School of Computing and Information Systems
COMP30026 Models of Computation Tutorial Week 11

8–12 October 2018

The plan

Try to get through all of this week's exercises. Reminder: A good text on context-free languages is available under "Readings Online".

The exercises

85. Give a context-free grammar for $\{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i = j \vee j = k \text{ where } i, j, k \geq 0\}$. Is your grammar ambiguous? Why or why not?

86. Consider the context-free grammar $G = (\{S, A, B\}, \{\mathtt{a}, \mathtt{b}\}, R, S)$ with rules R:

$$\begin{array}{lcl} S & \rightarrow & A\ B\ A \\ A & \rightarrow & \mathtt{a}\ A \mid \epsilon \\ B & \rightarrow & \mathtt{b}\ B \mid \epsilon \end{array}$$

   (a) Show that G is ambiguous.

   (b) The language generated by G is regular; give a regular expression for $L(G)$.

   (c) Give an unambiguous context-free grammar, equivalent to G. Hint: As an intermediate step, you may want to build a DFA for $L(G)$.

87. Construct a push-down automaton which recognises the language from Exercise 75, that is, $\{\mathtt{a}^i\mathtt{b}\mathtt{a}^j \mid i > j \geq 0\}$.

88. We have seen that the set of context-free languages is not closed under intersection. However, it *is* closed under intersection with regular languages. That is, if L is context-free and R is regular then $L \cap R$ is context-free.

    We can show this if we can show how to construct a push-down automaton P' for $L \cap R$ from a push-down automaton P for L and a DFA D for R. The idea is that we can do something similar to what we did in Exercise 61 when we built "product automata", that is DFAs for languages $R_1 \cap R_2$ where R_1 and R_2 were regular languages. If P has state set Q_P and D has state set Q_D, then P' will have state set $Q_P \times Q_D$.

    More precisely, let $P = (Q_P, \Sigma, \Gamma, \delta_P, q_P, F_P)$ and let $D = (Q_D, \Sigma, \delta_D, q_D, F_D)$. Recall the types of the transition functions:

$$\begin{array}{l} \delta_P : (Q_P \times \Sigma_\epsilon \times \Gamma_\epsilon) \rightarrow \mathcal{P}(Q_P \times \Gamma_\epsilon) \\ \delta_D : (Q_D \times \Sigma) \rightarrow Q_D \end{array}$$

    We construct P' with the following components: $P' = (Q_P \times Q_D, \Sigma, \Gamma, \delta, (q_P, q_D), F_P \times F_D)$. Give a formal definition of δ, the transition function for P'.

89. (a) Consider the language $A = \{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^i\mathtt{b}^j \mid i \geq 0 \wedge j \geq 0\}$. Use the pumping lemma for context-free languages to show that A is not context-free.

   (b) Now consider $B = \{\mathtt{a}^i\mathtt{b}^j\mathtt{a}^j\mathtt{b}^i \mid i \geq 0 \wedge j \geq 0\}$. Give a context-free grammar for B.

   (c) A and B look very similar. We might try to prove B not context-free by doing what we did to prove that A is not context-free. Where does the attempted proof fail?

90. 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:

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

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