School of Computing and Information Systems
COMP30026 Models of Computation Tutorial Week 8

23–25 September 2020

Plan

This week's exercises cover formal languages, DFAs, NFAs, and minimization. Exercises 65-67 are important because they teach you a systematic approach to building DFAs for intersection, complements and differences of languages.

Some of the exercises on automata come from Sipser, *Introduction to the Theory of Computation.* Chapter 1, on regular languages, is available on Canvas under Readings Online. The book has many examples and it contains many more exercises, plus answers to selected exercises.

The exercises

63. For two languages $L_1 = \{\mathtt{ab}, \mathtt{c}\}$, $L_2 = \{\mathtt{ca}, \mathtt{c}\}$, construct a language:

    (a) $L_1 \cup L_2$

    (b) $L_1 \circ L_2$

    (c) L_1^*

    (d) $L_1^* \backslash L_2^*$

64. Draw DFAs recognising the following languages. Assume that the alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.

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

    (b) $\{w \mid w$ is not empty and contains only $\mathtt{0}$s or only $\mathtt{1}$s$\}$

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

    (d) $\{w \mid w$ has length at least 3 and its third symbol is $\mathtt{0}\}$

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

    (f) $\{w \mid$ the length of w is a multiple of $3\}$

    (g) $\{w \mid w$ is any string except $\mathtt{11}$ and $\mathtt{111}\}$

    (h) $\{w \mid$ every odd position of w is a $\mathtt{1}\}$

    (i) $\{w \mid w$ contains at least two $\mathtt{0}$s and at most one $\mathtt{1}\}$

    (j) $\{w \mid$ the last symbol of w is occured at least twice in $w\}$

    (k) $\{\epsilon, \mathtt{0}\}$

    (l) The empty set

    (m) All strings except the empty string

65. Each of the following languages is the intersection of two simpler languages. First construct the DFAs for the simpler languages, then combine them using the following idea: If the set of states for DFA D_1 is Q_1 and the set of states for D_2 is Q_2, we let the set of states for the combined DFA D be $Q_1 \times Q_2$. We construct D so that, having consumed a string s, D will be in state (q_1, q_2) iff D_1 is in state q_1, and D_2 is in state q_2 when they have consumed s. Throughout this question, assume that the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

    (a) $\{w \mid w$ has at least three $\mathtt{a}$s and at least two $\mathtt{b}$s$\}$

(b) $\{w \mid w$ has an even number of as and one or two bs$\}$

(c) $\{w \mid w$ has an odd number of as and ends with b$\}$

(d) $\{w \mid w$ has an odd number of as and has even length$\}$

66. Each of the following languages is the complement of a simpler language. Again, the best way to proceed is to first construct a DFA for the simpler language, then find a DFA for the complement by transforming that DFA appropriately. Throughout this question, assume that the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

(a) $\{w \mid w$ does not contain the substring bb$\}$

(b) $\{w \mid w$ contains neither the substring ab nor ba$\}$

(c) $\{w \mid w$ is any string not in $\mathtt{A}^* \circ \mathtt{B}^*$, where $\mathtt{A} = \{\mathtt{a}\}, \mathtt{B} = \{\mathtt{b}\}\}$

(d) $\{w \mid w$ is any string not in $\mathtt{A}^* \cup \mathtt{B}^*$, where $\mathtt{A} = \{\mathtt{a}\}, \mathtt{B} = \{\mathtt{b}\}\}$

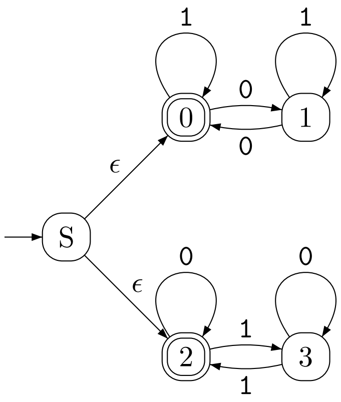
(e) $\{w \mid w$ is any string that doesn't contain exactly two as$\}$

(f) $\{w \mid w$ is any string except a and b$\}$

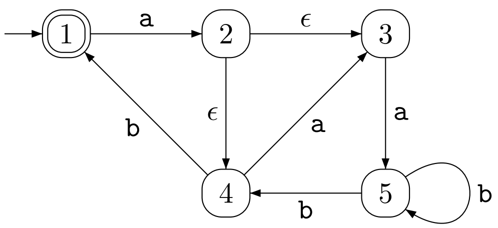
67. The following language is the difference of two simpler languages. First construct DFAs for simpler languages. Assume that the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

$\{w \mid$ the lenght of w is a multiple of 2 and is not multiple of 3$\}$

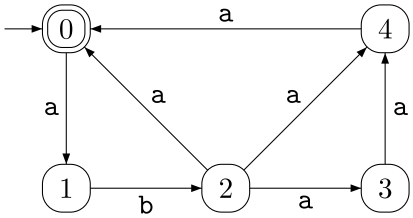
68. (An example from Lecture 7). Use the subset construction method to turn this NFA into an equivalent DFA:



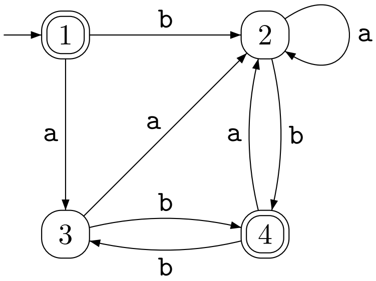
69. Use the subset construction method to turn this NFA into an equivalent DFA:



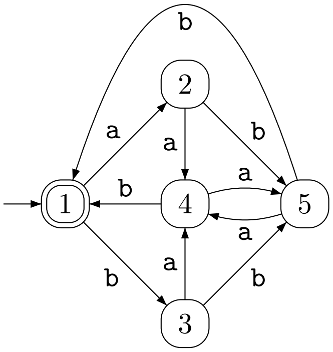
70. Use the subset construction method to turn this NFA into an equivalent DFA:



71. Find a minimal DFA which is equivalent to this one:



72. Find a minimal DFA which is equivalent to this one:


b
2
a
a
b
a
1
b
4
5
a
a
b
b
3