PDA Using Two Stacks

Dan Rinehimer
rinehi_d@denison.edu

Derek Wilson
wilson_d@denison.edu

December 10, 2004

---

1 Definition

Our formal definition for a pushdown automaton involves seven components. We write the specification of a PDA P as follows:
$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where:

Q = A set of finite states

Σ = A set of finite input symbols

Γ = A finite stack alphabet for both stacks

δ = The transition function that takes the form $\delta(q, a, X_1, X_2) = (p, \gamma_1, \gamma_2)$ where:

$q \subseteq Q$
$a \subseteq \Sigma$ or $a = \epsilon$
$X_1, X_2 \subseteq \Gamma$
$p \subseteq Q$
$\gamma_1, \gamma_2 \subseteq \Gamma$

q_0 is the start state

Z_0 is the start symbol found on the bottom of both stacks

F is the set of accepting states

2 Relation To Chomsky Hierarchy

The Pushdown Automaton with Two Stacks is equivalent to the power of a Turing Machine with one tape. Therefore, the PDA with Two Stacks falls into the same category as the Turing Machine in the Chomsky Hierarchy:

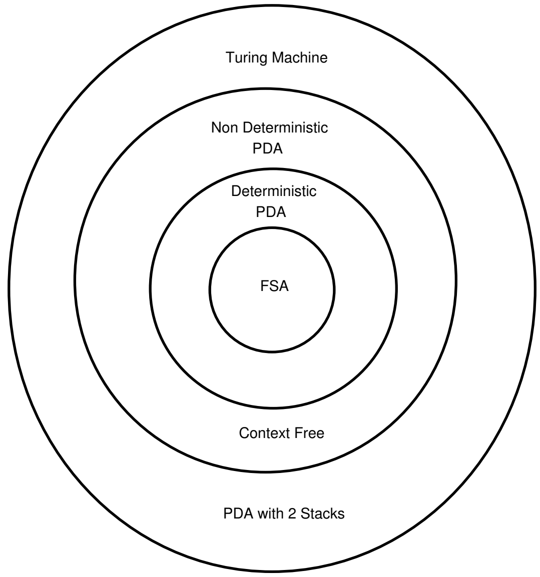

Figure 1: Chomsky Hierarchy

The two stacks of the PDA will represent the different sides of the head on a Turing Machine tape where the first stack represents the left of the head and the second stack represents the head and the data to the right of the head. (Note: The stacks do not contain the blanks at either end of the tape).

Step 1: Copy the input string by pushing all symbols (in order) into the first stack of the PDA.

Step 2: Pop the symbol in the first stack and push this symbol onto the second stack. Repeat this step until only the end of stack symbol remains on the first stack. This leaves the leftmost symbol of the input string at the top of the second stack.

Step 3: A move on a Turing Machine is done on the PDA with 2 Stacks as follows:

i. Moving right

If the tape symbol needs to be changed (say to Z), pop the second stack and push Z onto the first stack and move right.

- or -

If the tape symbol does not need to be changed, pop the second stack and push the popped symbol onto the first stack and move right.

Exceptions

1. If the head of the tape is already at a blank (thus the second stack is empty), nothing happens to the second stack.
2. If the Turing Machine is replacing the current head with a blank and there is nothing to the left of the head, then the first stack remains in its current status.

ii. Moving Left

If the tape symbol needs to be changed (say to Z), pop the first stack and push Z onto the second stack and move left.

- or -

If the tape symbol does not need to be changed, pop the first stack and push the popped symbol onto the second stack and move left.

Exception

1. If the first stack is empty, then do not pop the first stack.

3 Relation to Algorithms

Since the Pushdown Automaton with Two Stack is equivalent to that of a Turing Machine, all PDAs with 2 Stacks can be considered algorithms. If an algorithm is defined as a series of finite procedures that are designed to solve a particular problem, this machine would be considered as one designed to be an algorithm.

4 Examples

1. L = $\{0^n 1^n 2^n | n \geq 1\}$

It is already proven by the pumping lemma that a regular PDA with one stack cannot solve this problem, but the PDA with 2 Stacks can achieve this.

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1, 2\}$

$\Gamma = \{0, 1\}$

Transition Functions:

1. $\delta(q_0, 0, Z_0, Z_0) = (q_0, 0, Z_0)$
2. $\delta(q_0, 0, 0, Z_0) = (q_0, 00, Z_0)$
3. $\delta(q_0, 1, 0, Z_0) = (q_0, 0, 1)$
4. $\delta(q_0, 1, 0, 1) = (q_0, 0, 11)$
5. $\delta(q_0, 2, 0, 1) = (q_1, \epsilon, \epsilon)$
6. $\delta(q_1, 2, 0, 1) = (q_1, \epsilon, \epsilon)$

7. $\delta(q_1, \epsilon, Z_0, Z_0) = (q_2, Z_0, Z_0)$

$F = \{q_2\}$

2. L = {The language of equal 0's and 1's}

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1\}$

Transition Functions:

1. $\delta(q_0, 0, Z_0, Z_0) = (q_0, 0, Z_0)$
2. $\delta(q_0, 0, 0, Z_0) = (q_0, 00, Z_0)$
3. $\delta(q_0, 0, 0, 1) = (q_0, 00, 1)$
4. $\delta(q_0, 0, Z_0, 1) = (q_0, 0, 1)$
5. $\delta(q_0, 1, Z_0, Z_0) = (q_0, Z_0, 1)$
6. $\delta(q_0, 1, 0, Z_0) = (q_0, 0, 1)$
7. $\delta(q_0, 1, Z_0, 1) = (q_0, Z_0, 11)$
8. $\delta(q_0, 1, 0, 1) = (q_0, 0, 11)$
9. $\delta(q_0, \epsilon, 0, 1) = (q_1, 0, 1)$
10. $\delta(q_0, \epsilon, Z_0, Z_0) = (q_2, Z_0, Z_0)$
11. $\delta(q_1, \epsilon, 0, 1) = (q_1, \epsilon, \epsilon)$
12. $\delta(q_1, \epsilon, Z_0, Z_0) = (q_2, Z_0, Z_0)$

$F = \{q_2\}$

3. L = $\{ww | w \text{ is in } \{0, 1\}^*\}$

1. Push all of w onto the first stack stopping at an arbitrary point since the machine cannot tell where the first w ends and the second w begins.
2. Pop all the items on the first stack and push each item onto the second stack. (Note: this procedure will allow the second stack to contain w (or what the machine believes to be w) with its left most character on the top of the stack.)
3. If the input string is the same symbol as that of the top of the second stack, pop the second stack. Repeat this until there is no longer input.
4. The machine reaches a final state when the second stack is empty and there is no more input.
5. If the machine fails to reach a final state, the machine must start the process again, choosing a new position where it believes the first w ends and the second begins. Only when all combinations are checked can the string ww be rejected.

5 Non-Deterministic vs. Deterministic

The Deterministic PDA with Two Stacks is as powerful as the Non-Deterministic PDA with Two Stacks. One difference between the two types is that the deterministic machine would be much more complex having many more steps that may not be necessary in order to solve the problem. It has been proved that all Non-Deterministic Turing Machines can be solved by a deterministic Turing Machine. It has also already been proved that all Turing Machines can be simulated by a PDA with Two Stacks, which includes all non-determinstic Turning Machines. Therefore, it follows that all Non-Deterministic PDAs with Two stacks can also be solved by a deterministic PDA with Two Stacks.

6 Quiz Questions

1. Give the tuple and transition function for the PDA with 2 Stacks to solve the following language:

$$\text{L} = \{a^i b^j c^k | i < j < k\}$$

2. What is the most complex type of problem that all PDA with Two Stack machines can solve?

7 Quiz Solutions

1. Tuple: $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

   $Q = \{q_0, q_1, q_2, q_3\}$

   $\Sigma = \{a, b, c\}$

   $\Gamma = \{a, b\}$

   Transition function:

   1. $\delta(q_0, a, Z_0, Z_0) = (q_0, a, Z_0)$
   2. $\delta(q_0, a, a, Z_0) = (q_0, aa, Z_0)$
   3. $\delta(q_0, b, a, Z_0) = (q_0, a, b)$
   4. $\delta(q_0, b, Z_0, Z_0) = (q_0, Z_0, b)$
   5. $\delta(q_0, b, Z_0, b) = (q_0, Z_0, bb)$
   6. $\delta(q_0, b, a, b) = (q_0, a, bb)$
   7. $\delta(q_0, c, a, b) = (q_1, \epsilon, \epsilon)$
   8. $\delta(q_0, c, Z_0, b) = (q_2, Z_0, \epsilon)$
   9. $\delta(q_1, c, a, b) = (q_1, \epsilon, \epsilon)$
   10. $\delta(q_1, c, Z_0, b) = (q_2, Z_0, \epsilon)$
   11. $\delta(q_2, c, Z_0, b) = (q_2, Z_0, \epsilon)$
   12. $\delta(q_2, c, Z_0, Z_0) = (q_3, Z_0, Z_0)$

$F = \{q_3\}$

2. Since we have already proven that the PDA with Two Stacks is equivalent to a Turing Machine with one tape, we know that the most complex type of problem that the PDA with Two Stacks can solve is Recursive Enumerable because that is the most complex set of problems that a Turing Machine can solve.