Pushdown Automata

Lecture 21
Section 7.1

Robb T. Koether

Hampden-Sydney College

Wed, Oct 12, 2016

Outline

Outline

1. Machines for CFGs
2. Pushdown Automata
3. Examples
   - Balanced Parentheses
   - Algebraic Expressions
4. Assignment

Grammars vs. Machines

Definition (Context-free language)

A context-free language (CFL) is the language $L(G)$ of a context-free grammar G.

- Given a regular language, we can describe it by using a regular grammar or a machine (a DFA).
- A context-free language can be described by using a context-free grammar.
- Can it also be described by a machine?

Machines for CFLs

- If a machine is to process the string $\mathbf{a}^n\mathbf{b}^n$, then it must be able to "remember" the number of **a**'s.
- We will use a stack to do this.
- As each **a** is read, push it onto the stack.
- When **b** is read, pop an **a**.
- When we are finished reading the string, the stack should be empty.

Machines for CFLs

- Each transition will include five parts.
  - The symbol read.
  - The symbol popped.
  - The string pushed.
  - Two states (the state we go from and the state we go to).
- Such a machine is called a push-down automaton (PDA).

Machines for CFLs

- The symbol read and the string pushed could be λ, but we must pop exactly one symbol from the stack.

Machines for CFLs

- The symbol read and the string pushed could be λ, but we must pop exactly one symbol from the stack. Them's the rules.

Machines for CFLs

- The symbol read and the string pushed could be λ, but we must pop exactly one symbol from the stack. Them's the rules.
- The current state and the symbols read and popped serve as "input."
- The destination state and the string pushed serve as "output."

Machines for CFLs

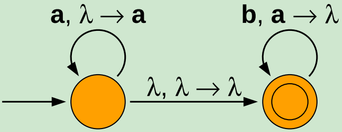
- The symbol read and the string pushed could be λ, but we must pop exactly one symbol from the stack. Them's the rules.
- The current state and the symbols read and popped serve as "input."
- The destination state and the string pushed serve as "output."
- PDAs are inherently nondeterministic. They are sometimes called NPDAs.

Machines for CFLs

- The symbol read and the string pushed could be λ, but we must pop exactly one symbol from the stack. Them's the rules.
- The current state and the symbols read and popped serve as "input."
- The destination state and the string pushed serve as "output."
- PDAs are inherently nondeterministic. They are sometimes called NPDAs.
- There are also deterministic PDAs, called DPDAs.

Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$

Example (Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)



A machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$: First attempt

Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$

Example (Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)

- The first shortcoming is that we reach the final state without reading any **b**'s.
- Thus, this machine would accept, for example, **aaaa**, **aaaab**, and **aaaabb**, as well as **aaaabbbb**.
- We need to read all the **b**'s in one state and them move to a final state.

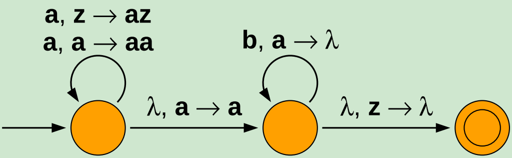
Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$

Example (Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)

- A second shortcoming is that not every transition pops exactly one symbol, as required.
- To make the initial transition, there must already be a symbol on the stack.
- We will designate a special symbol, typically **z**, to be the start stack symbol.

Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$

Example (Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)



A mahcine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$: Second attempt

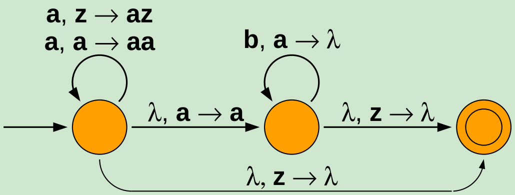
Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$

Example (Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)

- This machine still has one shortcoming.
- If we require the transition $\lambda, \mathbf{a} \to \mathbf{a}$ to move from the first state to the second state, then the machine will not accept λ.
- To remedy this, we could add a second possibility: $\lambda, \mathbf{z} \to \mathbf{z}$, but it would be simpler just to add a transition $\lambda, \mathbf{z} \to \mathbf{z}$ directly from the start state to the final state.

Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$

Example (Machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)



A machine for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$: Final attempt

Outline

Pushdown Automaton

Definition (Pushdown automaton)

A pushdown automaton, abbreviated PDA, is a septuple $(Q, \Sigma, \Gamma, \delta, q_0, z, F)$, where

- Q is a finite set of states.
- Σ is a finite input alphabet.
- Γ is a finite stack alphabet.
- $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to \mathcal{P}'(Q \times \Gamma^*)$ is the transition function.
- $q_0 \in Q$ is the start state.
- $z \in \Gamma$ is the start stack symbol.
- $F \subseteq Q$ is the set of accept states.

where $\mathcal{P}'$ means the set of finite subsets.

Example

Example (PDA for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)

- The PDA that accepts

$$\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$$

  is

  - $Q = \{q_0, q_1, q_2\}$
  - $\Sigma = \{\mathbf{a}, \mathbf{b}\}$
  - $\Gamma = \{\mathbf{a}, \mathbf{z}\}$
  - $F = \{q_2\}$

Example

Example (PDA for $\{\mathbf{a}^n\mathbf{b}^n \mid n \geq 0\}$)

- and δ is given by
  - $\delta(q_0, \mathbf{a}, \mathbf{z}) = \{(q_0, \mathbf{az})\}$
  - $\delta(q_0, \mathbf{a}, \mathbf{a}) = \{(q_0, \mathbf{aa})\}$
  - $\delta(q_0, \lambda, \mathbf{z}) = \{(q_2, \lambda)\}$
  - $\delta(q_0, \lambda, \mathbf{a}) = \{(q_1, \mathbf{a})\}$
  - $\delta(q_1, \mathbf{b}, \mathbf{a}) = \{(q_1, \lambda)\}$
  - $\delta(q_1, \lambda, \mathbf{z}) = \{(q_2, \lambda)\}$

Outline

Example

Example (Pushdown automaton)

- Design a PDA that accepts the language

$$\{w \mid w \text{ contains an equal number of } \mathbf{a}\text{'s and } \mathbf{b}\text{'s}\}.$$

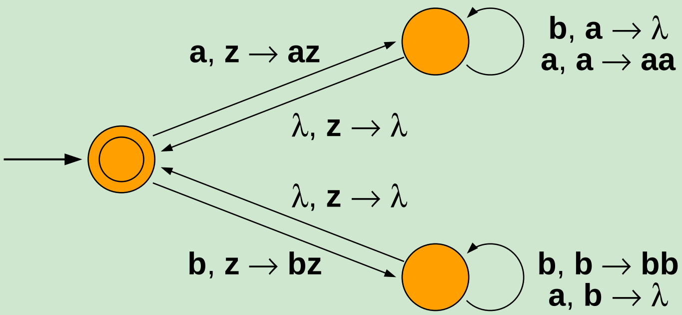
Example

Example (Pushdown automaton)

- The strategy will be to keep the excess symbols, either **a**'s or **b**'s, on the stack.
- One state will represent an excess of **a**'s.
- Another state will represent an excess of **b**'s.
- We can tell when the excess switches from one symbol to the other because at that point the stack will be empty.
- In fact, when the stack is empty, we may return to the start state.

Example

Example (Pushdown automaton)



Outline

Examples

Example (Pushdown automata)

- Let $\Sigma = \{\mathbf{a}, (,)\}$. Design a PDA whose language is

$$\{w \in \Sigma^* \mid w \text{ contains balanced parentheses}\}.$$

Outline

Examples

Example (Pushdown automata)

- Let $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, +, \times, (,)\}$. Design a PDA whose language is

$$\{w \in \Sigma^* \mid w \text{ is a valid algebraic expression}\}.$$

Outline

Assignment

Assignment

- Section 7.1 Exercises 1, 3, 6bcdfj.