Model of Computation for Deterministic Pushdown Automata

To define the model of computation for a DPDA, we define the extended transition function, $\delta^*$, similarly to how we did for DFAs, except we need to model the stack.

Recall that the transition function is:

$$\delta : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma$$

What is the type of the extended transition function of a DPDA, $\delta^*$:

As with DFAs, we can define $\delta^*$ for all possible inputs using induction on the input string. But, we need to be careful to consider all cases for the stack transitions.

$$\delta^*(q, \epsilon, s) = (q, s)$$

For all $a \in \Sigma, x \in \Sigma^*, \gamma \in \Gamma^*$: $\delta^*(q, ax, \gamma) =$

1. if $(q_t, \epsilon) \in \delta(q, a, \epsilon)$:

2. $\forall h_i \in \Gamma_\epsilon$, if $\gamma = \text{push}(h, \gamma_r)$:

$$\delta^*(q, ax, \text{push}(h, \gamma_r)) = (q_t, \text{push}(h_i, \gamma_r))$$

3. What is missing? (left as exercise for PS3)

NPDA-1
**Accepting State Model:** A deterministic pushdown automata, \( A = (Q, \Sigma, \Gamma, \delta, q_0, F) \) accepts a string \( w \in \Sigma^* \) if and only if:

\[
\delta^*(q_0, w, []) \rightarrow (q_f, s) \text{ and } q_f \in F
\]

**Weak Empty Stack Model:** A deterministic pushdown automata, \( A = (Q, \Sigma, \Gamma, \delta, q_0) \) (note there is no \( F \) now) accepts a string \( w \in \Sigma^* \) if and only if:

\[
\delta^*(q_0, w, []) \rightarrow (q, s) \text{ and } s = \epsilon
\]

Can all languages that can be accepted by the *accepting state model* be accepted by the *weak empty stack model*?

**Empty Stack Model:** A deterministic pushdown automata, \( A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0) \) accepts a string \( w \in \Sigma^* \) if and only if:

\[
\delta^*(q_0, w, Z_0) \rightarrow (q, s) \text{ and } s = \epsilon
\]

Challenge question: is the set of languages that can be recognized by a DPDA under the accepting state model equivalent to the set of languages that can be recognized by a DPDA under the empty stack model?

**Nondeterministic Pushdown Automaton**

A *nondeterministic pushdown automaton* (this is what Sipser calls a *pushdown automaton*) is a 6-tuple \( (Q, \Sigma, \Gamma, \delta, q_0, F) \) where \( Q, \Sigma, \Gamma, q_0, F \) are defined as they are for DPDA and the transition function is defined:

\[
\delta : Q \times \Sigma \times \Gamma \rightarrow \{\epsilon\}
\]

**Example.** Define a NPDA that recognizes the language \( \{ww | w \in \Sigma^*\} \).