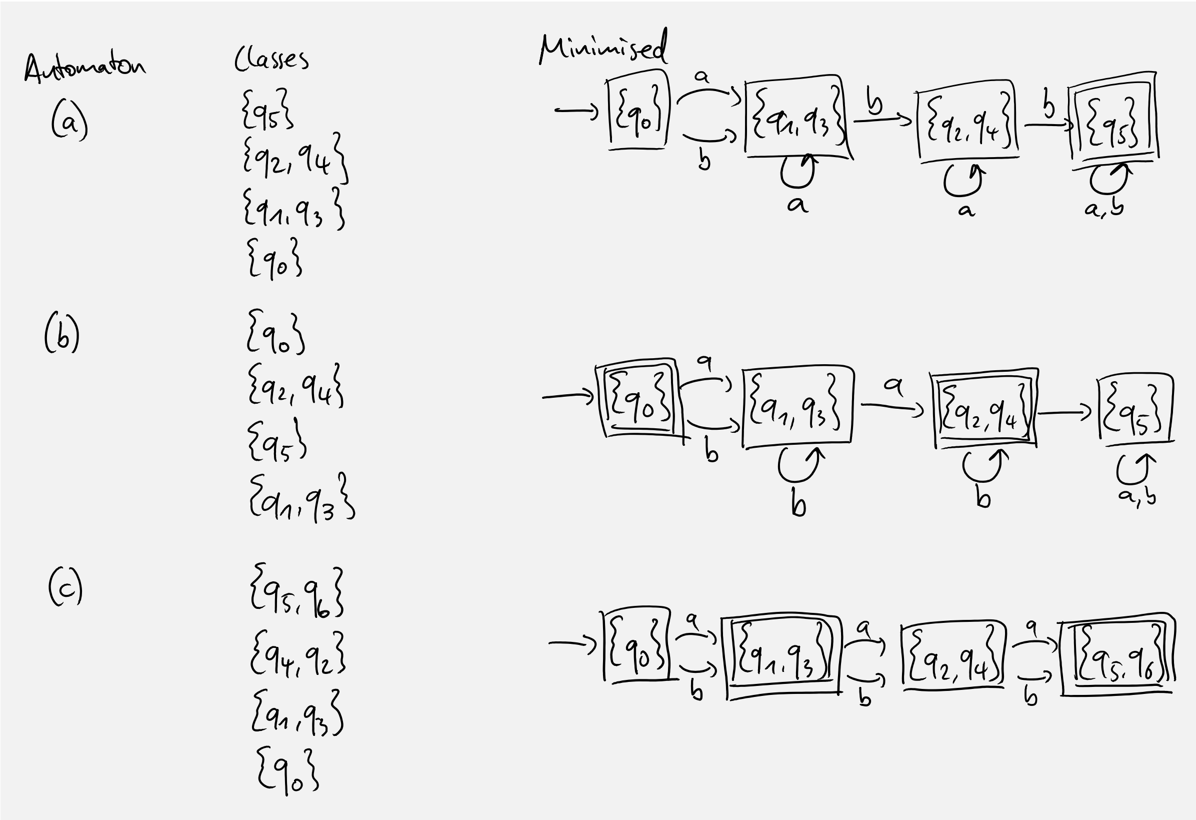


Algorithm 5.7.2
Determination of Equivalent States of DFA

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


Automaton
Classes
Minimised
(a)
{q5}
{q2, q4}
{q1, q3}
{q0}
(b)
{q0}
{q2, q4}
{q5}
{q1, q3}
(c)
{q5, q6}
{q4, q2}
{q1, q3}
{q0}
a
b
a, b