

**MAU22C00: TUTORIAL 7 SOLUTIONS**  
**FORMAL LANGUAGES AND GRAMMARS**

1) Let the formal language  $L$  over the alphabet  $\{a, b, c\}$  be generated by the context-free grammar whose only non-terminal is  $\langle S \rangle$ , whose start symbol is  $\langle S \rangle$ , and whose production rules are the following:

- (1)  $\langle S \rangle \rightarrow b$
- (2)  $\langle S \rangle \rightarrow c$
- (3)  $\langle S \rangle \rightarrow a\langle S \rangle$

- (a) Describe  $L$ . In other words, describe the structure of the strings generated by this grammar.
- (b) Which words does  $aa\langle S \rangle$  directly yield?
- (c) Which words does  $aa\langle S \rangle$  yield?

**Solution:** (a) This context-free grammar produces strings of the type  $a^m b$  or  $a^m c$  for  $m \geq 0$ , so

$$L = \{a^m b \mid m \in \mathbb{N}, m \geq 0\} \cup \{a^m c \mid m \in \mathbb{N}, m \geq 0\}.$$

(b) We are being asked which words we can obtain from  $aa\langle S \rangle$  via the application of one production rule. The context-free grammar has three production rules, and the left-hand side of each one of these consists of the non-terminal  $\langle S \rangle$ , which appears exactly once in the word  $aa\langle S \rangle$  that we are given. Therefore, we expect  $aa\langle S \rangle$  to directly yield three different words. Indeed, via rule (1) we obtain  $aab$ , via rule (2) we obtain  $aac$ , and via rule (3) we obtain  $aaa\langle S \rangle$ .

(c) In this question part, we are being asked which words we can obtain from  $aa\langle S \rangle$  via the application of finitely many production rules. We saw that the application of one production rule gave us a set of three words  $\{aab, aac, aaa\langle S \rangle\}$ . The first two consist of terminals, so only in the last one we can apply another production rule. In fact, in  $aaa\langle S \rangle$  we can apply each of our three production rules to get  $\{aaab, aaac, aaaa\langle S \rangle\}$ . Thus, after two steps, we have obtained the set of words  $\{aab, aac, aaab, aaac, aaaa\langle S \rangle\}$ . By the same analysis, after three steps we will have obtained

$$\{aab, aac, aaab, aaac, aaaab, aaaac, aaaaa\langle S \rangle\},$$

and so on. After  $n$  steps, we will have the set

$$\{a^p b \mid 2 \leq p \leq n+1\} \cup \{a^p c \mid 2 \leq p \leq n+1\} \cup \{a^{n+2}\langle S \rangle\}.$$

Since we are allowed to have any finite number of steps, the set of words that  $aa\langle S \rangle$  yields is given by

$$\bigcup_{n \geq 1} \{a^p b \mid 2 \leq p \leq n+1\} \cup \{a^p c \mid 2 \leq p \leq n+1\} \cup \{a^{n+2} \langle S \rangle\},$$

which equals

$$\{a^p b \mid p \geq 2\} \cup \{a^p c \mid p \geq 2\} \cup \{a^p \langle S \rangle \mid p \geq 3\}.$$

2) Consider the binary alphabet  $\{0, 1\}$ , start symbol  $\langle S \rangle$ , set of non-terminals consisting of  $\{\langle S \rangle, \langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle, \langle E \rangle\}$ , and production rules given by

- (1)  $\langle S \rangle \rightarrow \langle A \rangle \langle B \rangle \langle C \rangle$
- (2)  $\langle A \rangle \langle B \rangle \rightarrow 0 \langle A \rangle \langle D \rangle$
- (3)  $\langle A \rangle \langle B \rangle \rightarrow 1 \langle A \rangle \langle E \rangle$
- (4)  $\langle D \rangle \langle C \rangle \rightarrow \langle B \rangle 0 \langle C \rangle$
- (5)  $\langle E \rangle \langle C \rangle \rightarrow \langle B \rangle 1 \langle C \rangle$
- (6)  $\langle D \rangle 0 \rightarrow 0 \langle D \rangle$
- (7)  $\langle D \rangle 1 \rightarrow 1 \langle D \rangle$
- (8)  $\langle E \rangle 0 \rightarrow 0 \langle E \rangle$
- (9)  $\langle E \rangle 1 \rightarrow 1 \langle E \rangle$
- (10)  $0 \langle B \rangle \rightarrow \langle B \rangle 0$
- (11)  $1 \langle B \rangle \rightarrow \langle B \rangle 1$
- (12)  $\langle A \rangle \langle B \rangle \rightarrow \epsilon$
- (13)  $\langle C \rangle \rightarrow \epsilon$

- (a) What type of grammar is this (context-free or phrase structure)? Justify your answer.
- (b) What language does this grammar generate? (Hint: Rules (12) and (13) show you that the word before the last non-terminals are swapped out can contain only non-terminals  $\langle A \rangle, \langle B \rangle, \langle C \rangle$ . Figure out how the other rules combine to give you words consisting of the terminals and  $\langle A \rangle, \langle B \rangle, \langle C \rangle$ .)

**Solution:** (a) Production rule (2) has two non-terminals on the left-hand side, so this grammar is clearly a phrase structure grammar as context-free grammars have production rules that allow only one non-terminal to be swapped for something else.

(b) Rules (1), (2), and (4) yield  $0 \langle A \rangle \langle B \rangle 0 \langle C \rangle$ , whereas rules (1), (3), and (5) yield  $1 \langle A \rangle \langle B \rangle 1 \langle C \rangle$ . We can get to  $01 \langle A \rangle \langle B \rangle 01 \langle C \rangle$  from  $0 \langle A \rangle \langle B \rangle 0 \langle C \rangle$  by applying rules (3), (8), (5), and (10). Similarly, we can get  $w \langle A \rangle \langle B \rangle w \langle C \rangle$

for any word  $w \in \{0, 1\}^*$ . Rules (6) to (11) are there to rearrange non-terminals so that rules (2) to (5) can be applied. Altogether, we generate the language  $L = \{ww \mid w \in \{0, 1\}^*\}$ . Note that the language  $L$  CANNOT be generated by a context-free grammar as a context-free grammar could NOT keep track of the duplicate  $w$  by replacing only one non-terminal at a time. We have thus constructed an example of a language that is generated by a phrase structure grammar and cannot be generated by a context-free grammar thus showing that the former category is more general than the latter.