CENG 280 - Chapter 3: Context Free Grammer

Burak Metehan Tunçel - May, June 2022

1 Pushdown Automata and Context Free Grammars

Theorem 1

The class of languages accepted by pushdown automata is exactly the class of context-free languages.

Proof: We break this proof into two parts.

Lemma 1

Each context-free language is accepted by some pushdown automaton.

Proof. Let $G = (V, \Sigma, R, S)$ be a context-free grammar; we must construct a pushdown automaton M such that $L(M) = L(G)$. The machine we construct has only two states, p and q, and remains permanently in state q after its first move. Also, M uses V, the set of terminals and nonterminals, as its stack alphabet. We let $M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$, where Δ contains the following transitions:

1. $((p, e, e),\ (q, S))$
2. $((q, e, A),\ (q, x))$ for each rule $A \to x$ in R.
3. $((q, a, a),\ (q, e))$ for each $a \in \Sigma$.

The pushdown automaton M begins by pushing S, the start symbol of G, on its initially empty pushdown store, and entering state q (transition 1). On each subsequent step, it either replaces the topmost symbol A on the stack provided that it is a nonterminal, by the right-hand side x of some rule $A \to x$ in R (transitions of type 2), or pops the topmost symbol from the stack, provided that it is a terminal symbol that matches the next input symbol (transitions of type 3).

The transitions of M are designed so that the pushdown store during an accepting computation mimics a leftmost derivation of the input string.

To continue the proof of the Lemma, in order to establish that $L(M) = L(G)$, we prove the following claim

Claim: Let $w \in \Sigma^*$ and $\alpha \in (V - \Sigma)V^* \cup \{e\}$. Then $S \overset{L}{\Rightarrow}{}^* w\alpha$ iff $(q, w, S) \vdash_M^* (q, e, \alpha)$.

This claim will suffice to establish Lemma 1, since it will follow (by taking $\alpha = e$) that $S \overset{L}{\Rightarrow}{}^* w$ if and only if $(q, e, S) \vdash_M^* (q, e, e)$ in other words, $w \in L(G)$ if and only if $w \in L(M)$.

(*Only if*) Suppose that $S \overset{L}{\Rightarrow}{}^* w\alpha$, where $w \in \Sigma^*$, and $\alpha \in (V - \Sigma)V^* \cup \{e\}$. We shall show by induction on the length of the leftmost derivation of w from S that $(q, w, S) \vdash_M^* (q, e, \alpha)$.
Basis Step. If the derivation is of length 0, then $w = e$, and $\alpha = S$, and hence indeed $(q, w, S) \vdash_M^* (q, e, \alpha)$.
Induction Hypothesis. Assume that if $S \overset{L}{\Rightarrow}{}^* w\alpha$ by a derivation of length n or less, $n \geq 0$, then $(q, w, S) \vdash_M^* (q, e, \alpha)$.
Induction Step. Let

$$S = u_0 \overset{L}{\Rightarrow} u_1 \overset{L}{\Rightarrow} \cdots \overset{L}{\Rightarrow} u_n \overset{L}{\Rightarrow} u_{n+1} = w\alpha$$

be a leftmost derivation of $w\alpha$ from S. Let A be the leftmost nonterminal of u_n. Then $u_n = xA\beta$, and $u_{n+1} = x\gamma\beta$ where $x \in \Sigma^*$, β, $\gamma \in V^*$, and $A \to \gamma$ is a rule in R. Since there is a leftmost derivation of length n of $u_n = xA\beta$ from S, by the induction hypothesis

$$(q, x, S) \vdash_M^* (q, e, A\beta) \tag{1}$$

Since $A \to \gamma$ is a rule in R,

$$(q, e, A\beta) \vdash (q, e, \gamma\beta) \tag{2}$$

by a transition of type 2.

Now notice that u_{n+1} is $w\alpha$, but it is also $x\gamma\beta$. Hence, there is a string $y \in \Sigma^*$ such that $w = xy$ and $y\alpha = \gamma\beta$. Thus, we can rewrite (1) and (2) above as

$$(q, w, S) \vdash_M^* (q, y, \gamma\beta) \tag{3}$$

However, since $y\alpha = \gamma\beta$

$$(q, y, \gamma\beta) \vdash_M^* (q, e, \alpha) \tag{4}$$

by a sequence of $|y|$ transitions of type 3. Combining (3) and (4) completes the induction step.

(*If*) Now suppose that $(q, w, S) \vdash_M^* (q, e, \alpha)$ with $w \in \Sigma^*$ and $\alpha \in (V - \Sigma)V^* \cup \{e\}$; we show that $S \overset{L}{\Rightarrow} w\alpha$. Again, the proof is by induction, but this time on the number of transitions of type 2 in the computation by M.

Basis Step. Since the first move in any computation is by a transition of type 2, if $(q, w, S) \vdash_M^* (q, e, \alpha)$ with no type-2 transitions, then $w = e$ and $\alpha = S$, and the result is true.

Induction Hypothesis. If $(q, w, S) \vdash_M^* (q, e, \alpha)$ by a computation with n type 2 steps or fewer, $n \geq 0$, then $S \overset{L}{\Rightarrow} w\alpha$.

Induction Step. Suppose that $(q, w, S) \vdash_M^* (q, e, \alpha)$ in $n+1$ type-2 transitions, and consider the next-to-Iast such transition, say,

$$(q, w, S) \vdash_M^* (q, y, A\beta) \vdash (q, y, \gamma\beta) \vdash_M^* (q, e, \alpha)$$

where $w = xy$ for some $x, y \in \Sigma^*$, and $A \to \gamma$, is a rule of the grammar. By the induction hypothesis we have that $S \overset{L}{\Rightarrow} xA\beta$, and thus $S \overset{L}{\Rightarrow} x\gamma\beta$. Since however $(q, y, \gamma\beta) \vdash_M^* (q, e, \alpha)$, presumably by transitions of type 3, it follows that $y\alpha = \gamma\beta$, and thus $S \overset{L}{\Rightarrow} xy\alpha = w\alpha$. This completes the proof of Lemma 1, and with it half the proof of Theorem 2. □

We now turn to the proof of the other half of Theorem 2.

Lemma 2

If a language is accepted by a pushdown automaton, it is a context-free language.

Proof. It will be helpful to restrict somewhat the pushdown automata under consideration. Call a pushdown automaton **simple** if the following is true:

> Whenever $((q,\ a,\ \beta),\ (p,\ \gamma))$ is a transition of the pushdown automaton and q is not the start state, then $\beta \in \Gamma$, and $|\gamma| \leq 2$.

In other words, the machine always consults its topmost stack symbol (and no symbols below it), and replaces it either with e, or with a single stack symbol, or with two stack symbols. Now it is easy to see that no interesting pushdown automaton can have only transitions of this kind, because then it would not be able to operate when the stack is empty (for example, it would not be able to start the computation, since in the beginning the stack is empty). This is why we do not restrict transitions from the start state.

We claim that if a language is accepted by an unrestricted pushdown automaton, then it is accepted by a simple pushdown automaton. To see this, let $M = (K,\ \Sigma,\ \Gamma,\ \Delta,\ s,\ F)$ be any pushdown automaton; we shall construct a simple pushdown automaton $M' = (K',\ \Sigma,\ \Gamma \cup \{Z\},\ \Delta',\ s',\ \{f\})$ that also accepts $L(M)$; here s' and f' are new states not in K, and Z is a new stack symbol, the stack *bottom symbol*, also not in Γ. We first add to Δ the transition $((s',\ e,\ e),\ (s,\ Z))$; this transition starts the computation by placing the stack bottom symbol in the bottom of the stack, where it will remain throughout the computation. No rule of Δ' will ever push a Z in the stack - except to replace it at the bottom of the stack. We also add to Δ' the transitions $((f,\ e,\ Z),\ (f',\ e))$ for each $f \in F$. These transitions end the computation by removing Z from the bottom of the stack and accepting the input seen so far.

Initially, Δ' consists of the start and final transitions described above, and all transitions of Δ. We shall next replace all transitions in Δ' that violate the simplicity condition by equivalent transitions that satisfy the simplicity condition. We shall do this in three stages:

- First we shall replace transitions with $|\beta| \geq 2$.
- Then we shall get rid of transitions with $|\gamma| > 2$, without introducing any transitions with $|\beta| \geq 2$.
- Finally, we shall get rid of transitions with $\beta = e$, without introducing any transitions with $|\beta| \geq 2$ or $|\gamma| > 2$.

□