Parul® University

NAAC A++

Non-Deterministic Pushdown Automata (PDA) and Equivalence with CFG
Chapter 3: Grammars

Prof. Riddhi Atulkumar Mehta
Assistant Professor
Department of Computer Science and Engineering

Parul® University

NAAC A++

INDEX

Content

Introduction

- What is a Pushdown Automaton (PDA)?

  A PDA is a finite automaton equipped with a stack.

  Used to recognize context-free languages (CFLs).

- Two types:

  Deterministic PDA (DPDA)

  Non-deterministic PDA (NPDA)

Formal Definition of NPDA

NPDA is a 7-tuple:

$M=(Q,\Sigma,\Gamma,\delta,q0,Z0,F)$

where:

- Q: set of states
- Σ: input alphabet
- Γ: stack alphabet
- δ: transition function
- q0: start state
- Z0: initial stack symbol
- F: set of accepting states

NPDA Transition Function

$\delta: Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P(Q \times \Gamma*)$

Meaning:

- Based on current state, input symbol (or ε), and top of the stack.
- Can move to new states and update the stack.

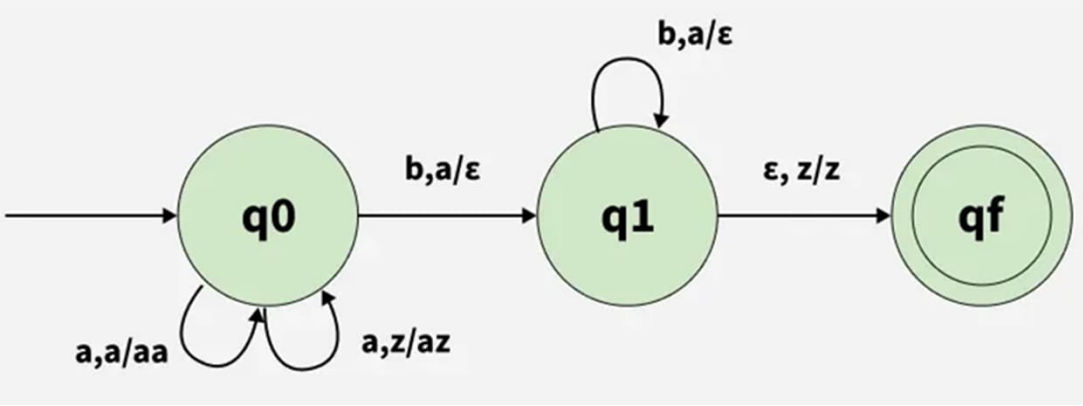
How NPDA Works

- Stack enables NPDA to:
  - Remember past input.
  - Handle nested structures (like parentheses).
- Non-determinism:
  - Multiple possible moves at each step.
  - Accepts if any computation path leads to an accepting state.

Example of NPDA

Design a non deterministic PDA for accepting the language L = {an bn | n>=1}, i.e.,

L = {ab, aabb, aaabbb, aaaabbbb, ...}



Stack transition functions

δ (q0, a, z) ⊢ (q0, az) [push a on empty stack]

δ (q0, a, a) ⊢ (q0, aa) [push a's]

Context-Free Grammars (CFG)

- CFG is a 4-tuple (V,Σ,R,S)
- V: variables (non-terminals)
- Σ: terminals
- R: production rules
- S: start symbol
- Generates context-free languages (CFLs)

Equivalence of NPDA and CFG

- Theorem: Every language accepted by a NPDA is generated by a CFG.
- And vice versa: For every CFG, there exists an equivalent NPDA.

Parul® University

NAAC GRADE A++

https://paruluniversity.ac.in/