School of Computing and Information Systems
COMP30026 Models of Computation Tutorial Week 12

16–20 October 2017

The plan

Sadly, the last tutorial session is here. We have carried Exercise 82 over from Week 11; otherwise the last exercises are about computability, decidability, reduction and simulation.

The exercises

82. Construct a push-down automaton which recognises the language $M = \{\mathtt{a}^i\mathtt{b}\mathtt{a}^j \mid i > j \geq 0\}$.

84. 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:

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

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

85. Show that the halting problem for Turing machines is undecidable. More precisely, show that the language

$$\mathit{Halt}_{TM} = \{\langle M, w\rangle \mid M \text{ is a Turing machine and } M \text{ halts when run on input } w\}$$

    is undecidable. Hint: Use reduction from A_{TM}, that is, show that if we did have a decider for Halt_{TM} then we could also build a decider for A_{TM}.

86. Consider the problem of whether a given context-free grammar with alphabet $\{\mathtt{0}, \mathtt{1}\}$ is able to generate a string in $\mathtt{1}^*$. Is that decidable? In other words, is the language

$$\{\langle G\rangle \mid G \text{ is a CFG over } \{\mathtt{0}, \mathtt{1}\} \text{ and } L(G) \cap \mathtt{1}^* \neq \emptyset\}$$

    decidable? Hint: Consider known closure properties for context-free languages.

87. 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 Lecture 17 we used the pumping lemma for context-free languages to established that the language $B = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ is not context-free. However, B has a 2-PDA that recognises it. Outline in English or pseudo-code how that 2-PDA operates.

88. 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.

89. (Optional.) Consider the alphabet $\Sigma = \{0, 1\}$. The set Σ^* consists of all the *finite* bit strings, and the set, while infinite, turns out to be countable. (At first this may seem obvious, since we can use the function $binary : \mathbb{N} \to \Sigma^*$ defined by

$$binary(n) = \text{ the binary representation of } n$$

as enumerator; however, that is not a surjective function, because the legitimate use of leading zeros means there is no unique binary representation of n. For example, both 101 and 00101 denote 5. Instead the idea is to list all binary strings of length 0, then those of length 1, then those of length 2, and so on: $\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, \ldots$.)

Now consider instead the set $\mathcal{B}$ of *infinite* bit strings. Show that $\mathcal{B}$ is much larger than Σ^*. More specifically, use diagonalisation to show that $\mathcal{B}$ is not countable.

90. (Optional.) Perhaps surprisingly, a Post machine can simulate a Turing machine. We can think of the '\$' marker as marking the end of (the contents of) a Turing machine tape. If the Post machine queue is of the form $v\$w$, we think of this as representing a Turing machine tape containing wv, with the tape head positioned over the first symbol in v. (If $v = \epsilon$, that indicates the tape head is positioned over the blank cell immediately to the right of w.) For example, the queue $abb\$ba$ represents the tape $baabb$, with the tape head sitting over the second a. (The front of the queue is to the left.)

Given this representation, show how a Post machine can simulate the operations of a Turing machine. A move '$x \to y, R$' is easy to simulate with dequeueing and en-queueing. However, a move '$x \to y, L$' is harder to handle. (Hint: You can use a sequence of left rotations to create the effect of a right rotation, but to have sufficient control, you may want to add some additional marker symbols to the queue alphabet.) Assume that the Turing machine is two-way, so there is no concern about a left-move possibly hitting the end of the tape.