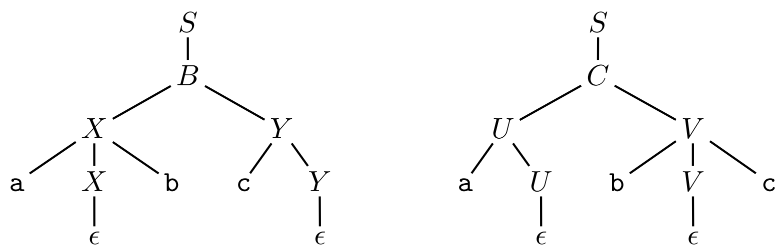
The University of Melbourne
School of Computing and Information Systems
COMP30026 Models of Computation

Selected Tutorial Solutions, Week 11

85. To find a context-free grammar for $\{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i = j \vee j = k \text{ where } i, j, k \geq 0\}$ we note that the language is the union of two context-free languages, generated by the two context-free grammars

$$\begin{array}{lll} B & \rightarrow & X\ Y \\ X & \rightarrow & \epsilon \mid \mathtt{a}\ X\ \mathtt{b} \\ Y & \rightarrow & \epsilon \mid \mathtt{c}\ Y \end{array} \qquad\qquad \begin{array}{lll} C & \rightarrow & U\ V \\ U & \rightarrow & \epsilon \mid \mathtt{a}\ U \\ V & \rightarrow & \epsilon \mid \mathtt{b}\ V\ \mathtt{c} \end{array}$$

Hence we get a context-free grammar for the language by adding the rule $S \rightarrow B \mid C$ and making S the start symbol. The grammar is ambiguous. We get two different parse trees for any string of form $\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n$. For example, for abc:



86. We are looking at the context-free grammar G:

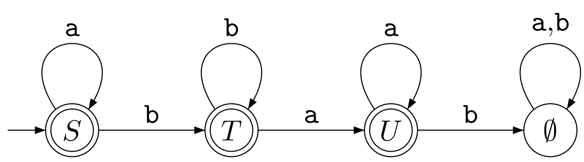
$$\begin{array}{lll} S & \rightarrow & A\ B\ A \\ A & \rightarrow & \mathtt{a}\ A \mid \epsilon \\ B & \rightarrow & \mathtt{b}\ B \mid \epsilon \end{array}$$

(a) The grammar is ambiguous. For example, a has two different leftmost derivations:

$$S \Rightarrow A\ B\ A \Rightarrow B\ A \Rightarrow A \Rightarrow \mathtt{a}\ A \Rightarrow \mathtt{a}$$
$$S \Rightarrow A\ B\ A \Rightarrow \mathtt{a}\ A\ B\ A \Rightarrow \mathtt{a}\ B\ A \Rightarrow \mathtt{a}\ A \Rightarrow \mathtt{a}$$

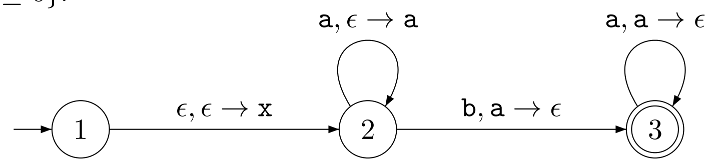
(b) $L(G) = \mathtt{a}^*\mathtt{b}^*\mathtt{a}^*$.

(c) To find an unambiguous equivalent context-free grammar it helps to build a DFA for $\mathtt{a}^*\mathtt{b}^*\mathtt{a}^*$. (If this is too hard, we can always construct an NFA, which is easy, and then translate the NFA to a DFA using the subset construction method, which is also easy.) Below is the DFA we end up with. The states are named S, T, and U to suggest how they can be made to correspond to variables in a context-free grammar. The DFA translates easily to the grammar on the right. The resulting grammar is a so-called *regular* grammar, and it is easy to see that it is unambiguous—there is never a choice of rule to use.



$$\begin{array}{lll} S & \rightarrow & \epsilon \mid \mathtt{a}\ S \mid \mathtt{b}\ T \\ T & \rightarrow & \epsilon \mid \mathtt{b}\ T \mid \mathtt{a}\ U \\ U & \rightarrow & \epsilon \mid \mathtt{a}\ U \end{array}$$

87. Here is a PDA for $\{\mathsf{a}^i\mathsf{b}\mathsf{a}^j \mid i > j \geq 0\}$:



Note that the stack won't be empty when this PDA halts; and that's okay.

88. For the case $v \neq \epsilon$ we define

$$\delta((q_p, q_d), v, x) = \left\{ \ ((r_p, r_d), y) \,\middle|\, (r_p, y) \in \delta_P(q_p, v, x) \wedge r_d = \delta_D(q_d, v) \ \right\}$$

But we must also allow transitions that don't consume input, so:

$$\delta((q_p, q_d), \epsilon, x) = \left\{ \ ((r_p, q_d), y) \,\middle|\, (r_p, y) \in \delta_P(q_p, \epsilon, x) \ \right\}$$

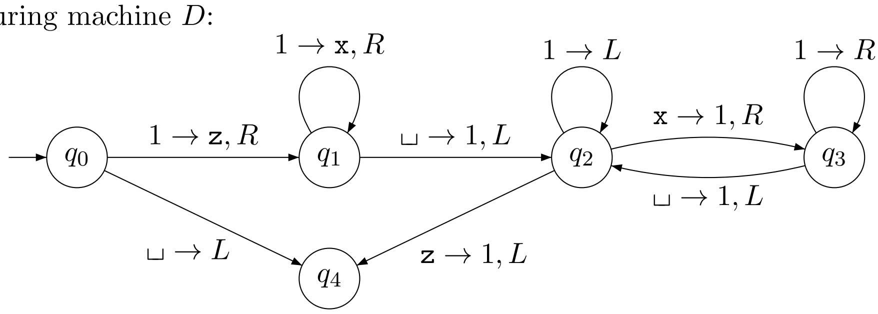
89. (a) Assume that A is context-free and let p be the pumping length. Consider the string $\mathsf{a}^p\mathsf{b}^p\mathsf{a}^p\mathsf{b}^p \in A$. The pumping lemma tells us that the string can be written $uvxyz$, with v and y not both empty, and with $|vxy| \leq p$, such that $uv^ixy^iz \in A$ for all i. Clearly, if one (or both) of v and y contains an a as well as a b then pumping up will lead to a string that is not in A, as the result will have more than two substrings ab. So each of v and y must contain as only, or bs only, unless it is empty. If neither v nor y contains a b then both must come from the same a^i segment (the first or the last), because $|vxy| \leq p$; and then, when we pump up, that segment alone grows, while the other a^i segment is untouched. Similarly, if neither v nor y contains an a. So in these cases the result of pumping is not in A. The only remaining cases are when v is from the first a^i segment and y is from the first b^j segment, or v is from the first b^j segment and y is from the second a^i segment, or v is from the second a^i segment and y is from the second b^j segment. In each case, pumping up will take the string outside A. We conclude that A is not context-free.

(b) Here is a context-free grammar for B:

$$\begin{array}{lcl} S & \rightarrow & T \mid \mathsf{a}\ S\ \mathsf{b} \\ T & \rightarrow & \epsilon \mid \mathsf{b}\ T\ \mathsf{a} \end{array}$$

(c) If we pick the obvious candidate string $\mathsf{a}^p\mathsf{b}^p\mathsf{a}^p\mathsf{b}^p \in B$, we fail to get a contradiction with the pumping lemma. Namely we might have $v = \mathsf{b}^k$ and $y = \mathsf{a}^k$ $(0 < k \leq p/2)$ where v is a substring of the first b^j segment and y is a substring of the second a^j segment. In that case, the result of pumping (up or down) is in B, and we don't get a contradiction.

90. Here is the Turing machine D:



Note that, for the two transitions into q_4, the 'L' has no effect, since, at that point, the tape head is positioned over the leftmost tape cell. The machine doubles the number of ones that it finds on the tape.