Problem 6.2.2 part b

Design a PDA to accept the set of all strings with twice as many 0’s as 1’s.

Converting a CFG

One easy solution is to write a CFG:

\[ S \rightarrow S0S1S | S0S1S0S | S1S0S0S | \epsilon \]

Then convert the CFG to a one-state PDA (start symbol S, accept by empty stack):

\[ \delta(q, \epsilon, S) = \{(q, S0S1S), (q, S0S1S0S), (q, S1S0S0S), (q, \epsilon)\} \]
\[ \delta(q, 0, 0) = \{(q, \epsilon)\} \]
\[ \delta(q, 1, 1) = \{(q, \epsilon)\} \]

Constructing a PDA directly

Think of this as an accounting problem. Seeing a 0 in the input gives you one dollar. Seeing a 1 in the input costs two. Maintain your credit or debt on the stack, and accept only if your final balance is nothing. Accept by empty stack. Use the symbol \( C \) to represent a unit of credit and \( D \) to represent a unit of debt. If you:

1. Read a 0 and you have debt, pay off one unit.
   \[ \delta(p, 0, D) = \{(p, \epsilon)\} \]
2. Read a 0 and you have credit, save one more unit.
   \[ \delta(p, 0, C) = \{(p, CC)\} \]
   \[ \delta(p, 0, Z_0) = \{(p, CZ_0)\} \]
3. Read a 1 and you have debt, add two units of debt.
   \[ \delta(p, 1, D) = \{(p, DDD)\} \]
   \[ \delta(p, 1, Z_0) = \{(p, DDZ_0)\} \]
4. Read a 1 and you have credit, then pay as much as you can. Guess whether there is another credit on the stack. If so, pay one unit and go to an intermediate state that pays one unit without consuming input. Otherwise, pay one unit up front and put one unit of debt on the stack.

\[
\delta(p, 1, C) = \{(q, \epsilon), (p, D)\}
\]

\[
\delta(q, \epsilon, C) = \{(p, \epsilon)\}
\]

5. Finally, you can choose to accept whenever the balance is 0 by emptying the stack.

\[
\delta(p, \epsilon, Z_0) = \{(p, \epsilon)\}
\]