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

10–14 September 2018

Plan

This week's exercises cover termination and DFAs. Exercises 61 and 62 are important because they teach you a systematic approach to building DFAs for intersections and complements 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 the LMS under Readings Online. The book has many examples and it contains many more exercises, plus answers to selected exercises.

The exercises

58. (Optional.) Lecture 12 introduced the function $c : \mathbb{N} \hookrightarrow \mathbb{N}$, defined recursively like so:

$$c(n) = \begin{cases} 1 & \text{if } n = 0 \text{ or } n = 1 \\ c(n/2) & \text{if } n \text{ is even and } n > 1 \\ c(3n+1) & \text{if } n \text{ is odd and } n > 1 \end{cases}$$

Write a Haskell function `hailstone :: Integer -> Int` which calculates the number of recursive calls made when computing $c(n)$. For example, `hailstone 5` should evaluate to 5, and `hailstone 27` should evaluate to 111.

c is known to terminate for all natural numbers up to 10^{20}. It is conjectured to terminate for all $n \in \mathbb{N}$, but whether this is is actually the case is an open problem. There are examples where similar conjectures have been refuted. One famous example has to do with prime factorisations. Say that $n > 1$ is *peven* if its prime factorisation has an even number of factors; otherwise n is *podd*. So $28 = 2 \cdot 2 \cdot 7$ is podd, and $40 = 2 \cdot 2 \cdot 2 \cdot 5$ is peven. Pólya conjectured that, for any k, the set $\{2, 3, 4, \ldots, k\}$ never has a majority of peven elements. However, that turned out to be false, the smallest counter-example being $k = 906150257$.

59. Recall that a binary relation $\prec$ over set S is a well-founded relation iff there is no infinite sequence $s_0, s_1, s_2, s_3, \ldots$ such that $s_i \succ s_{i+1}$ for all $i \in \mathbb{N}$. That is, each sequence of elements from S, when listed in decreasing order, is finite. For each of the following, say whether it is well-founded:

    (a) The usual "smaller than" relation, $<$, on the natural numbers $\mathbb{N}$.

    (b) The usual "smaller than" relation, $<$, on the rational numbers, $\mathbb{Q}$.

    (c) The relation "is a proper sublist of" on the set of lists.

    (d) The (strict) lexicographic ordering of pairs of natural numbers, that is, the relation $\prec$ defined by $(m, m') \prec (n, n')$ iff $m < n \vee (m = n \wedge m' < n')$.

    For the last question, it may help to draw the Hasse diagram for the partially ordered set $\mathbb{N} \times \mathbb{N}$, ordered by $\preceq$, the reflexive closure of $\prec$.

60. You have a bag of ($n > 0$) coloured marbles. There are three colours: red, blue, and white, and on the table, next to the bag, is a huge box with marbles of all three colours, enough that you never run out. Now repeat the following process:

   (a) If the bag contains at most one marble, halt; otherwise remove two marbles from the bag (without looking).
   (b) If one of the two marbles is red, move both to the box.
   (c) If both are white, put one of them back into the bag, together with 5 blue marbles from the box (the other white marble goes in the box).
   (d) Otherwise move both to the box, and move 10 red marbles from the box to the bag.

   Show that the process must halt.

61. 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 `a`s and at least two `b`s$\}$
   (b) $\{w \mid w$ has an even number of `a`s and one or two `b`s$\}$
   (c) $\{w \mid w$ has an odd number of `a`s and ends with `b`$\}$
   (d) $\{w \mid w$ has an odd number of `a`s and has even length$\}$

62. 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 $\mathtt{a}^*\mathtt{b}^*\}$
   (d) $\{w \mid w$ is any string not in $\mathtt{a}^* \cup \mathtt{b}^*\}$
   (e) $\{w \mid w$ is any string that doesn't contain exactly two `a`s$\}$
   (f) $\{w \mid w$ is any string except `a` and `b`$\}$

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

   (a) $\{w \mid w$ begins with a `1` and ends with a `0`$\}$
   (b) $\{w \mid w$ contains the substring `0101`$\}$ (so $w = x\mathtt{0101}y$ for some strings x and y)
   (c) $\{w \mid w$ has length at least 3 and its third symbol is `0`$\}$
   (d) $\{w \mid$ the length of w is at most 5$\}$
   (e) $\{w \mid w$ is any string except `11` and `111`$\}$
   (f) $\{w \mid$ every odd position of w is a `1`$\}$
   (g) $\{w \mid w$ contains at least two `0`s and at most one `1`$\}$
   (h) $\{\epsilon, \mathtt{0}\}$
   (i) The empty set
   (j) All strings except the empty string