COMPSCI350 (2019)

Automata

Test Preparation

Main points

1. Strings, languages operations with strings (concatenation) and languages (complement, union, intersection, Kleene star, concatenation, power).
2. DFAs: definition by components and by diagram, examples, going from components to diagram and conversely, definition of a computation (trace), showing that a string is accepted by a DFA, showing that a string is rejected by a DFA, constructing a DFA accepting a given language, determining the language accepted by a DFA, proving that a language is not accepted by any DFA.
3. NFAs: definition by components and by diagram, examples, going from components to diagram and conversely, definition of a computation (trace), showing that a string is accepted by an NFA, showing that a string is rejected by an NFA, constructing an NFA accepting a given language, determining the language accepted by an NFA.
4. Constructing a DFA equivalent with a given NFA.
5. The following problems are algorithmically decidable:
    - a DFA M accepts the empty string
    - a DFA M accepts a string w
    - a DFA M accepts no string
    - a DFA M accepts only finitely many strings
    - a DFA M accepts infinitely many strings
    - an NFA N accepts infinitely many strings
    - two DFAs accept the same language
    - a DFA M accepts the complement of the language accepted by a DFA M'
    - a DFA M accepts the same language as an NFA N
    - a DFA M accepts only one string w.
6. The class of regular languages is closed under
    - complement
    - mirror (reverse)
    - union
    - intersection
    - Kleene star
    - concatenation
    - power
7. Regular expressions denote regular languages. Given a DFA M find a regular expression denoting the language accepted by M. Given a regular expression α find a DFA accepting the language denoted by α.
8. Finding the minimal DFA equivalent with a given DFA (NFA).
9. Pattern matching.
10. Revise all examples in lecture notes.

Sample questions[1]

1. All examples in the textbook from pages 31–82.
2. Exercises 1–9, 12,16–20, 29 in the textbook.
3. Build DFAs for the following languages:
   - $\emptyset$,
   - $\{\varepsilon\}$,
   - $\{a^n b^m c^k \mid n \geq 0, m, k \geq 1\}$,
   - $\{1(01)^n \mid n \geq 0\}$,
   - $\{w \in \{a, b\}^* \mid w \neq \varepsilon\}$

   Prove that your solutions are correct.
4. Devise an algorithm that, given a DFA M, produces an equivalent DFA M' in which the start state, once left, cannot be re-entered. Prove that your solution is correct.
5. Prove that the language $A(w) = \{uwv \mid u, v \in \{a, b\}^*\}$ is regular for each string w.
6. Prove that the language $\{a^n b^n c^n \mid n \geq 1\}$ is not accepted by any DFA.
7. Build NFAs for the following languages:
   - $\{w \in \{0, 1\}^* \mid w$ contains any of the substrings $010, 011$ or $1100\}$,
   - $\{w \in \{0, 1\}^* \mid w$ contains the substrings $010, 011$ and $1100\}$,
   - $\{w \in \{0, 1\}^* \mid w$ has a 0 in the third place$\}$,
   - $\{w \in \{0, 1\}^* \mid w$ has a 0 in the third place from the end$\}$,
   - $\{w \in \{a, b\}^* \mid |w| > 2\}$.
   - $\{avb \mid v \in \{a, b\}^*\}$.

   Prove that your solutions are correct.
8. Given two DFAs M_1 and M_2, construct an NFA N such that $L(N) = L(M_1) \setminus L(M_2)$. Justify your answer.
9. Given two DFAs M_1 and M_2, construct an NFA N such that $L(N) = L(M_1) \cup \overline{L(M_2)}$. Justify your answer.
10. Construct an algorithm which given a DFA M and an NFA N tests whether $L(M) = L(N)$. Justify your answer.
11. Using the equivalence between NFA and DFA, convert the following NFAs into equivalent DFAs:

    (a) Every NFA discussed in lecture notes.

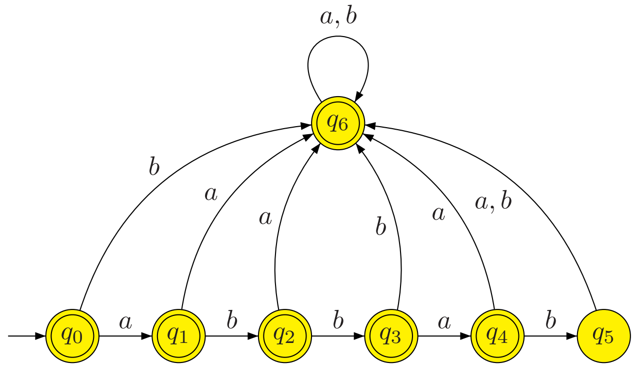
    (b) $Q = \{q_1, q_2\}, \delta(q_1, a) = \{q_1, q_2\}, \delta(q_1, b) = \{q_2\}, \delta(q_2, a) = \emptyset, \delta(q_2, b) = \{q_1\}, S = F = \{q_1\}$.

    (c) $Q = \{q_1, q_2, q_3\}, \delta(q_1, a) = \{q_3\}, \delta(q_1, b) = \emptyset, \delta(q_2, a) = \emptyset, \delta(q_2, b) = \{q_1\}, \delta(q_3, a) = \emptyset, \delta(q_3, b) = \{q_3\}, S = F = \{q_1\}$.
12. Construct regular expressions denoting the languages accepted by each DFA/NFA discussed in lecture notes. Prove that your solutions are correct.
13. Minimise each DFA/NFA discussed in lecture notes. (For NFA, convert first to DFA, then minimise.)
14. Design the Aho-Corasick automaton for a given simple pattern.
15. What is the language denoted by the Kleene regular expression $(a|b)^*$? Justify your answer.
16. What is the language denoted by the Kleene regular expression $ab^*(c|\varepsilon)$? Justify your answer.
17. Write a Kleene regular expression (or NFA or DFA) for the set of all correct email addresses of the form `user@ec.auckland.ac.nz`, where `user` is a string on the alphabet of lower case letter and digits $\{a, b, \ldots, z\} \cup \{0, 1, \ldots, 9\}$ that starts with a letter, is followed by at least three letters and exactly three digits.

[1]Questions in test may be similar, but not identical.

Sample questions with solutions

1. Consider the following DFA:



DFA M

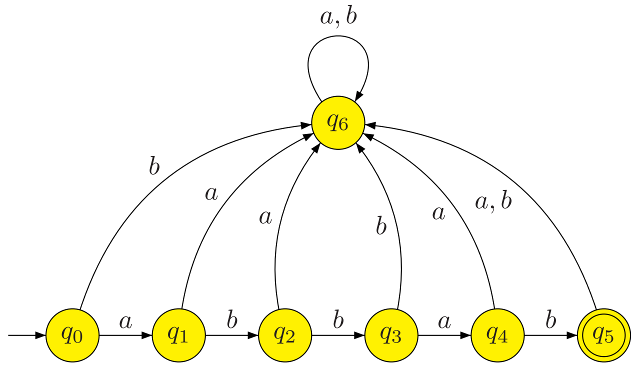
(a) Write the traces (computations) in M for the following strings: $abbab, ab, \varepsilon$. Which of these strings are accepted by M? Justify your answers.

Solution. We have:
$abbab : q_0, q_1, q_2, q_3, q_4, q_5$ is not accepted because q_5 is not final;
$ab : q_0, q_1, q_2$ is accepted because q_2 is final;
$\varepsilon : q_0$ is accepted because q_0 is final.

(b) What is the language accepted by M? Justify your answer.

Solution. We have $L(M) = \{w \in \{a, b\}^* \mid w \neq abbab\}$.

First solution. The language $L(M)$ is the complement of $L(M')$, where M' is



DFA M'

We have proved in class that $L(M') = \{abbab\}$. Because $L(M) = \overline{L(M')}$ we have $L(M) = \{w \in \{a, b\}^* \mid w \neq abbab\}$.

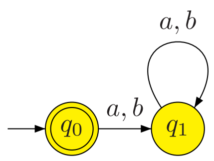
Second solution. As q_5 is the only non final state of M, a string w is not accepted by M if and only if there a trace from q_0 to q_5 labelled by w. There is only one string w satisfying the above property, namely $w = abbab$: every other string is accepted by M, hence $L(M) = \{w \in \{a, b\}^* \mid w \neq abbab\}$.

(c) Is M minimal? Justify your answer.

*Solution.*The DFA M is minimal because: $\equiv_0 = \{\{q_0, q_1, q_2, q_3, q_4, q_6\}, \{q_5\}\}$ and for every $i, j = 0, 1, 2, 3, 4, 6, i \neq j$ there exists a k such that $q_i \not\equiv_k q_j$. For example, $q_4 \not\equiv_1 q_6$ because $\delta(q_4, b) \not\equiv_0 \delta(q_6, b)$. Fill in the missing parts.

2. Prove that there is an algorithm which receives as input a DFA M over the alphabet $\{a,b\}$ and decides whether $L(M) = \{\varepsilon\}$ or $L(M) \neq \{\varepsilon\}$. Clearly state all results you use.

   Solution. It is known from class that there is an algorithm deciding whether two DFAs accept the same language. The language $L = \{\varepsilon\}$ is accepted by the DFA M':



   because the initial state is final and there is no other final state. So, we can apply the above algorithm to the DFAs M and M' to decide whether $L(M) = L(M')$, that is, $L(M) = \{\varepsilon\}$.

3. (a) What is the complement of a language $A \subseteq \{a,b\}^*$?

   Solution. The complement of a language $A \subset \{a,b\}^*$ is the language $\overline{A} = \{x \in \{a,b\}^* \mid x \notin A\}$.

   (b) Construct a DFA M' accepting the complement of the language accepted by the DFA $M = (Q, \{a,b\}, \delta, s, F)$. Justify your construction.

   Solution. The DFA $M' = (Q, \{a,b\}, \delta, s, Q \setminus F)$ accepts the complement of the language accepted by the DFA M because a string $w = w_1 w_2 \dots w_n \in L(M')$ if and only if there is a sequence of states $q_0, q_1, \dots q_n \in Q$ such that $q_0 = s, q_n \in Q \setminus F$ and for each $0 \leq i \leq n-1, q_{i+1} = \delta(q_i, w_{i+1})$ if and only if $w \notin L(M)$.

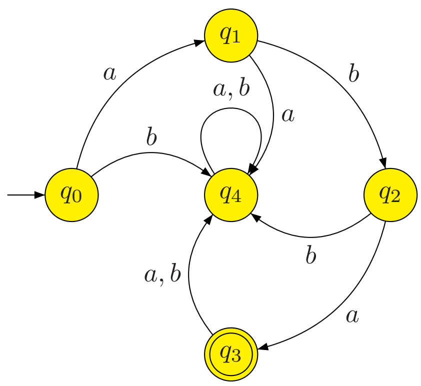
   (c) Construct an NFA N' accepting the complement of the language accepted by the NFA N. Justify your construction. Clearly state all results you use.

   Solution. First we use the result stating that every NFA can be simulated by a DFA to construct a DFA $M = (Q, \{a,b\}, \delta, s, F)$ such $L(M) = L(N)$.
   Then we define $M' = (Q, \{a,b\}, \delta, s, Q \setminus F)$. We know from class that $L(M') = \overline{L(N)}$. Moreover, because the DFA M' is also an NFA we have $L(M') = \overline{L(N)}$.

4. (a) Present an algorithm which tests whether an arbitrary DFA M accepts only finitely many strings.

   Solution. A DFA M accepts only finitely many strings if and only if there is no path from the initial state to a final state which has a loop. The algorithm generates all paths from the initial state to every final state till the first path containing a loop is found; if no path with a loop is found, then the DFA accepts only finitely many strings; otherwise, the DFA accepts infinitely many strings.

   (b) Use your algorithm to test whether the following DFA

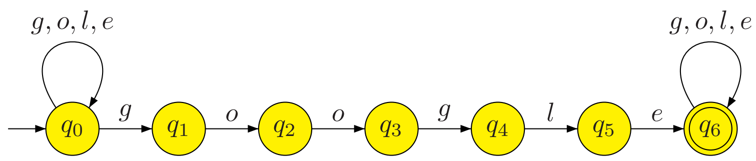


   accepts infinitely many strings.

   Solution. There is a unique path from the initial state q_0 to the (unique) final state q_3, namely q_0, q_1, q_2, q_3, and it contains no loop. Hence, this DFA accepts finitely many strings. In fact it accepts only one string, aba.

5. Construct an NFA N_3 with $\Sigma = \{g, o, l, e\}$ accepting the language $A(P)$ where $P = google$. Recall that $A(P) = \{uPv : u, v \in \Sigma^*\}$.

   Solution.



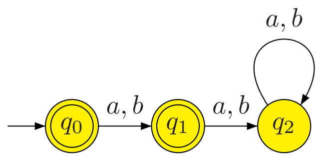
   It is seen that that there is a unique trace from the initial state q_0 to the unique accept state q_6, namely

$$q_0, q_1, q_2, q_3, q_4, q_5, q_6,$$

   corresponding to the string *google*, hence $L(N) = \{google\}$.

6. Prove that there exist infinitely many DFA's each of which recognises exactly the language $\{\varepsilon, a, b\}$.

   Solution. The language $L = \{\varepsilon, a, b\}$ is accepted by the DFA M_1:



   because the initial state of Q_0 M_1 is also final (hence $\varepsilon \in L(M_1)$), a and b are accepted by the traces q_0, q_1 and no other string is accepted: $abu \notin L(M_1)$, for every $u \in \{a, b\}$.

   Moreover, the DFA M_2 which consists of DFA M_1 plus one isolated state q_3 has the property $L(M_2) = L(M_1) = L$. Also DFA M_3 which consists of DFA M_1 plus two isolated states $q_3, q_4, \ldots,$ DFA M_k which consists of DFA M_1 plus $k-2$ isolated states $q_3, q_4, \ldots, q_k, \ldots$, so we have an infinity of distinct DFAs $M_i, i = 1, 2, \ldots$ each of which recognises L.