Pushdown Automata

Ling 106

October 29, 2003

1. Some Properties of Pushdown Automata

Pushdown automata (PDA) recognize context free languages
These automata are like non-deterministic finite state automata but have an extra component called a **stack**.
It is this extra component that allows the automaton to have memory (in principle, infinite amount of memory), and to recognize some nonregular languages.

A PDA can write (push) a symbol on the top of the stack or remove (pop) a symbol from the top of the stack. The stack is, in principle, unlimited, and works as a LAST IN, FIRST OUT storage device.

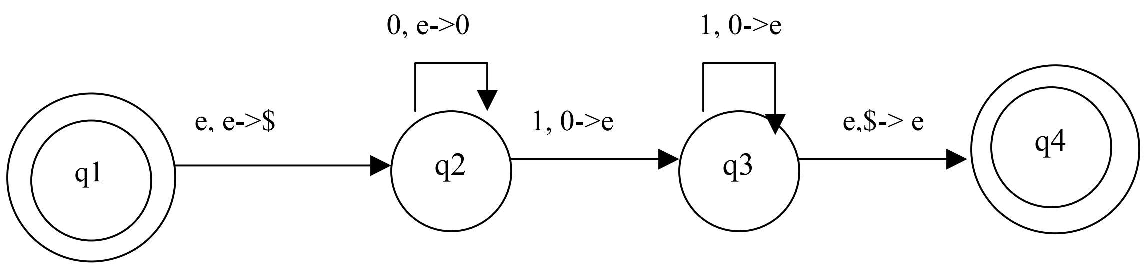
Recall that $\{0^n1^n \mid n \geq 0\}$ cannot be recognized by a finite state automaton. But PDA can recognize it with the help of the stack.

1. As the machine reads a 0 from the input string, push it on top of the stack.
2. As soon as 1s are encountered from the input string, pop a 0 off the stack for each 1 read.
3. If reading the input string is finished exactly when the stack becomes empty of 0s, accept the input. Reject otherwise.

Conditions under which the in[u string is accepted by PDA:

1. ACCEPT the input string if the stack empties when the last symbol is read.
2. REJECT otherwise.

Schematic example:



Notations:

- a, b ->c: when the machine is reading an *a* from the input, it may replace the symbol *b* on the top of the stack with a *c* (pop b and push c).
- Any of *a, b* and *c* may be ϵ.
  If *a* is ϵ, the machine may make this transition without reading any symbol from the input. If *b* is ϵ, the machine may make this transition without popping any symbol from the stack (pop nothing and push c).
  If c is ϵ, the machine does not write any symbol on the stack when going along this transition (pop b and push nothing).
  ϵ ->$ places a special symbol $ on the stack. This mechanism allows the PDA to test for an empty stack. By initially placing $ on the stack, when the machine sees the $ again, it knows that the stack is effectively empty.

1. Formal Definition of Pushdown Automata

A pushdown automaton is a 6-tuple <Q, Σ, Γ, δ, q0, F>, where Q, Σ, Γ, and F are all finite sets and:

1. Q is the set of states,
2. Σ is the input alphabet,
3. Γ is the stack alphabet,
4. δ: $Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow \wp(Q \times \Gamma_\epsilon)$ is the transition function,
5. $q0 \in Q$ is the start state, and
6. $F \subseteq Q$ is the set of accept states.

2. Examples of Pushdown Automata

PDA that recognizes the language $\{0^n 1^n | n \geq 0\}$.
Let M1 be <Q, Σ, Γ, δ, q0, F>, where
Q={q1, q2, q3, q4},
Σ={0, 1},
Γ= {0, $},
F={q1, q4}, and

δ is given by the following table:

Input	0			1			ϵ		
Stack	0	$	ϵ	0	$	ϵ	0	$	ϵ
q1	∅	∅	∅	∅	∅	∅	∅	∅	{(q2, $)}
q2	∅	∅	{(q2, 0)}	{(q3, ϵ)}	∅	∅	∅	∅	∅
q3	∅	∅	∅	{(q3, ϵ)}	∅	∅	∅	{(q4, ϵ)}	∅
q4	∅	∅	∅	∅	∅	∅	∅	∅	∅

Exercises: Construct pushdown automata for each of the languages described below:

a) L1 = $\{a^n bc^n$ where $n \geq 1\}$ $\Sigma = \{a, b, c\}$

b) L2 = $\{ a^n b^{n+1}$ where $n \geq 1\}$ $\Sigma = \{a, b\}$