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

Selected Tutorial Solutions, Week 11

75. Here are the context-free grammars:

    (a) $\{w \mid w$ starts and ends with the same symbol$\}$:

    $$\begin{array}{lcl} S & \rightarrow & 0\,T\,0 \mid 1\,T\,1 \mid 0 \mid 1 \\ T & \rightarrow & 0\,T \mid 1\,T \mid \epsilon \end{array}$$

    (b) $\{w \mid$ the length of w is odd$\}$:

    $$S \rightarrow 0 \mid 1 \mid 0\,0\,S \mid 0\,1\,S \mid 1\,0\,S \mid 1\,1\,S$$

    (c) $\{w \mid$ the length of w is odd and its middle symbol is $\mathtt{0}\}$:

    $$S \rightarrow 0 \mid 0\,S\,0 \mid 0\,S\,1 \mid 1\,S\,0 \mid 1\,S\,1$$

    (d) $\{w \mid w$ is a palindrome$\}$:

    $$S \rightarrow 0\,S\,0 \mid 1\,S\,1 \mid 0 \mid 1 \mid \epsilon$$

76. Here is a context-free grammar for M:

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

    The grammar is ambiguous; for example, aaaabb has two different parse trees.

77. The class of context-free languages is closed under the regular operations: union, concatenation, and Kleene star.

    Let G_1 and G_2 be context-free grammars generating L_1 and L_2, respectively. First, if necessary, rename variables in G_2 so that the two grammars have no variables in common. Let the start variables of G_1 and G_2 be S_1 and S_2, respectively. Then we get a CFG for $L_1 \cup L_2$ by keeping the rules from G_1 and G_2, adding

    $$\begin{array}{lcl} S & \rightarrow & S_1 \\ S & \rightarrow & S_2 \end{array}$$

    where S is a fresh variable, and making S the new start variable.

    We can do exactly the same sort of thing for $L_1 \circ L_2$. The only difference is that we now just add one rule:

    $$S \rightarrow S_1\,S_2$$

    again making (the fresh) S the new start variable.

    Let G be a CFG for L and let S be fresh. If we add two rules to those from G:

    $$\begin{array}{lcl} S & \rightarrow & \epsilon \\ S & \rightarrow & S\,S' \end{array}$$

    where S' is G's start variable, then we have a CFG for L^* (it has the fresh S as its start variable).

78. Here are some sentences generated from the grammar:

   (a) A dog runs
   (b) A dog likes a bone
   (c) The quick dog chases the lazy cat
   (d) A lazy bone chases a cat
   (e) The lazy cat hides
   (f) The lazy cat hides a bone

   The grammar is concerned with the structure of well-formed sentences; it says nothing about meaning. A sentence such as "a lazy bone chases a cat" is syntactically correct—its structure makes sense; it could even be semantically correct, for example, "lazy bone" may be a derogatory characterisation of some person. But in general there is no guarantee that a well-formed sentence carries meaning.

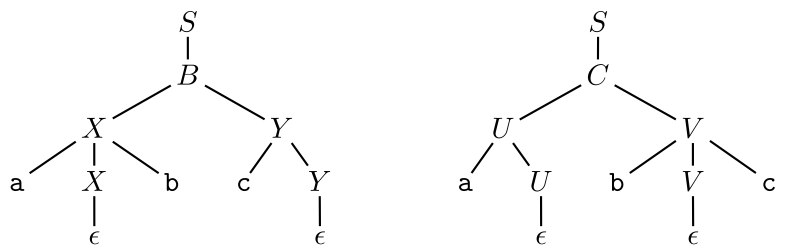
79. We can easily extend the grammar so that a sentence may end with an optional adverbial modifier:

$$\begin{array}{lcl} S & \rightarrow & NP\ VP\ PP \\ & \vdots & \\ PP & \rightarrow & \epsilon \\ PP & \rightarrow & \text{quietly} \\ PP & \rightarrow & \text{all day} \\ & \vdots & \end{array}$$

80. 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 CFGs

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

   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`:



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

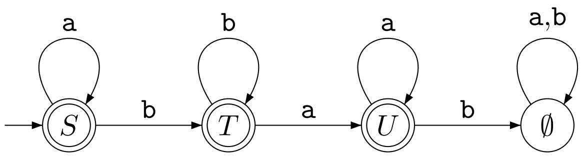
$$\begin{array}{lcl} S & \to & A\ B\ A \\ A & \to & \mathtt{a}\ A \mid \epsilon \\ B & \to & \mathtt{b}\ B \mid \epsilon \end{array}$$

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

$$\begin{array}{l} S \Rightarrow A\ B\ A \Rightarrow B\ A \Rightarrow A \Rightarrow \mathtt{a}\ A \Rightarrow \mathtt{a} \\ S \Rightarrow A\ B\ A \Rightarrow \mathtt{a}\ A\ B\ A \Rightarrow \mathtt{a}\ B\ A \Rightarrow \mathtt{a}\ A \Rightarrow \mathtt{a} \end{array}$$

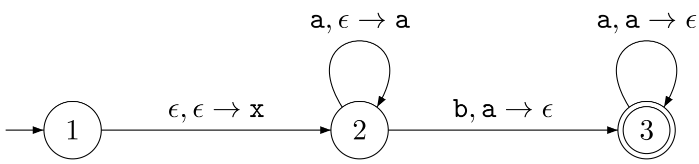
(b) $L(G) = \mathtt{a}^*\mathtt{b}^*\mathtt{a}^*$.

(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.



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

82. Here is a PDA for $M = \{\mathtt{a}^i\mathtt{b}\mathtt{a}^j \mid i > j \geq 0\}$:



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