School of Computing and Information Systems
COMP30026 Models of Computation Tutorial Week 9

17–21 September 2018

Plan

Let's make sure there's enough to practice with, especially for that boring teaching-free week. Exercise 73 will reappear on the exercise sheet for Week 10, because we don't really expect to get that far this week. But if you have time in your tute, do Exercise 73 as well.

The exercises

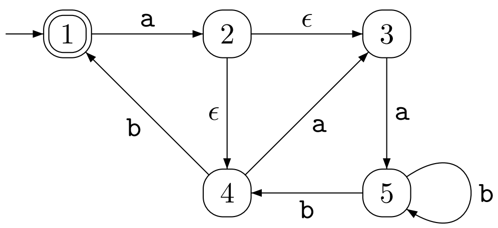
64. Give transition diagrams for NFAs for the following languages, with the specified number of states. Throughout this question, assume that the alphabet $\Sigma = \{\texttt{0}, \texttt{1}\}$.

    (a) $\{w \mid w \text{ ends with } \texttt{00}\}$ using three states

    (b) $\{w \mid w \text{ contains the substring } \texttt{0101}\}$ using five states

    (c) The language $\{\epsilon\}$ with one state

65. Use the subset construction method to turn this NFA into an equivalent DFA:



66. Give regular expressions for the following languages.

    (a) $\{w \mid w \text{ begins with a } \texttt{1} \text{ and ends with a } \texttt{0}\}$

    (b) $\{w \mid w \text{ contains the substring } \texttt{0101}\}$ (so $w = x\texttt{0101}y$ for some strings x and y)

    (c) $\{w \mid w \text{ has length at least 3 and its third symbol is } \texttt{0}\}$

    (d) $\{w \mid \text{the length of } w \text{ is at most } 5\}$

    (e) $\{w \mid w \text{ is any string except } \texttt{11} \text{ and } \texttt{111}\}$

    (f) $\{w \mid \text{every odd position of } w \text{ is a } \texttt{1}\}$

    (g) $\{w \mid w \text{ contains at least two } \texttt{0}\text{s and at most one } \texttt{1}\}$

    (h) $\{\epsilon, \texttt{0}\}$

    (i) The empty set

    (j) All strings except the empty string

67. String s is a *suffix* of string t iff there exists some string u (possibly empty) such that $t = us$. For any language L we can define the set of suffixes of strings in L:

$$\textit{suffix}(L) = \{x \mid x \text{ is a suffix of some } y \in L\}$$

Let A be any regular language. Show that $\textit{suffix}(A)$ is a regular language. Hint: Think about how a DFA for A can be transformed to recognise $\textit{suffix}(A)$.

68. In general it is difficult, given a regular expression, to find a regular expression for its complement. However, it can be done, and you have been given all the necessary tricks and algorithms. This question asks you to go through the required steps for a particular example.

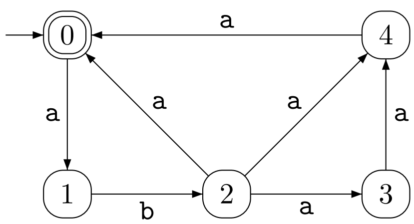
Consider the regular language $(\mathtt{ba}^*\mathtt{a})^*$. Assuming the alphabet is $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, we want to find a regular expression for its complement, that is, for

$$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ is not in } (\mathtt{ba}^*\mathtt{a})^*\}$$

To complete this task, go through the following steps.

(a) Construct an NFA for $(\mathtt{ba}^*\mathtt{a})^*$. Two states suffice.
(b) Turn the NFA into a DFA using the subset construction method.
(c) Do the "complement trick" to get a DFA D for L.
(d) Reflect on the result: Wouldn't it have been better/easier to apply the "complement trick" directly to the NFA?
(e) Turn DFA D into a regular expression for L using the NFA-to-regular-expression translation shown in the lecture on regular expressions (Lecture 16).

69. (Drill.) Use the subset construction method to turn this NFA into an equivalent DFA:



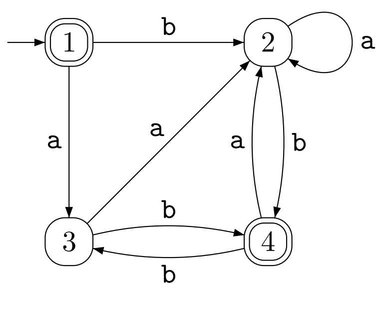
70. (Drill.) Consider the language

$$A = \left\{ w \;\middle|\; \begin{array}{l} w \text{ contains an even number of } \mathtt{a}\text{s and an odd} \\ \text{number of } \mathtt{b}\text{s and does not contain the substring } \mathtt{ab} \end{array} \right\}$$

over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Give a regular expression for A. (You may first want to think of a different way of expressing A in English.)

71. (Drill.) Construct a five-state DFA which recognises A from the previous question.

72. (Optional; DFA minimisation will not be examinable.) Find a minimal DFA which is equivalent to this one:



73. A *palindrome* is a string that reads the same forwards and backwards. Use the pumping lemma for regular languages and/or closure results to prove that the following languages are not regular:

(a) $A = \{\mathtt{0}^n\mathtt{1}^n\mathtt{2}^n \mid n \geq 0\}$
(b) $B = \{\mathtt{a}^i\mathtt{b}\mathtt{a}^j \mid i > j \geq 0\}$
(c) $C = \{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid w \text{ is not a palindrome}\}$