CS 252

Theory of Automata, Formal Languages, and Computation

Lecture 16: Pushdown Automata

Xenofon Koutsoukos
Department of Electrical Engineering and Computer Science
Vanderbilt University

Spring 2004

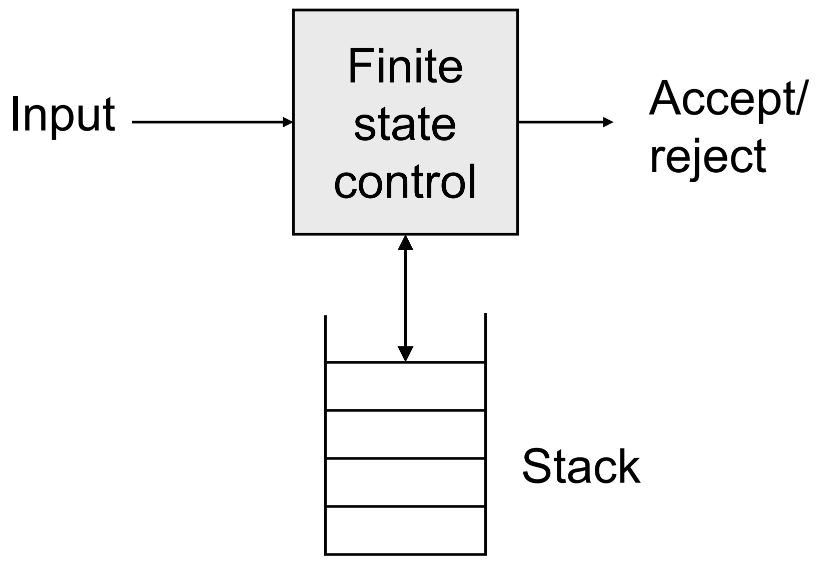
Outline

- Informal introduction to pushdown automata
- Formal definition of pushdown automata
- A graphical notation of PDAs
- Instantaneous descriptions of a PDA
- Examples

Pushdown Automata

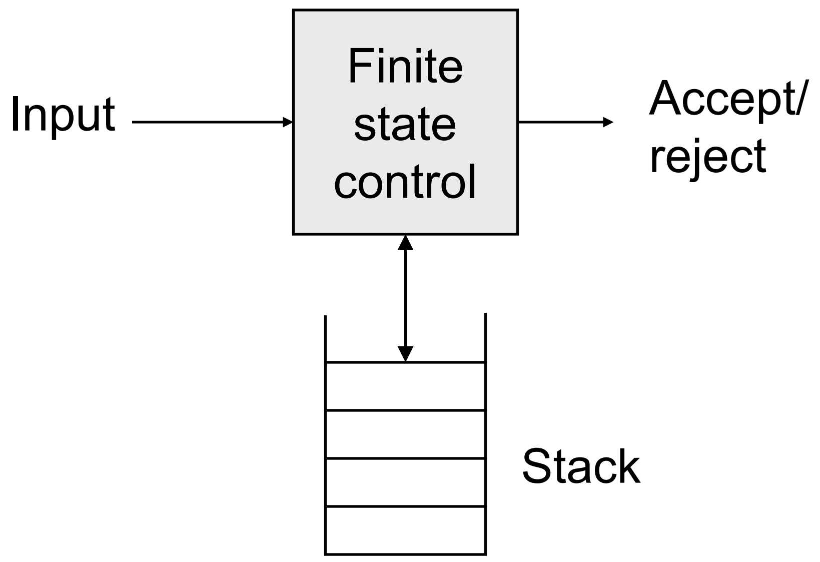
- Pushdown automata are computing models that define context-free languages
- The pushdown automaton consists of
  - A nondeterministic finite automaton with ε-transitions
  - A stack for storing strings of symbols
- The presence of the stack means that the automaton can "remember" more information than the information that it is captured at its current state

Informal Introduction to PDAs



- A PDA informally consists of a "finite state control" which is an ε-NFA and a stack
- The automaton reads inputs one symbol at a time
- The automaton observes the symbol at the top of the stack
- The transitions are based on the current state, input symbol, and the symbol at the top of the stack

Informal Introduction to PDAs



- In one transition, the PDA
  1. Consumes an input symbol
     - If ε is the input, no symbol is consumed
  2. Goes to a new state
     - Which may be its old state
  3. Replaces the symbol at the top of the stack by any string
     - The same symbol that appeared in the stack previously (no change)
     - A new symbol that changes the top of the stack
     - A string with two or more symbols

Example

- Consider the "*w-w-reversed*" language
  $L_{wwr} = \{ww^R \mid w \in \{0,1\}^*\}$
  - The even-length palindromes over alphabet $\{0,1\}$

- L_{wwr} is a CFL generated by the grammar
  $P \to 0P0 \mid 1P1 \mid \varepsilon$

- A PDA for L_{wwr} has three states and operates as follows:
  1. Start in a state q_0 that represents a "guess" that we have not yet seen the middle. While in state q_0, we read input symbols and push them in the stack
  2. At any time, we guess that we have seen the middle and we spontaneously go to state q_1
  3. Once in q_1 we compare the input symbol with the symbol at the top of the stack. If they match, we consume the input symbol, pop the stack, and proceed. If not this branch dies.
  4. If the stack is empty go to state q_2 and accept

Formal Definition of PDA

- A PDA is a seven-tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
  - Q is a finite set of *states*
  - Σ is a finite *input alphabet*
  - Γ is a finite *stack alphabet*
  - δ is the *transition function*
  - q_0 is the *start state*
  - Z_0 is the *start symbol* for the stack
  - $F \subseteq Q$ is the set of *accepting states*

Transition Function

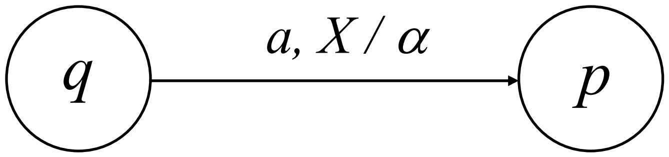
- The transition function is defined as

$$\delta : Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$$

- δ takes as argument a triple $\delta(q,a,X)$
  1. q is a state in Q
  2. a is an input symbol in Σ or $a = \varepsilon$
  3. X is a stack symbol in Γ
- The output of δ is a set of pairs (p,γ)
  1. p is the new state
  2. γ is the string of stack symbols that replaces X at the top of the stack
     - If $\gamma = \varepsilon$ the stack is popped
     - If $\gamma = X$ the stack is unchanged
     - If $\gamma = YZ$, X is replaced by Z and Y is pushed onto the stack

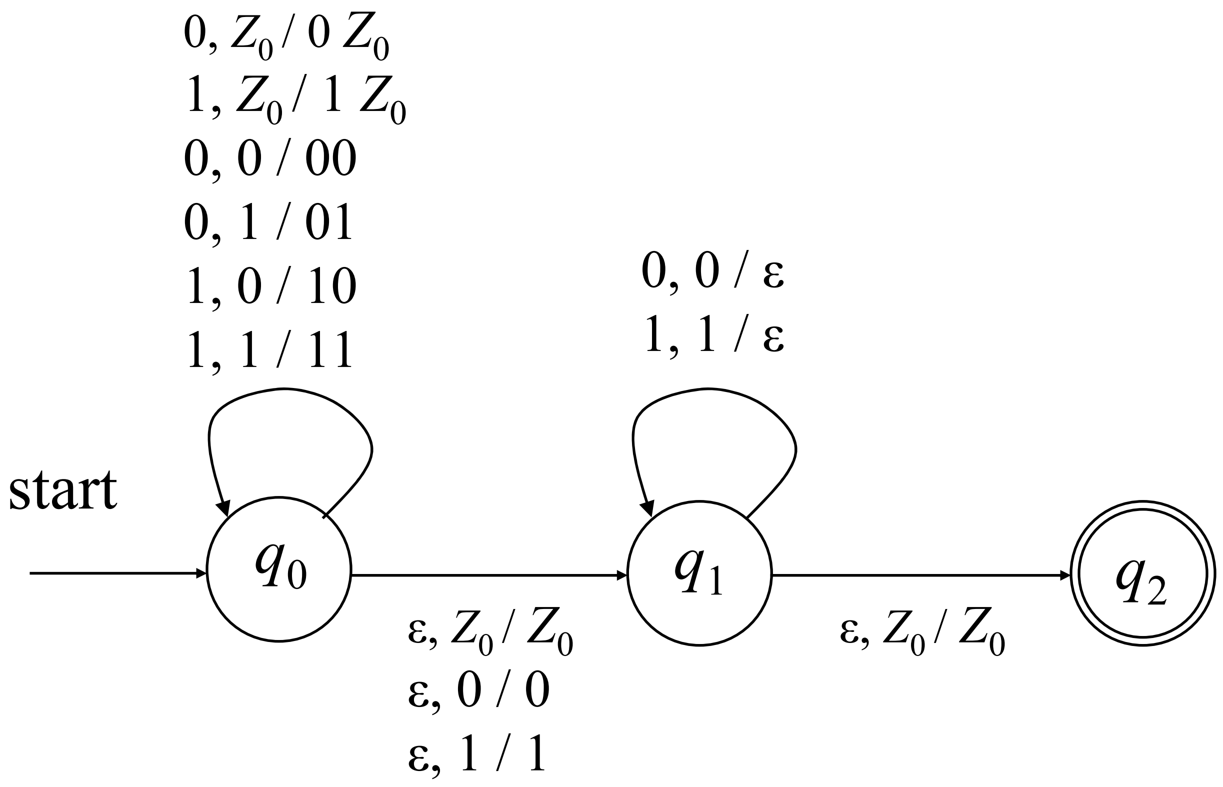
A Graphical Notation for PDAs

- PDAs are represented by a *transition diagram* where
  1. The nodes correspond to the states of the PDA
  2. An arrow labeled *Start* indicates the start state
  3. The arcs correspond to transitions
     - An arc from state q to state p labeled by $a, X / \alpha$ means that $\delta(q,a,X)$ contains the pair (p,α)
     - The arc label tells us what input symbol is used, and the old and new top of the stack



Example

- The following PDA accepts L_{wwr}



$$P = (\{q_0, q_1, q_2\}, \{0,1\}, \{0,1,Z_0\}, \delta, q_0, Z_0, \{q_2\})$$

Instantaneous Descriptions

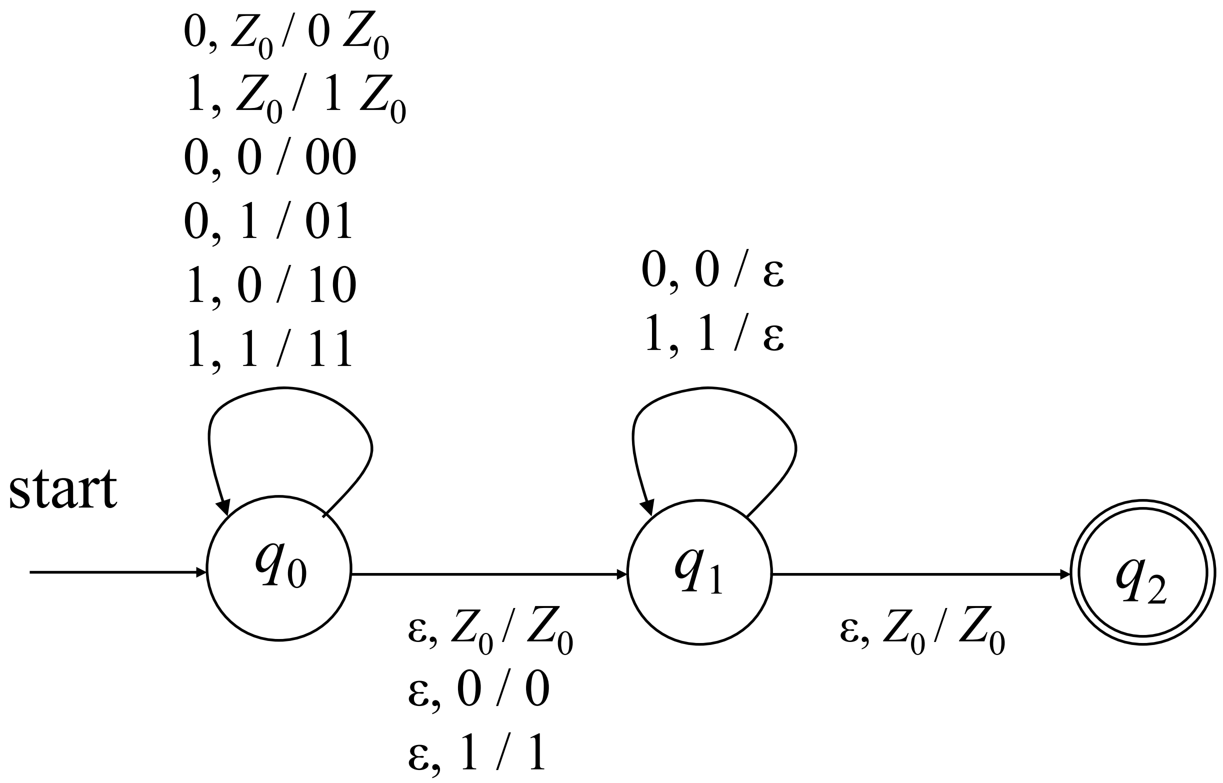
- The PDA goes from configuration to configuration in response to input symbols
- The PDA's configuration involves both the state of the automaton and the contents of the stack
- We represent the configuration of a PDA by the triple (q, w, γ)
  1. q is the state
  2. w is the remaining input, and
  3. γ is the stack content
- Such a triple is called ***instantaneous description (ID)*** of the PDA

Instantaneous Descriptions

- Let $P = (Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$ be a PDA and assume that $\delta(q,a,X)$ contains (p,α)
- A move of the PDA is denoted as $(q,aw,X\beta) \vdash (p,w,\ \alpha\beta)$
- What remains on the input, w, and what is below the top of the stack, β, do not influence the action of the PDA
- We use the symbol $\overset{*}{\vdash}$ to represent zero or more moves of the PDA

Example

- Show all the IDs for the following PDA when the input is 1111



Properties of IDs

- Data that P never looks at cannot affect its computation
  1. If a sequence of IDs (***computation***) is legal for a PDA P, then the computation formed by adding the same input string at the end of the input (second component) in each ID is also legal
  2. If a computation is legal for a PDA P, then the computation formed by adding the same stack symbols below the stack in each ID is also legal
  3. If a computation is legal for a PDA P, and some tail of the input is not consumed, then removing this tail from the input in each ID results in a legal computation

Example

- Construct a PDA for the language
  $L = \{w \in \{a,b\}^* | \, n_a(w) = n_b(w)\}$
- Show the sequence of IDs for the string $baab$

Next Lecture

- The language for a PDA