CMSC 141 – Automata and Language Theory
Handout: Pushdown Automata
By: Gianina Renee V. Vergara

Pushdown Automata

- Like NFA but have an extra component called a **stack**.
- Pushdown automata are equivalent in power to context-free grammars.
- The stack...
  - provides additional memory beyond the finite amount available in the control
  - valuable because it can hold an unlimited amount of information
  - allows pushdown automata to recognize some non-regular languages
- The **current state**, the next **input symbol** read and the **top symbol of the stack** determine the next move of a pushdown automaton.

Def'n: A pushdown automaton is a 7-tuple (Q, Σ, Γ, Δ, q0, Z, F) …

- Γ is the finite set of stack alphabet.
- Z is the start stack symbol (must be a capital Z)
- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$ is the transition function
  - Δ(current state, input, tos) = {(next state, push/pop)}

There are two ways by which a string is accepted by the PDA:

- by final state: $L(M) = \{w \mid (q0, w, Z) \mid\!-^* (p,\varepsilon,\gamma) \text{ for some } p \in F \text{ and } \gamma \in \Gamma^*\}$
- by empty stack: $L(M) = \{w \mid (q0, w, Z) \mid\!-^* (p,\varepsilon,\varepsilon) \text{ for some } p \in Q\}$

We write **a, b → c** to signify that when the machine is reading an **a** from the input it may replace the top of the stack symbol **b** with **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 reading and popping any symbol from the stack.
- If c is ε, the machine does not write any symbol on the stack when going along this transition.

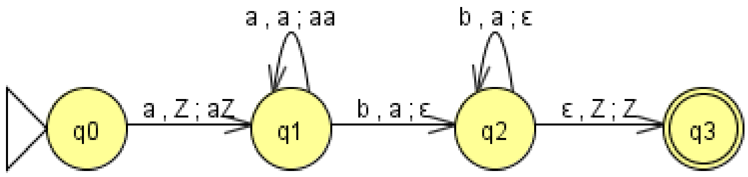
Example 1: Create a PDA for $L = \{a^nb^n \mid n > 0\}$. One approach is to push an **a** on the stack for each **a** read in the input, and to pop an **a** off the stack for each **b** read in the input.

1. Read input symbol. When reading the input symbol, the possible combinations of the input alphabet and stack alphabet are:

input	top of stack	
a	Z	✓ first transition: the stack is empty.
a	a	✓
b	a	✓
~~a~~	~~b~~	✘ no occurrence: **b** will not be pushed on the stack
~~b~~	~~b~~	✘ no occurrence: **b** will not be pushed on the stack
~~b~~	~~Z~~	✘ no occurrence: the string must always begin with **a**

2. If the input symbol is **a**, push **a** on the stack, else if **b**, pop an **a** off the stack.
   - From q0, on input a, go to q1 and remember a: **a, Z → aZ** Δ(q0, a, Z) = {(q1, aZ)}
   - Continue remembering a: **a, a → aa** Δ(q1, a, a) = {(q1, aa)}

- Encountered a **b**, pop its pair off the stack: **b, a → ε** Δ(q1, b, a) = {(q2, ε)}
- Continue to pop **a** for every **b:** Δ(q2, b, a) = {(q2, ε)}
- End of string and bottom of stack reached: **ε, Z → Z** Δ(q2, ε, Z) = {(q3, Z)}



Let us trace the execution of the transition diagram on the input string aaaabbbb

(q0, aaaabbbb, Z)	|- (q1, aaabbbb, aZ)	using Δ(q0, a, Z) = {(q1, aZ)}
	|- (q1, aabbbb, aaZ)	using Δ(q1, a, a) = {(q1, aa)}
	|- (q1, abbbb, aaaZ)	using Δ(q1, a, a) = {(q1, aa)}
	|- (q1, bbbb, aaaaZ)	using Δ(q1, a, a) = {(q1, aa)}
	|- (q2, bbb, aaaZ)	using Δ(q1, b, a) = {(q2, ε)}
	|- (q2, bb, aaZ)	using Δ(q2, b, a) = {(q2, ε)}
	|- (q2, b, aZ)	using Δ(q2, b, a) = {(q2, ε)}
	|- (q2, ε, Z)	using Δ(q2, b, a) = {(q2, ε)}
	|- (q3, ε, Z)	using Δ(q2, ε, Z) = {(q3, Z)}

Building Your First Pushdown Automaton with JFLAP

Equivalence of CFG and PDA

- we can construct a PDA M = ({q0,q1}, T, V, Δ, q0, Z, {q1}) where Δ is given by:
  1. Δ(q0, ε, Z) = {(q1, SZ)}
  2. for each rule A → x in P,
     Δ(q1, ε, A) = {(q1, x)}
  3. for each a ∈ T,
     Δ(q1, a, a) = {(q1, ε)}

Example 2: Construct a PDA for the grammar whose set of productions is:

P = { S → 0S0 | 1S1 | c }

Δ(q0, ε, Z) = {(q1, SZ)}

Δ(q1, ε, S) = {(q1, 0S0)}
Δ(q1, ε, S) = {(q1, 1S1)}
Δ(q1, ε, S) = {(q1, c)}

Δ(q1, 0, 0) = {(q1, ε)}
Δ(q1, 1, 1) = {(q1, ε)}
Δ(q1, c, c) = {(q1, ε)}

Δ(q1, ε, Z) = {(q2, ε)}

Convert CFG to PDA (LL) with JFLAP
Convert CFG to PDA (LR) with JFLAP

Links are all from jflap.org/tutorial/