CS 121 Section 4

Harvard University

October 3, 2013

1 Concept Review

1.1 Context Free Grammars

A context-free grammar G is a four-tuple, $G = (V, \Sigma, R, S)$, defined as follows:

- V is the set of variables
- Σ is the set of terminals, and so must be disjoint from V
- R is a finite set of rules, where each rule consists of a variable transforming into a string of variables and terminals
- S is the start symbol, and is an element of V

The idea is that the grammar consists of all strings over Σ^*, our terminal symbols, which we can get by starting with S and following the rules. The process of moving from S to a final string of terminals is known as a *derivation*.

1.2 Derivations

If x, y, and z are strings of variables and terminals and $A \to y$ is a rule of the grammar, then we can write $xAz \Rightarrow xyz$ and say xAz yields xyz in one step.

Extending that idea, if x_1 and x_n are strings of variables and terminals then we can say $x_1 \overset{*}{\Rightarrow} x_n$, or x_1 derives x_n, if we can get from x_1 to x_n by following 0 or more rules in succession. More formally, $x_1 \overset{*}{\Rightarrow} x_n$ if $x_1 = x_n$ or there is a sequence x_1, $x_2 \dots x_n$ such that for all i, $x_i \Rightarrow x_{i+1}$. In practice, we often aren't very careful about distinguishing between 'derive' and 'yield', and it is ok to use them interchangeably.

The language of a grammar G is then defined as $L(G) = \{w \in \Sigma^* : S \overset{*}{\Rightarrow} w\}$

A derivation for a string w in a grammar G is any series of strings $S \Rightarrow x_1 \cdots \Rightarrow w$ that show how to get w from the rules of the grammar. A leftmost derivation for a string is a derivation where in each step, the leftmost variable in the string is substituted. A grammar is said to be ambiguous if there exists a string in the language of the grammar which has two different leftmost derivations. We often visualize derivations using parse trees.

2 Exercises

Exercise 2.1. *Show that the following languages are context-free:*

1. $L = \{a^i b^j c^k : i, j, k \in \mathbb{N},$ *and if* $i = 1$ *then* $j \geq k\}$ *over* $\Sigma = \{a, b, c\}$;
2. $L = \{w : w = w^R\}$;

1. $S \to aJ \mid aaABC \mid BC$
   $J \to \varepsilon \mid bJ \mid bJc$
   $A \to aA \mid \varepsilon$
   $B \to bB \mid \varepsilon$
   $C \to cC \mid \varepsilon$
2. $S \to a \mid b \mid aSa \mid bSb \mid \varepsilon$

Exercise 2.2. *Let* $G = (V, \Sigma, R, S)$ *be the following grammar.*

$$\begin{aligned} S &\to AS \mid \varepsilon \\ A &\to A1 \mid 0A1 \mid \varepsilon \\ \Sigma &= \{0, 1\} \\ V &= \{A, S\} \end{aligned}$$

1. *Show that* G *is ambiguous.*

   For this we can generate the string 011 *with two different derivations (both replacing leftmost variable first):*
   $S \to AS \to 0A1S \to 01S \to 01AS \to 01A1S \to 011S \to 011$ *or*
   $S \to AS \to 0A1S \to 0A11S \to 011S \to 011$

2. *Give a new grammar that generates the same language as* G *but is unambiguous. Justify briefly why your grammar generates the same language and why it is unambiguous.*

   This language is a little hard to describe, it's like $(0^m 1^n)^*$, *with* $m \leq n$. *New grammar:*

$$\begin{aligned} S &\to AS \mid 1S \mid \varepsilon \\ A &\to 01 \mid 0A1 \end{aligned}$$

   Quick explanation: If a string has more 1s than 0s following every "clump" of 0s, then there are two cases: If w *starts with a* 1, *we can write it as* $1w_1$, *with* $w_1 \in L$ *and use the rule* $S \to 1S$. *If* w *starts with a* 0, *we can write it as* $0^m 1^m w_2$ *with* $w_2 \in L$ *and use the rule* $S \to AS$. *Our grammar covers either case and because the cases are disjoint it should be unambiguous.*

Exercise 2.3. *Consider the following grammar:*

$$\begin{aligned} S &\to \langle SUBJECT\rangle\langle VERB\rangle\langle OBJECT\rangle\langle MODIFIER\rangle \\ \langle SUBJECT\rangle &\to \textit{The woman} \\ \langle VERB\rangle &\to \textit{hit} \\ \langle OBJECT\rangle &\to \textit{the man } \langle MODIFIER\rangle \\ \langle MODIFIER\rangle &\to \textit{with an umbrella} \mid \varepsilon \end{aligned}$$

1. *Show that this grammar is ambiguous.*

   For example:
   (The woman hit (the man with the umbrella)), or
   (The woman hit (the man) with the umbrella)

Exercise 2.4. *Show that every regular language has an unambiguous context-free grammar.*

Proof. (Sketch) Let L be a regular language. We will construct a CFG for L from the DFA $M = (\Sigma, Q, q_0, F, \delta)$ that accepts L. We let Q be the set of variables of the CFG, introduce the rule $q \longrightarrow \sigma q'$ for every $q \in Q$, $\sigma \in \Sigma$, $q' = \delta(q, \sigma)$, and introduce the rule $q \longrightarrow \epsilon$ for every $q \in F$. We let q_0 be the starting variable. This is a CFG for L, since for every $x \in L$ the transition function δ takes q_0 to some state in F, thus we have $q_0 \overset{*}{\Longrightarrow} x$, and vice versa. Furthermore, the CFG is unambiguous since for every x where $q_0 \overset{*}{\Rightarrow} x$, only one rule can be applied at each step (by an easy induction). □

Exercise 2.5. *Given an arbitrary context free grammar G, provide a general procedure to determine if $L(G)$ is empty.*

Proof. (Sketch) Call a variable *generating* if it yields at least some string of terminal symbols. We build the set of generating variables iteratively, and accept that $L(G)$ is empty if the starting variable is not in the set.

We build the set Y of generating variable as follows. Intially set $Y = \emptyset$. Scan each rule $A \longrightarrow \ldots$ where $A \notin Y$, and add A to Y if all variables in the RHS are in Y. Stop when no more variable can be added. Clearly the procedure stops after finitely many steps since each iteration adds another variable to Y.

To argue correctness we need to show that Y is exactly the set of generating variables. Clearly every variable in Y is generating, by induction on the order they are added to Y. Also, every generating variable A is in Y, by induction on the parse tree of any terminal string that A yields.

□