Introducing Context-Free Grammars

Question 1. Describe the set of strings that are described by the regular expression $a(a^* \cup b^*)b$.

Question 2. Give a context free grammar that generates the same set of strings.

Question 3. Show how the grammar generates the string *aaab*.

Question 4. What do the word "derivation" refer to in this context?

Formally, a **Context Free Grammar** G is a quadruple (V, Σ, R, S), where

- V is an alphabet.
- Σ (the set of **terminals**) is a subset of V
- R (the set of **rules**) is a finite subset of $(V - \Sigma) \times V^*$, and
- S (the **start symbol** is an element of $V - \Sigma$

Question 5. Give a Context Free Grammar for $\{a^n b^n \mid n \geq 0\}$

Question 6. Give five strings in the language described by the following Context Free Grammar with start symbol S.

- $S \rightarrow PVP$
- $P \rightarrow N$
- $P \rightarrow AP$
- $A \rightarrow$ big
- $A \rightarrow$ green
- $N \rightarrow$ cheese
- $N \rightarrow$ Jim
- $V \rightarrow$ ate

Question 7. Given the following CFG, give a derivation of the string `( id * id + id)`. Give also a parse tree for the same string. The start symbol is E.

- $E \rightarrow E + T$
- $E \rightarrow T$
- $T \rightarrow T * F$
- $T \rightarrow F$
- $F \rightarrow (E)$
- $F \rightarrow$ id

Question 8. In a parse tree, what are the leaf nodes? The interior nodes? What restriction(s) are there on interior nodes' children? What is the root node?

Question 9. Consider the grammar $S \to \varepsilon \mid SS \mid (S)$. Give two derivations of the string ()(())

Question 10. What does it mean for a grammar to be ambiguous? Is that a property of the grammar or of the language?

Question 11. Give an *unambiguous* grammar for the same language. Give the only derivation for the string ()(()) in via that grammar.

Question 12. Does every language that can be represented by a CFG have an unambiguous grammar?

Question 13. What do we mean when we say a grammar is LL(1)? Specifically, what does each 'L' stand for, and what does the one represent?

Question 14. Outline the proof that every regular language can be represented by a context free grammar.

Question 15. This question is from Lewis/Papadimitriou 3.1.3(a). Give a CFG for $\{wcw^R : w \in \{a, b\}^*\}$

Introduction to Push Down Automata

Question 16. Give a push-down automata that recognizes the language $a^n b^n$

Question 17. Give a push-down automata that recognizes the language of $w \in \{a,b\}^*$ with an equal number of a and b.

A **pushdown automata** is $M = (K, \Sigma, \Gamma, \Delta, s, F)$ where

- K is a finite set of states
- Σ is an alphabet (**input symbols**)
- Γ is an alphabet (the **stack symbols**)
- $s \in K$ is the initial state
- $F \subseteq K$ is the set of final states
- Δ is the **transition relation**, a finite subset of $(K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*) \times (K \times \Gamma^*)$

Question 18. What constitutes a *deterministic* PDA? Which of our earlier PDA(s) was deterministic?

Question 19. Give a push-down automata that recognizes the language $\{a^i b^j c^k \; : \; i, j, k \geq 0$ and $i == j$ OR $i == k\}$

Question 20. Give a push-down automata that recognizes the language $\{ww^R \; : \; w \in \{a, b\}^*\}$

The equivalence of PDAs and CFGs

This lecture is going to proceed in two parts: first, we will show that every context-free language is accepted by *some* PDA. We will then show that anything you can write a PDA for is a context free language.

Every CFL is accepted by some PDA

We will start with an example that will illustrate the proof.

Suppose I want a PDA for $\{wcw^R \; : \; w \in \{a, b\}^*\}$

We already saw a grammar for this[1]. The grammar was $S \to aSa \mid bSb \mid c$

Let's define a PDA that accepts the same language. Rather than create it from scratch, we are going to create it *from the grammar*. From this, we can derive some general rules for how to turn *any* context-free grammar into an equivalent PDA.

[1] We also saw a PDA for something very similar, but nevermind that right now!

Let's walk through this PDA accepting the string *abbcbba*

General Method

Given a grammar $G = (V, \Sigma, R, S)$, create a PDA as follows:

Every language accepted by a PDA is Context-Free

Given a PDA, make a CFG that generates all the strings that the PDA accepts.

To simplify the task, we will assume:

1. There is only one accept state
2. The PDA empties the stack before accepting
3. Each transition pushes XOR pops, not both (and exactly one!)

Why is this assumption valid?

We will now create a series of rules; for each pair of states p, q in the PDA, we create A_{pq}, which will generate all strings that take our PDA from state p to state q with an empty stack before and after[2].

[2] Or, if the stack wasn't empty, can do so. That is, if the stack was non-empty, then this set of strings can take us from p to q without touching anything that is on the stack at the start.

We need to prove two statements to demonstrate correctness.

First, if rule A_{pq} generates x, then x can bring the PDA from state p with an empty stack to state q with an empty stack.

Then, if string x can do so for the PDA, then A_{pq} can generate x.

Cleaner Grammar and Non-Context Free Languages

Question 21. Consider the following grammar. In a sentence or two of English, what does this grammar produce? Can you give a simpler version?

- $S \to AB$
- $A \to aA \mid a$
- $B \to AB$

Question 22. How do we test if a variable derives some terminal string?

Question 23. Can you give an algorithm to eliminate variables that derive nothing?

Example:

- $S \to AB \mid C$
- $A \to aA \mid a$
- $B \to bB$
- $C \to c$

Question 24. When simplifying a grammar, can we avoid ε productions?

Question 25. What are all the nullable symbols in the following grammar?

- $S \to AB$
- $A \to aA|\varepsilon$
- $B \to bB|A$

Question 26. How do we eliminate ε productions?

The idea is to turn each rule into a *family* of productions.

Example:

- $S \to ABC$
- $A \to aA \mid \varepsilon$
- $B \to bB \mid \varepsilon$
- $C \to \varepsilon$

Question 27. How do we eliminate unit productions?

A context-free grammar is in **Chomsky Normal Form** if every production is one of these forms:

- $A \to BC$ where rhs is *two variables*
- $A \to a$ where rhs has *just one terminal*

Pumping Lemma for Context Free Languages

If A is a CFL, there is a value p where if s is any string in A of length at least P, s can be partitioned into five pieces $s = uvxyz$ such that:

1. $\forall i \geq 0 \;\; uv^ixy^iz \in A$
2. $|vy| > 0$
3. $|vxy| \leq p$

Question 28. Prove that the following language is not context free: $L_1 = \{a^nb^nc^n \mid n \geq 0\}$

Question 29. Prove that the following language is not context free: $L_2 = \{a^ib^jc^k \mid 0 \leq i \leq j \leq k\}$

Question 30. Prove that the following language is not context free: $L_3 = \{ww \mid w \in \{a,b\}^*\}$

Closures

We saw at the end of unit one that regular languages have *closure* properties; the first few questions are to remind you about these, and then we will apply this concept to context-free languages.

Question 31. Show that if L_1 and L_2 are regular, so is $L_3 = L_1 \cup L_2$

Question 32. Show that if L_1 and L_2 are regular, so is $L_4 = \{w_1 w_2 \ : \ w_1 \in L_1, w_2 \in L_2\}$

Question 33. Show that if L_1 is regular, so is $L_5 = L_1^*$

Question 34. Show that if L_1 is regular, so is $L_6 = \overline{L_1}$

Question 35. Show that if L_1 and L_2 are regular, so is $L_7 = L_1 \cap L_2$

Question 36. Show that if L_1 and L_2 are CFLs, then so is $L_3 = L_1 \cup L_2$

Question 37. Show that if L_1 and L_2 are CFLs, then so is $L_4 = L_1 L_2$.

The proof looks a lot like Union:

- $G_1 = (V_1, \Sigma_1, R_1, S_1)$
- $G_2 = (V_2, \Sigma_2, R_2, S_2)$
- $G_4 = (V_1 \cup V_2 \cup S, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{ \qquad\qquad\qquad\qquad \}, S)$

Question 38. Show that if L_1 is a CFL, then so is $L_5 = L_1^*$.

Proof looks a lot like Union:

- $G_1 = (V_1, \Sigma_1, R_1, S_1)$
- $G_5 = (V_1 \cup V_2 \cup S, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{ \qquad \}, S)$

Question 39. Show that the intersection of a CFL and a Regular Language is a CFL.

- PDA $M_1 = (K_1, \Sigma, \Gamma_1, \Delta_1, s_1, F_1)$
- DFA $M_2 = (K_2, \Sigma, \delta, s_2, F_2)$
- Result PDA is $M = (K, \Sigma, \Gamma, \Delta, s, F)$ where:
  - $K =$
  - $\Gamma =$
  - $s =$
  - $F =$
  - What is Δ?

Question 40. Show that if L_1 and L_2 are CFLs, is $L_7 = L_1 \cap L_2$ CFL?

Question 41. Show that if L_1 is a CFL, is $L_8 = \overline{L_1}$?