Push Down Automata and Deterministic Push Down Automata

Definitions

We use the following definitions given in your textbook, Formal Languages and Automata, by Peter Linz.

**Definition 1** If \( \Sigma \) is any alphabet, let \( \Sigma_\lambda = \Sigma \cup \{ \lambda \} \). (Most books use \( \varepsilon \) instead of \( \gamma \), but we’ll stick to the notation in our textbook.)

**Definition 2** If \( S \) is any set, \( \mathcal{P}(S) = 2^S \) is the set of all subsets of \( S \). We write \( \mathcal{F}(S) \subseteq \mathcal{P}(S) \) to be the set of all finite subsets of \( S \).

**Definition 3** A PDA is a septuple (7-tuple) \( M = (Q, \Sigma, \Gamma, \delta, q_0, z, F) \) where

1. \( Q \) is the set of states,
2. \( \Sigma \) is the input alphabet,
3. \( \Gamma \) is the stack alphabet,
4. \( \delta : Q \times \Sigma \delta \times \Gamma \rightarrow \mathcal{F}(Q \times \Gamma^*) \) is the transition function,
5. \( q_0 \in Q \) is the start state, and
6. \( z \in \Gamma \) is the bottom-of-stack symbol.
7. \( F \subseteq Q \) is the set of accept states.

Configurations and Computations of a PDA

If \( M = (Q, \Sigma, \Gamma, \delta, q_0, F) \) is a PDA, we define a configuration of \( M \) to be an ordered triple \( i = (\sigma, q, w) \) where \( \sigma \in \Gamma^* \) is the current stack, \( q \in Q \) is the current state, and \( w \in \Sigma^* \) is the current (unread) input.

The start configuration of \( M \) is the configuration \( (z, q_0, w) \) for some \( w \in \Sigma^* \). A configuration \( (\sigma, q, w) \) is final if \( q \in F \) and \( w = \lambda \). If \( i \) and \( i' \) are configurations of \( M \), we write \( i \xrightarrow{\cdot} i' \) to mean that, if \( M \) is in configuration \( i \), then \( M \) can possibly read a symbol, pop the top symbol off stack, then push any string onto the stack, then change its configuration to \( i' \). We say that \( M \) is deterministic if, for each \( i \), there is at most one choice of \( i' \).

A computation of \( M \) is a finite sequence of configurations \( i_0, i_1 \ldots i_n \) such that \( i_i \xrightarrow{\cdot} i_{i+1} \) for all \( i \). \( i_0 \) is a start configuration of \( M \). and \( i_n \) is a final configuration of \( M \).
Examples

\[
a/zuza
b/u/λ
a/u/uu
\]

Palindromes

\[
$\lambda/z/λ$
\]

What is the language?