6 Pushdown Automata

6.1 Pushdown automata (PDA)

Reading: Sipser 2.2 (pp. 102-114)

- PDA = NFA + Stack (still with limited memory but more than that in FAs)
- PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where
  - Q: A finite set of states
  - Σ: A finite set of input symbols (input alphabet)
  - Γ: A finite set of stack symbols (stack alphabet)
  - δ: The transition function from $Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})$ to $2^{Q \times (\Gamma \cup \{\varepsilon\})}$
  - q_0: The start state
  - F: The set of final states
- What does $\delta(q, a, X) = \{(p, Y)\}$ mean? If the current state is q, the current input symbol is a, and the stack symbol at the top of the stack is X, then the automaton changes to state p and replace X by Y. What if ε replaces a, or X, or Y?
- The state diagram of PDAs: For transition $\delta(q, a, X) = \{(p, Y)\}$, draw an arc from state q to state p labeled with $a, X \to Y$.
- Instantaneous description (ID) of a PDA: (q, w, γ) represents the configuration of a PDA in the state of q with the remaining input of w yet to be read and the stack content of γ. (The convention is that the leftmost symbol in γ is at the top of the stack.)
- Binary relation $\vdash$ on ID's: $(q, aw, X\beta) \vdash (p, w, Y\beta)$ if $\delta(q, a, X)$ contains (p, Y). $\vdash$ represents one move of the PDA, and $\overset{*}{\vdash}$ represents zero or more moves of the PDA.
- Language of a PDA M (or language recognized by M) is $L(M) = \{w | (q_0, w, \varepsilon) \overset{*}{\vdash} (f, \varepsilon, \gamma) \text{ for } f \in F\}$.
- How does a PDA check the stack is empty? At the beginning any computation, it always pushes a special symbol \$ to the initially empty stack by having transition $\delta(q_0, \varepsilon, \varepsilon) = \{(q, \$)\}$.

Example (Sipser p. 112): A PDF that recognizes $\{0^n 1^n | n \geq 0\}$.
Example (Sipser p. 114): A PDA that recognized for $\{a^i b^j c^k | i, j, k \geq 0, i = j \text{ or } i = k\}$.
Example: How the PDA in the above example accepts input *aabbcc*.

6.2 Equivalence of PDAs and CFGs

Reading: Sipser 2.2 (pp. 115-112)

- From CFG to PDA: Let G be the CFG for CFL A. Construct a PDA M such that $L(M) = L(G) = A$.

  Given a CFG $G = (V, \Sigma, R, S)$. Define a PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ to simulate leftmost derivations in CFGs, where
  - $Q = \{q_0, q_1, q_2, q_3\} \cup Q'$, where Q' is a set of additional states that may be added later on.
  - $F = \{q_3\}$
  - $\Gamma = V \cup \Sigma \cup \{\$\} \cup \{\varepsilon\}$
  - The transition function is defined as follows:
    * $\delta(q_0, \varepsilon, \varepsilon) = \{(q_1, \$)\}$
    * $\delta(q_1, \varepsilon, \varepsilon) = \{(q_2, S)\}$

* For each rule $X \to X_k \cdots X_1$ in R with $k \geq 1$, $\delta(q_2, \varepsilon, X) = \{(q_2^1, X_1)\}$, $\delta(q_2^1, \varepsilon, \varepsilon) = \{(q_2^2, X_2)\}$, ..., $\delta(q_2^{k-1}, \varepsilon, \varepsilon) = \{(q_2, X_k)\}$
* For eachh rule $X \to \varepsilon$ in R, $\delta(q_2, \varepsilon, X) = \{(q_2, \varepsilon)\}$
* For each $a \in \Sigma$, $\delta(q_2, a, a) = \{(q_2, \varepsilon)\}$.
* $\delta(q_2, \varepsilon, \$) = \{(q_3, \varepsilon)\}$

Example (Sipser p. 118): $S \to aTb|b$, $T \to Ta|\varepsilon$.

- From PDA to CFG: Let M be a PDA, where there is only one final state, the stack is emptied upon acceptance, and any transition either pushes or pops (but not both or neither). Construct a CFG G such that $L(M) = L(G)$.

  Given a PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, \{q_f\})$. Define CFG $G = (V, \Sigma, R, S)$, where $V = \{A_{pq} | p, q \in Q\}$, $S = A_{q_0 q_f}$, and R has the following rules:

  – For each $p \in Q$, $A_{pp} \to \varepsilon$. (There may be other rules for A_{pp}.)

  – For each $p, q, r \in Q$, $A_{pq} \to A_{pr} A_{rq}$. (This is when the stack becomes empty at least once between states p and q.)

  – For each $p, q, r, s \in Q$, $t \in \Gamma$, and $a, b \in \Sigma \cup \{\varepsilon\}$, if $(r, t) \in \delta(p, a, \varepsilon)$ and $(q, \varepsilon) \in \delta(s, b, t)$, then $A_{pq} \to a A_{rs} b$. (This is when the stack never becomes empty between states p and q.)

  Note: Variable A_{pq} is intended to generate all strings that can take M from state p with an empty stack to state q with an empty stack. This is why we set $S = A_{q_0 q_f}$.

  Theorem: The equivalence of PDAs, CFGs, and CFLs.

6.3 Deterministic pushdown automata

- DPDA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where $\delta(q, a, X)$ has at most one member for any $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, and $X \in \Gamma$. Further, since a DPDA is allowed to make a move on ε, there must be that $\delta(q, a, X) \neq \emptyset$ for some $a \in \Sigma$ implies $\delta(q, \varepsilon, X) = \emptyset$.

- Set of regular languages $\subset$ set of languages accepted by DPDAs.

  Given any DFA $M_1 = (Q, \Sigma, \delta_1, q_0, F)$, we define an equivalent DPDA $M_2 = (Q \cup \{s\}, \Sigma, \{\$\}, \delta_2, s, F)$, where $\delta_2(s, \varepsilon, \varepsilon) = \{(q_0, \$)\}$ and $\delta_2(q, a, \$) = \{(p, \$)\}$ if $\delta_1(q, a) = p$. Furthermore, there are languages accepted by DPDAs that are not regular languages. Can you find an example?

- Set of languages accepted by DPDAs $\subset$ set of context-free languages.

  Any language accepted by a DPDA is also a context-free language since any DPDA is a special PDA. Furthermore, there are context-free languages that are not acceptable by any DPDA's. Can you find an example?