School of Computing and Information Systems
COMP30026 Models of Computation
Week 10: Turing Machines

Exercises

T10.1 The following Turing machine D was written to perform certain manipulations to its input—it isn't intended as a recogniser for a language, and so we don't bother to identify an accept or a reject state. The machine stops when no transition is possible, and whatever is on its tape at that point is considered output.

D's set of states is $\{q_0, q_1, q_2, q_3, q_4\}$, with q_0 being the initial state. The input alphabet is $\{1\}$ and the tape alphabet is $\{1, \mathtt{x}, \mathtt{z}, \sqcup\}$, where, as usual, $\sqcup$ stands for 'blank', or absence of a proper symbol. D's transition function δ is defined like so:

$$\begin{array}{lcl@{\qquad}lcl@{\qquad}lcl}
\delta(q_0, 1) & = & (q_1, \mathtt{z}, R) & \delta(q_1, \sqcup) & = & (q_2, 1, L) & \delta(q_2, \mathtt{z}) & = & (q_4, 1, L) \\
\delta(q_0, \sqcup) & = & (q_4, \sqcup, L) & \delta(q_2, 1) & = & (q_2, 1, L) & \delta(q_3, 1) & = & (q_3, 1, R) \\
\delta(q_1, 1) & = & (q_1, \mathtt{x}, R) & \delta(q_2, \mathtt{x}) & = & (q_3, 1, R) & \delta(q_3, \sqcup) & = & (q_2, 1, L)
\end{array}$$

Draw D's diagram and determine what D does to its input.

T10.2 Consider the following language over the alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}, \#\}$,

$$A = \{w\#w \mid w \in \{\mathtt{0}, \mathtt{1}\}^*\}$$

Show that this language is decidable by following these steps:

(i) Describe, using pseudocode, the behaviour of a Turing machine that recognises A.

(ii) Draw a Turing machine that does exactly what you described in your pseudocode. You may omit the reject state, to which all missing transitions are assumed to go.

(iii) Argue that your Turing machine will halt on every input.

Continued in P10.6, P10.7

T10.3 Here is how we can see that the class of decidable languages is closed under intersection. Let A and B be decidable languages, let M_A be a decider for A and M_B a decider for B. We construct a decider for $A \cap B$ as a Turing machine which implements this routine:

On input w:
(1) Run M_A on input w and reject if M_A rejects.
(2) Run M_B on input w and reject if M_B rejects; else accept.

Show that the class of decidable languages is closed under union.

Continued in P10.1, P10.2 & P10.3

T10.4 The class of Turing recognisable languages is closed under intersection, and the construction we gave in the previous question, for decidable languages, can equally be used to prove this. (Convince yourself that the argument is still right, even though we now have no guarantee that M_A and M_B always terminate.)

The class of Turing recognisable languages is also closed under union, but we can't argue that the same way, by constructing a Turing machine which effectively runs M_A and M_B, one after the other. (Why not?)

Show that the class of Turing recognisable languages is closed under union.

Homework problems

P10.1 Show that the class of decidable languages is closed under complement. Why can't we use the same argument to show that the class of Turing recognisable languages is closed under complement?

P10.2 Show that the class of decidable languages is closed under concatenation.

P10.3 Show that the class of decidable languages is closed under Kleene star.

P10.4 A 2-PDA is a pushdown automaton that has two stacks instead of one. In each transition step it may consume an input symbol, pop and/or push to stack 1, and pop an/or push to stack 2. It can also leave out any of these options (using ϵ moves) just like the standard PDA.

In the lectures, we used the pumping lemma for context-free languages to established that the language $B = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \in \mathbb{N}\}$ is not context-free. However, B has a 2-PDA that recognises it. Outline in English or pseudo-code how that 2-PDA operates.

P10.5 In fact, a 2-PDA is as powerful as a Turing machine. Outline an argument for this proposition by showing how a 2-PDA can simulate a given Turing machine. *Hint:* arrange things so that, at any point during simulation, the two stacks together hold the contents of the Turing machine's tape, and the symbol under the tape head sits on top of one of the stacks.

P10.6 For each of the following languages, write an algorithm in pseudocode which describes a Turing machine that decides the following languages. Assume that $\Sigma = \{\mathtt{0}, \mathtt{1}\}$

(i) $\{w \mid w$ has an equal number of `0`'s and `1`'s$\}$

(ii) $\{w \mid w$ has twice as many `0`'s as `1`'s$\}$

(iii) $\{w \mid w$ does not have twice as many `0`'s as `1`'s$\}$

P10.7 For each language in P10.6, draw a Turing machine that carries out the pseudocode. Then write down the sequence of configurations your TM goes through on an interesting input.

P10.8 Show that the problem of whether the language of a DFA is empty, is decidable. That is, show that the language

$$E_{DFA} = \{\langle D\rangle \mid D \text{ is a DFA and } L(D) = \emptyset\}$$

is decidable. *Hint:* write pseudocode for an algorithm which analyses the graph of the DFA, and argue that your algorithm will not run forever on any input DFA $\langle D\rangle$.

P10.9 Show that the problem of whether the language of a CFG is empty, is decidable. That is, show that the language

$$E_{CFG} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \emptyset\}$$

is decidable. *Hint:* write pseudocode for an algorithm which analyses the rules of the CFG, and argue that your algorithm will not run forever on any input CFG $\langle G\rangle$.