CS301: Theory of Computation
Section 2.2 - Pushdown Automata

Daniel White

Gettysburg College
Computer Science

Spring 2024

We will introduce a new type of computational model called the **pushdown automata**. These are like nondeterministic finite automata, but with an infinitely large stack.

Pushdown automata are equivalent in power to context-free grammars, as we'll demonstrate in part. We will therefore have two ways of proving a language is context-free.

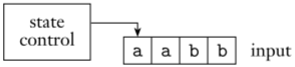


FIGURE 2.11
Schematic of a finite automaton

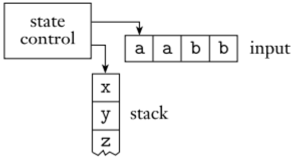


FIGURE 2.12
Schematic of a pushdown automaton

Pushdown automata (PDA) can write symbols on their stack and read them back later.

- Writing pushes down all the other symbols on the stack. This is known as **pushing** the symbol.
- The symbol on the top of the stack can be read and removed. This is known as **popping**. The remaining symbols move up.
- State diagrams will use labels of the form "$a, b \to c$" to signify that the machine reads a from the input string, pops b from the stack, and finally pushes c to the stack.
- Any of a, b, or c may be ε.

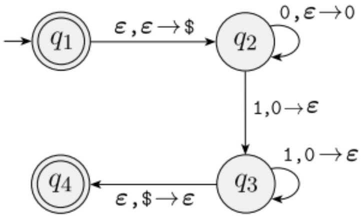
exercise: Assuming PDA have the same descriptive power as CFG, argue that the class of CFL properly contains the regular languages.

Definition 2.13 (pg. 113)

A **pushdown automaton** is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$, 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. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is the transition function,
5. $q_0 \in Q$ is the start state, and
6. $F \subseteq Q$ is the set of accept states.

We argued earlier that $\{0^n1^n \mid n \geq 0\}$ is context-free by the construction of a grammar that generates it. Below is a PDA, M_1, which recognizes this same language.



exercise: Express M_1 formally.

$\boxed{\text{homework: 2.7}}$

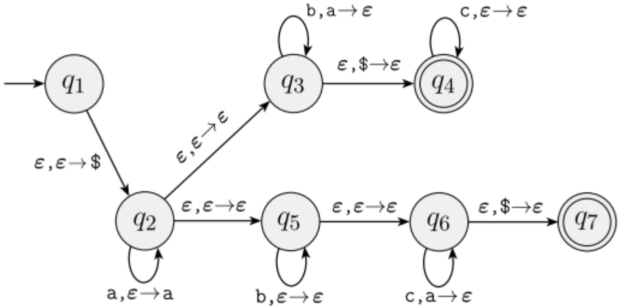
exercise: Create a PDA that recognizes the language

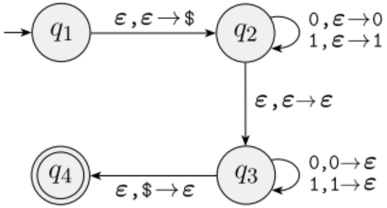
$$B = \{\mathrm{a}^i\mathrm{b}^j\mathrm{c}^k \mid i,j,k \geq 0 \text{ and } i = j \text{ or } i = k\}.$$

exercise: Create a PDA that recognizes the language

$$C = \{ww^{\mathcal{R}} \mid w \in \{0,1\}^*\}.$$

$\boxed{\text{homework: 2.10}}$


q_1
q_2
q_3
q_4
q_5
q_6
q_7
$\varepsilon,\varepsilon\to\$$
$\varepsilon,\varepsilon\to\varepsilon$
$b,a\to\varepsilon$
$\varepsilon,\$\to\varepsilon$
$c,\varepsilon\to\varepsilon$
$a,\varepsilon\to a$
$\varepsilon,\varepsilon\to\varepsilon$
$b,\varepsilon\to\varepsilon$
$\varepsilon,\varepsilon\to\varepsilon$
$c,a\to\varepsilon$
$\varepsilon,\$\to\varepsilon$


q_1
$\varepsilon, \varepsilon \to \$$
q_2
$0, \varepsilon \to 0$
$1, \varepsilon \to 1$
$\varepsilon, \varepsilon \to \varepsilon$
q_3
$0, 0 \to \varepsilon$
$1, 1 \to \varepsilon$
$\varepsilon, \$ \to \varepsilon$
q_4

Theorem 2.20 (pg. 117)

A language is context free if and only if some pushdown automaton recognizes it.

As usual, we have two directions that need proved. We will only prove the forward direction in this course.

Lemma 2.21 (pg. 117)

If a language is context free, then some pushdown automaton recognizes it.

Proof idea. Let A be a CFL generated by some CFG, G. We show how to convert G into an equivalent PDA, P ...

We design the state diagram so that all legal substitution rules can be applied nondeterministically. For each intermediate string, all input symbols to the left of the first variable are checked against the input string. The remainder is stored on the stack.

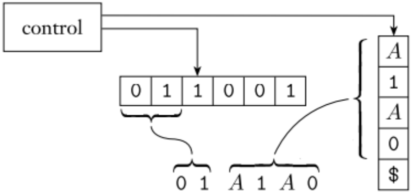


FIGURE 2.22
P representing the intermediate string $01A1A0$

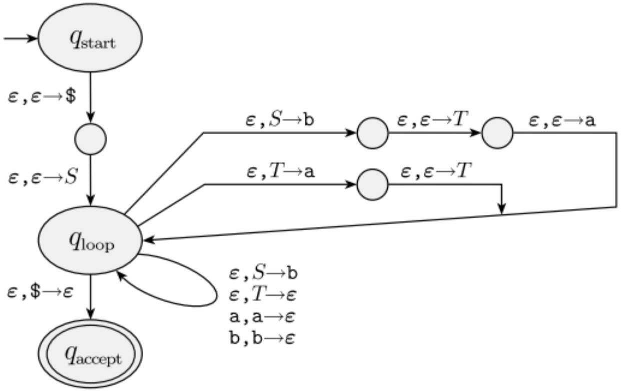
Create P in the following way.

1. Place the marker symbol \$ and the start variable on the stack.
2. Repeat the following steps sufficiently-many times.
   - If the top of the stack is a variable A, nondeterministically select one of the rules for A and perform that substitution for A on the stack.
   - If the top of the stack is a terminal a, read the next sumbol from the input and compare it to a. If they do not match, reject on this branch of nondeterminism.
   - If the top of the stack is the symbol \$, enter the accept state. The input is accepted if all input symbols have been read.

exercise: Use the procedure developed in Lemma 2.21 to construct a PDA P_1 from the following CFG, G.

$$S \to \mathtt{a}T\mathtt{b} \mid \mathtt{b}$$
$$T \to T\mathtt{a} \mid \varepsilon$$

$\boxed{\text{homework: 2.12}}$


q_{start}
$\varepsilon,\varepsilon\to\$$
$\varepsilon,\varepsilon\to S$
q_{loop}
$\varepsilon,S\to$ b
$\varepsilon,\varepsilon\to T$
$\varepsilon,\varepsilon\to$ a
$\varepsilon,T\to$ a
$\varepsilon,\varepsilon\to T$
$\varepsilon,S\to$ b
$\varepsilon,T\to\varepsilon$
a,a$\to\varepsilon$
b,b$\to\varepsilon$
$\varepsilon,\$\to\varepsilon$
q_{accept}