Inf2A: CFGs and PDAs Describe the same Class of Languages.

Stuart Anderson

School of Informatics
University of Edinburgh

October 16, 2009

Outline

Definition of A Pushdown Automaton

This is the formal definition of a Pushdown Automaton, $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, F)$:

- Q is the finite set of states.
- Σ – a finite set, the input alphabet.
- Γ – a finite set, the stack alphabet.
- $\delta \subseteq Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \cup \{\lambda\} \times Q \times \Gamma^*$ is the transition relation, we write λ for the empty string in Γ^*.
- $q_0 \in Q$ is the initial state.
- $\perp \in \Gamma$ is the initial stack symbol.
- $F \subseteq Q$ is the set of final states.

Configurations

- A *configuration* of a PDA M is a triple (q, x, β) where $q \in Q, x \in \Sigma^*, \beta \in \Gamma^*$ where q is the current state, x is the remaining input to be read, and β is the current stack.
- The *next configuration relation* for M is written $\rightarrow_M$ where:
  1. $(q, ay, A\beta) \rightarrow_M (q', y, \gamma\beta)$ iff $((q, a, A), (q', \gamma)) \in \delta$
  2. $(q, y, A\beta) \rightarrow_M (q', y, \gamma\beta)$ iff $((q, \varepsilon, A), (q', \gamma)) \in \delta$
  3. $(q, ay, \beta) \rightarrow_M (q', y, \gamma\beta)$ iff $((q, a, \lambda), (q', \gamma)) \in \delta$
  4. $(q, y, \beta) \rightarrow_M (q', y, \gamma\beta)$ iff $((q, \varepsilon, \lambda), (q', \gamma)) \in \delta$

Computations and Acceptance

- We define *computation* for M as a relation between configurations as follows:
  1. $C \to_M^0 D$ iff $C = D$
  2. $C \to_M^{n+1} D$ iff $C \to_M C'$ and $C' \to_M^n D$
  3. $C \to_M^* D$ iff, for some i, $C \to_M^i D$
- We have two possible acceptance criteria for PDAs. In specifying a PDA we ahould specify the intended criterion:
  - Acceptance by final state: *M accepts x* iff $(q_0, x, \perp) \to_M^* (f, \varepsilon, \gamma)$ and $f \in F$.
  - Acceptance by empty stack: *M accepts x* iff $(q_0, x, \perp) \to_M^* (q, \varepsilon, \lambda)$.

Constructing a PDA from a CFG

- Given a CFG $G = (N, \Sigma, P, S)$ in Griebach normal form
- Construct the PDA $M = (\{q_0\}, \Sigma, N, \delta, q_0, S, \emptyset)$ which accepts on empty stack. The PDA has transitions:
  1. For each production $A \rightarrow cB_1 \ldots B_k$ add the transition $((q_0, c, A), (q_0, B_1 \ldots B_k))$
- $L(M) = L(G)$, proof - for every left derivation in G, there is a corresponding computation in M
- How do we do the proof?

CFG to PDA: Example

Consider the grammar:

$$G = (\{S\}, \{[,]\}, \{S \rightarrow [BS \mid [B \mid [SB \mid [SBS,\ B \rightarrow]\}, S)$$

- $S \rightarrow [BS$ – add the transition $((q_0, [, S), (q_0, BS))$
- $S \rightarrow [B$ – add the transition $((q_0, [, S), (q_0, B))$
- $S \rightarrow [SB$ – add the transition $((q_0, [, S), (q_0, SB))$
- $S \rightarrow [SBS$ – add the transition $((q_0, [, S), (q_0, SBS))$
- $B \rightarrow]$ – add the transition $((q_0,], B), (q_0, \lambda))$

Constructing a CFG from a PDA

- See Kozen chapter 25.
- We can work either with the definition of halting as reaching a final state or that a PDA halts when it pops the initial symbol.
- Demonstrate every PDA can be converted to an equivalent 1-state PDA.
- Use the previous construction in reverse.

The Pumping Lemma for CFGs

- For every context-free language L there is a constant $k \geq 0$ such that for every $z \in L$ of length at least k, we can find strings u, v, w, x, y such that:
  1. $z = uvwxy$
  2. $vx \neq \varepsilon$
  3. $\mid vwx \mid \leq k$
  4. for all $i \geq 0$: $uv^iwx^iy \in L$
- An hence $\{a^nb^nc^n \mid n \geq 0\}$ is not a CFL.

The Pumping Lemma for CFGs – Example

- Show $L = \{a^n b^n c^n \mid n \geq 0\}$ is not a context-free language.
- The proof is by contradiction, so assume L is a CFL. So for some k the pumping lemma holds.
- Consider the string $s = a^k b^k c^k$
- From the assumption and the pumping lemma for CFGs, we know we can find $uvwxy = s$ with $vx \neq \varepsilon$ and $\mid vwx \mid \leq k$ such that for all $i \geq 0 : uv^i wx^i y \in L$
- Because of the restriction on the length of vwx any choice of $uvwxy$ means vwx is either (i) a member of $\{a, b\}^*$ or (ii) a member of $\{b, c\}^*$
- If we consider $uv^i wx^i y$ with $i = 2$, then in case (i) there are certainly too few c symbols and in case (ii) there are certainly too few a symbols. In both cases we have a contradiction.

Closure properties for CFLs

- If L_1 and L_2 are CFLs and R is a regular language then:
- $L_1 L_2$ is a CFL
- $L_1 \cup L_2$ is a CFL
- L_1^* is a CFL
- $L_1 \cap R$ is a CFL
- $L_1 \cap L_2$ is *not* a CFL.