CS304: Automata and Formal Languages

Lec 17

PDA = CFG

Rachit Nimavat

September 16, 2025

Outline

1. Recap
2. Defn: PDA
3. CFG $=$ PDA
4. CFG to PDA

Defn: CFG

A CFG is a set of recursive rules used to generate patterns of strings. Think of it as a blueprint for a language.

Definition 1.1

A Context-Free Grammar is a 4-tuple $G = (V, T, P, S)$, where:

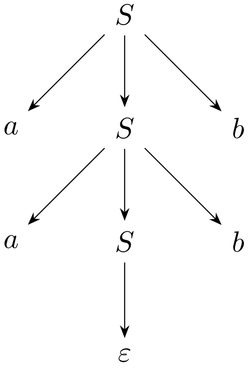
- **V**: Finite set of **variables**. think: syntactic categories or placeholders. (e.g., S, E, T in previous examples)
- **T**: Finite set of **terminals** or **tokens**. actual symbols or words of the language. (e.g., a, b, c, +, $\star$ in previous examples)
- **P**: Finite set of **production rules**. Rules have the form $A \to \alpha$, where $A \in V$ and $\alpha \in (V \cup T)^*$, i.e., α is a string consisting of variables and terminals. specifies how to replace variables with strings of variables and terminals. (e.g., $S \to \mathrm{a}S\mathrm{b}$ and $E \to T$) in previous examples
- **S**: The **start symbol** ($S \in V$).

Visualizing Derivations: Parse Trees

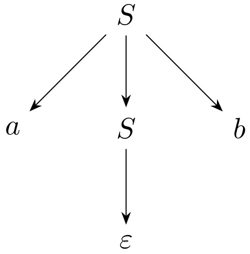
Similar to DFA, wer have **Parse Trees** to visualize derivation in CFGs.
Consider $L = \{a^n b^n \mid n \geq 0\}$ over $\Sigma = \{a, b\}$ and CFG $S \to \varepsilon \mid \mathrm{a}S\mathrm{b}$.

Parse tree for ε:

The parse tree for $aabb$:



Parse tree for ab:



Simplifying CFGs

Useless Productions and Symbols

Standardize CFGs! This makes grammars easier to work with, both for humans and for algorithms

Non-Generating Symbols. A variable A is non-generating if it can never derive a string of only terminals. It's a dead end.

Unreachable Symbols. A variable A is unreachable if there is no derivation from the start variable S that uses A.

Chomsky Normal Form (CNF)

Goal: Convert any CFG to a standard form without changing its language.

Definition 1.2

A CFG in CNF has production rules of the form:

- $A \to BC$ Variable yields two variables. Note B or C might be equal or equal to A.
- $A \to a$ Variable yields a single terminal.
- Does not have any useless symbols.
- The empty string ε is not allowed. If $\varepsilon \in L$, add a special rule $S \to \varepsilon$.

Benefits:

- Clean and Standard grammar
- No messy edge-cases, only two types of rules
- Great for theoretical analysis and writing algorithms

CFG to CNF Example

Balanced parenthesis: $S \rightarrow (\,) \,\big|\, (\,S\,) \,\big|\, SS$

On Board!

Equivalent of a DFA for CFGs?

Benefits of Hindsight

Consider a CFG S $\rightarrow$ ( ) | (S) | SS of the language L of balanced parentheses.
How to build a machine that accepts L?

```python
# Returns true if w is in the language L
def recognize(w):
    stack = []
    for c in w:
        if c == '(': stack.append(c)
        elif c == ')':
            if not len(stack):  return False
            stack.pop()
    return not len(stack)
```

Stripping the Syntax

- Read-only, one-way input (just like an NFA)
- Finite states: While no explicit state variable, we go in “reject state” once the stack is empty
- The stack: New component!
- Transition rules: What to do based on the current input symbol and the top of the stack

This machine (NFA + stack) is a Pushdown Automaton (PDA)!

Pushdown Automaton (PDA): Formal Definition

Definition 2.1

PDA is a 7-tuple: $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where:

- Q is a finite set of states
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accepting states
- Σ is the alphabet
- Γ is the stack alphabet
- $Z_0 \in \Gamma$ is the initial stack symbol
- δ is the transition function
  - $\delta : Q \times \Sigma \times \Gamma \to \mathcal{P}_{fin}(Q \times \Gamma^*)$ is the transition function
  - $\delta(q, a, x)$ is a finite set of choices for when we are in state q, read a, and the top of the stack is x: tells us which state to move to, and what stack alphabet symbols to replace x with in the stack

Example: Balanced Square Brackets

$$\mathsf{S} \to \varepsilon \,\big|\, [\,\mathsf{S}\,] \,\big|\, \mathsf{SS}$$

States $Q = \{q_0, q_f\}$ where q_0 is the start state and q_f is the only accepting state
$\Sigma = \{[\,,\,]\}$
$\Gamma = \{Z_0, [\,\}$ where Z_0 is the initial stack symbol
Transition function δ:

Push in the stack when we see a [:

$\delta(q_0, [\,, Z_0) = \{(q_0, Z_0[\,)\}$
$\delta(q_0, [\,, [\,) = \{(q_0, [\,[\,)\}$

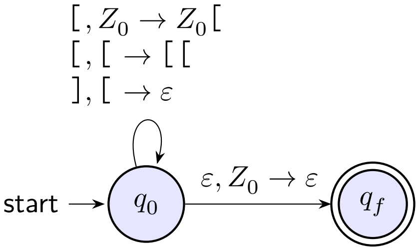
Pop from the stack when we see a], if possible:

$\delta(q_0,]\,, [\,) = \{(q_0, \varepsilon)\}$

Accept the string if we reach the end of input and the stack has only Z_0:

$\delta(q_0, \varepsilon, Z_0) = \{(q_f, \varepsilon)\}$

Example Diagram: Balanced Square Brackets



HW: Draw PDA for Balanced Square Brackets that does not accept ε

HW: There's another way to define PDA Acceptance: PDA accepts if it ends in empty stack (regardless of state). See Chapter 6.2 of ALC to see its equivalence with our accepting state definition.

Example: Even Palindromes

$$\mathsf{S} \rightarrow \varepsilon \,\big|\, \mathsf{aSa} \,\big|\, \mathsf{bSb}$$

States $Q = \{q_0, q_1, q_f\}$ where q_0 is the start state and q_f is the only accepting state
$\Sigma = \{\mathsf{a}, \mathsf{b}\}$
$\Gamma = \{Z_0, \mathsf{a}, \mathsf{b}\}$ where Z_0 is the initial stack symbol
Transition function δ:

Stack Push:

For all $c_1 \in \{\mathsf{a}, \mathsf{b}\}$ and $c_2 \in \{\mathsf{a}, \mathsf{b}, Z_0\}$: $\delta(q_0, c_1, c_2) = \{(q_0, c_2 c_1)\}$

Spontaneously move to state q_1 to start popping:

For all $c \in \{\mathsf{a}, \mathsf{b}, Z_0\}$: $\delta(q_0, \varepsilon, c) = \{(q_1, c)\}$

Stack Pop:

For all $c \in \{\mathsf{a}, \mathsf{b}\}$: $\delta(q_1, c, c) = \{(q_1, \varepsilon)\}$

Accept the string if we reach the end of input and the stack has only Z_0:

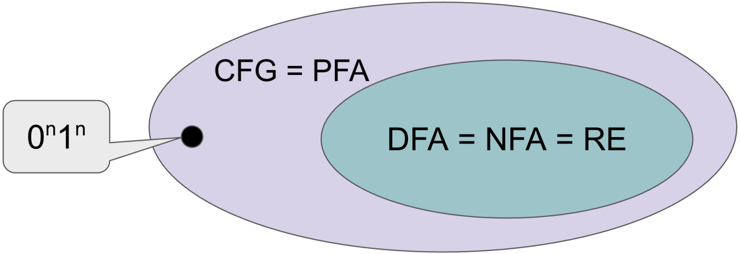
$\delta(q_1, \varepsilon, Z_0) = \{(q_f, \varepsilon)\}$

HW: Construct PDA and draw diagram for all Palindromes

Like Automata Trinity, we have another equivalence:

Theorem 3.1

A language is context-free if and only if it is recognized by a pushdown automaton.



Converting CFG to a PDA

Consider a CFG $G = (V, T, P, S)$. Construct a PDA M as follows:

States $Q = \{q_0, q_f\}$ where q_0 is the start state and q_f is the only accepting state

$\Sigma = T$

$\Gamma = V \cup T \cup \{Z_0\}$ where Z_0 is the initial stack symbol

Transition function δ:

For each variable $A \in V$,

$\delta(q_0, \varepsilon, A) = \{(q_0, \alpha) \mid \text{for each production rule } A \to \alpha \text{ of } P\}$

For each terminal $a \in T$,

$\delta(q_0, a, a) = \{(q_0, \varepsilon)\}$

For when the stack is empty (except for Z_0) and input is fully read,

$\delta(q_0, \varepsilon, Z_0) = \{(q_f, \varepsilon)\}$

HW: Prove that the above PDA M accepts the same language as the CFG G.