Monash University
Faculty of Information Technology
2^{nd} Semester 2018

FIT2014
Tutorial 4
Context-Free Grammars and the Pumping Lemma

Attempt all questions before the tutorial.

ASSESSED PREPARATION: Question 5.
You must present a serious attempt at this entire question to your tutor at the start of your tutorial.

1.
Consider the following four statements, which we call W, X, Y and Z.

W: Every context-free language is infinite.
X: Every context-free language is finite.
Y: There exists an infinite context-free language.
Z: There exists a finite context-free language.

(a)

(i). Which of X, Y, Z is the logical negation of W?

(ii). Which of W, X, Y, Z is true?

(b) Define the predicates **CFL** and **Infinite** to have the following meanings.

CFL(L)	L is context-free
Infinite(L)	L is infinite

The universe of discourse is *all languages.*

(i). Write statements W, X, Y, Z in predicate logic, using these two predicates.

(ii). Use the properties of quantifiers, and laws of propositional logic as needed, to derive an existential statement equivalent to $\neg W$.

(iii). Which of X, Y, Z is equivalent to your answer to (ii)?

2. Let LOL be the language generated by the following Context-Free Grammar.

$$
\begin{aligned}
S &\rightarrow P && (1)\\
P &\rightarrow PP && (2)\\
P &\rightarrow \texttt{ha}Q && (3)\\
Q &\rightarrow Q\texttt{a} && (4)\\
Q &\rightarrow \varepsilon && (5)
\end{aligned}
$$

(a) **Warm-up:**

(i). What is the shortest string in LOL? Give a derivation for it.

(ii). Is the above grammar regular?

(iii). Is LOL a regular language? If so, give a regular expression for it. If not, prove it.

(iv). Is the above grammar in Chomsky Normal Form? If so, why; if not, state why not.

(b) Here is an attempted proof that LOL is infinite. Comment on what is correct and incorrect in this proof.

1. Let x be some string in LOL.
2. The string x must have a derivation, using the above grammar.
3. The only rule in the grammar that does not have a nonterminal symbol (or empty string) on its right-hand side is the last one, (5).
4. So any derivation of x must finish with an application of (5).
5. Therefore the string just before x, in the derivation of x, must have an Q.
6. Therefore the derivation of x has the form

   $$S \Longrightarrow \cdots \Longrightarrow x_{\text{left}} Q x_{\text{right}} \overset{(5)}{\Longrightarrow} x_{\text{left}} x_{\text{right}} \quad = \ x,$$

   where x_{left} denotes the portion of x before Q, and x_{right} denotes the portion of x after Q.
7. So, instead of applying rule (5) to this last occurrence of Q, we could apply rule (4) followed by rule (5), giving

   $$S \Longrightarrow \cdots \Longrightarrow x_{\text{left}} Q x_{\text{right}} \overset{(4)}{\Longrightarrow} x_{\text{left}} Q \mathtt{a} x_{\text{right}} \overset{(5)}{\Longrightarrow} x_{\text{left}} \mathtt{a} x_{\text{right}}.$$

   This new string is also in LOL and is one letter longer than x. Call it y.
8. We can apply this argument again to y, to get a string in LOL that is one letter longer than y, and then we can get strings that are even longer than that, and so on, indefinitely.
9. So we can generate infinitely many strings. Therefore LOL is infinite.

(c) Prove by induction that, for all $n \geq 2$, the language LOL contains a string of length at least n.

(d) Here is an attempted *proof by contradiction* that LOL is infinite. It uses, in the middle, the proof given in (c). Comment on what is good and bad about this proof.

1. Assume, by way of contradiction, that LOL is finite.
2. $\langle$ *Insert here a correct proof, based on (c), that LOL has arbitrarily long strings.* $\rangle$
3. This contradicts our assumption that LOL is finite.
4. Therefore LOL is infinite.

(e) Construct a correct proof by contradiction that LOL is infinite, using the following approach:

1. Start by assuming LOL is finite, as in (c) step 1.
2. Show that LOL contains a nonempty string. (No need to use the assumed finiteness of LOL just yet.)
3. Among all the nonempty strings in LOL, choose x carefully ... How should you choose it?
4. Show how to construct a string in LOL that is longer than x.
5. ......
6. Therefore LOL is infinite.

3. This question is a sequel to Tute 3, Q4.

(a) Review Tute 3, Q4, reflecting on the discussion of it in your tutorial, and correcting your attempt at the question.

(b) Use the Pumping Lemma for CFLs to prove that the language $\{\mathbf{a}^{n^2} : n \in \mathbb{N}\}$ is not context-free.

(c) Hence prove that the language of binary string representations of adjacency matrices of graphs is not context-free.

You may assume that the intersection of a context-free language with a regular language is context-free. (Challenge: prove this.) Note that, unlike regular languages, the class of context-free languages is not closed under intersection. (Can you prove this?)

4. Recall the **Pumping Games** of Tute 3. An expansion pack has now been released, called **Double Pumping Games**.

You can play a Double Pumping Game for *any* language. The players are Noni and Con. First, the players are given a language L (which may or may not be context-free) and a number k. Noni moves first, then Con moves, then Noni moves. The rules for their moves are as follows.

- Noni first chooses a string $w \in L$ with $|w| > 2^{k-1}$.
- Then Con divides w up into substrings u, v, x, y, z such that $vy \neq \varepsilon$ and $|vxy| \leq 2^k$.
- Then Noni chooses a non-negative integer i.

The result of the game is that:

- if $uv^ixy^iz \in L$, then Con wins;
- if $uv^ixy^iz \notin L$, then Noni wins.

(a) Play the game for (i) PALINDROMES, and (ii) $\{\mathtt{a}^n\mathtt{b}^n\mathtt{a}^n : n \geq 0\}$.
(b) Using quantifiers, write down the assertion that Noni has a winning strategy.
(c) Using quantifiers, write down the assertion that Con has a winning strategy.
(d) What does the Pumping Lemma for CFLs tell us about circumstances in which winning strategies exist?

5. In Australian Rules Football, a team gains *points* by kicking *goals* and *behinds*. A goal is worth 6 points, and a behind is worth one point. The winner is the team with the greatest number of points at the end of the game.

A team's score is given as a triple of numbers (g, b, p), where g is the number of goals, b is the number of behinds, and p is the number of points. These numbers must be nonnegative and satisfy $6g + b = p$. For example, $(1, 2, 8)$ is a valid score, since $6 \times 1 + 2 = 8$.

Let FootyScore be the language of valid football scores (g, b, p), where each number is represented in unary[1], and the alphabet consists of 1, the two parentheses, and the comma. For example,

$$\texttt{(1,11,11111111)} \in \text{FootyScore}.$$

In this language, there is no upper limit on the scores.

(a) Is the language FootyScore regular? Prove or disprove.

(b) Is the language FootyScore context-free? Prove or disprove.

6. Let G be the following CFG.

$$\begin{align} S &\rightarrow AB \tag{6} \\ A &\rightarrow AC \tag{7} \\ B &\rightarrow CD \tag{8} \\ A &\rightarrow \texttt{a} \tag{9} \\ C &\rightarrow \texttt{b} \tag{10} \\ D &\rightarrow \texttt{a} \tag{11} \end{align}$$

Use the Cocke-Younger-Kasami (CYK) algorithm to determine whether or not the string `abbba` is generated by G.

7.
(a) Convert the CFG in Q2 into a Chomsky Normal Form grammar for LOL.
(b) Using this Chomsky Normal Form grammar, apply the CYK algorithm to the string `hahaa`.

8. Prove by induction that the CYK algorithm correctly determines whether or not a given string x is generated by a given context-free grammar G.

More specifically:

1. Prove by induction on n that, for every substring y of x, where $|y| = n$, the algorithm correctly finds all nonterminals in G that can generate y.

2. Explain briefly how, if we know all the nonterminals that can generate x, we can then determine whether x can be generated by G.

The idea of the proof is given in Lecture 13, slide 8. Your task is to avoid the gap in that sketch proof ("Continue, in this way, ...") and produce a correct proof by induction.

[1] In *unary* representation, a number n is represented as a string of n 1s. For example, in unary, the number 5 is represented by 11111, and 0 is represented by the empty string.