L4.1 Equivalence of CFG and PDA (Part 1)

Equivalence of CFG and PDA (Part-1)

Theorem: A language is Context Free iff some Pushdown Automata recognizes it.

Proof: Part 1: Given a CFG, show how to construct a PDA that recognizes it.

Part 2: Given a PDA, show how to construct a CFG that recognizes the same language.

Equivalence of CFG and PDA (Part-1)

(From CFG to PDA)

Theorem: A language is Context Free iff some Pushdown Automata recognizes it.

Proof: Part 1: Given a CFG, show how to construct a PDA that recognizes it.

Part 2: Given a PDA, show how to construct a CFG that recognizes the same language.

Given a grammar

$$S \rightarrow BS \mid A$$

$$A \rightarrow 0A \mid \epsilon$$

$$B \rightarrow BB1 \mid 2$$

Find or build a PDA

Equivalence of CFG and PDA (Part-1)

(From CFG to PDA)

Theorem: A language is Context Free iff some Pushdown Automata recognizes it.

Proof: Part 1: Given a CFG, show how to construct a PDA that recognizes it.

Part 2: Given a PDA, show how to construct a CFG that recognizes the same language.

Given a grammar

S→ BS|A
A→0A|∈
B→BB1|2

Find or build a PDA

--> S
--> BS
--> BB1S
--> 2B1S
--> 221S
--> 221A
--> 221∈
--> 221

(Left most derivation)

Equivalence of CFG and PDA (Part-1)

(From CFG to PDA)

Theorem: A language is Context Free iff some Pushdown Automata recognizes it.

Proof: Part 1: Given a CFG, show how to construct a PDA that recognizes it.

Part 2: Given a PDA, show how to construct a CFG that recognizes the same language.

Given a grammar

$S \rightarrow BS|A$

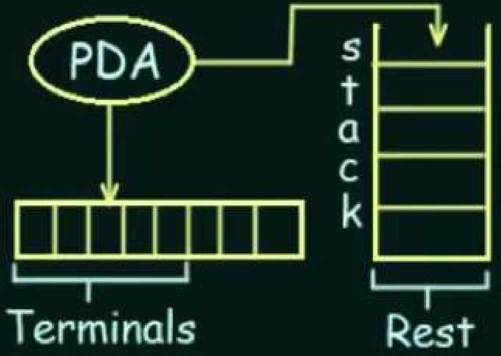
$A \rightarrow 0A|\epsilon$

$B \rightarrow BB1|2$

Find or build a PDA

--> S
--> BS
--> BB1S
--> 2B1S
--> 221S
--> 221A
--> 221∈
--> 221

(Left most derivation)



General Form:

$$\underbrace{aaaa}_{\text{Terminals}}\ \underbrace{BaBC}_{\text{Rest}}$$

Equivalence of CFG and PDA (Part-1)

(From CFG to PDA)

Theorem: A language is Context Free iff some Pushdown Automata recognizes it.

Proof: Part 1: Given a CFG, show how to construct a PDA that recognizes it.

Part 2: Given a PDA, show how to construct a CFG that recognizes the same language.

Given a grammar

$S \to BS|A$

$A \to 0A|\epsilon$

$B \to BB1|2$

Find or build a PDA

--> S

--> BS

--> BB1S

--> 2B1S

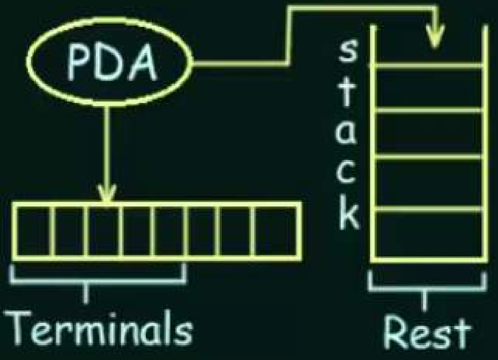
--> 221S

--> 221A

--> 221∊

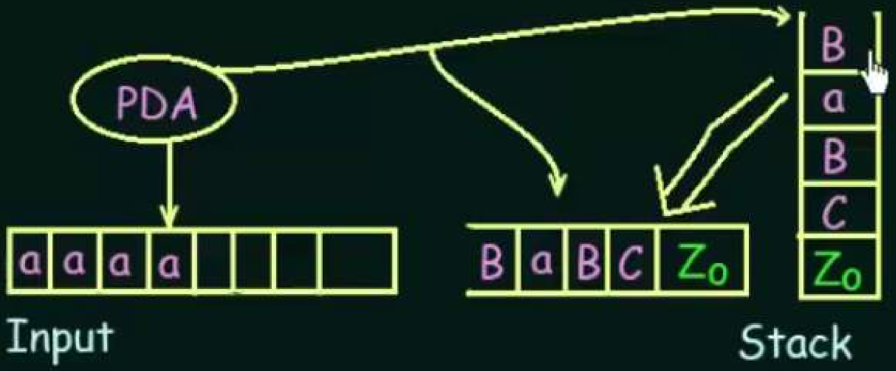
--> 221

(Left most derivation)



General Form:

$$\underbrace{aaaa}_{\text{Terminals}}\ \underbrace{BaBC}_{\text{Rest}}$$

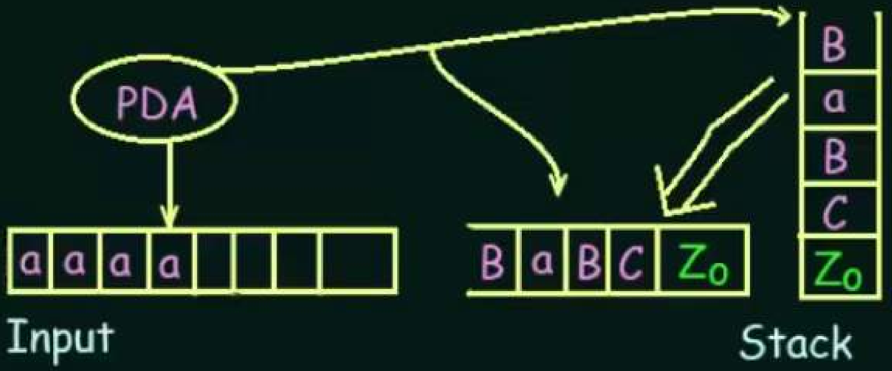


Left Most Derivation: S --> ..... a a a a B a B C --> ......

AT EACH STEP EXPAND LEFT-MOST DERIVATION

Eg. Rule: B→ASAxBA --> ..... a a a a A S A x B A a B C

- → Match Stack Top to a Rule
- → Pop Stack
- → Push Right Hand Side of Rule onto Stack



Left Most Derivation: S --> ..... a a a a B a B C --> ......

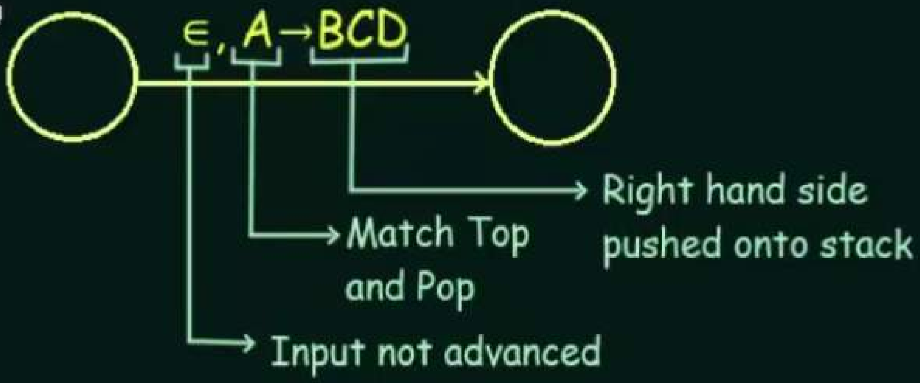
AT EACH STEP EXPAND LEFT-MOST DERIVATION

Eg. Rule: B→ASAxBA --> ..... a a a a A S A x B A a B C

- → Match Stack Top to a Rule
- → Pop Stack
- → Push Right Hand Side of Rule onto Stack

Rule: $A \rightarrow BCD$

Add this to the PDA



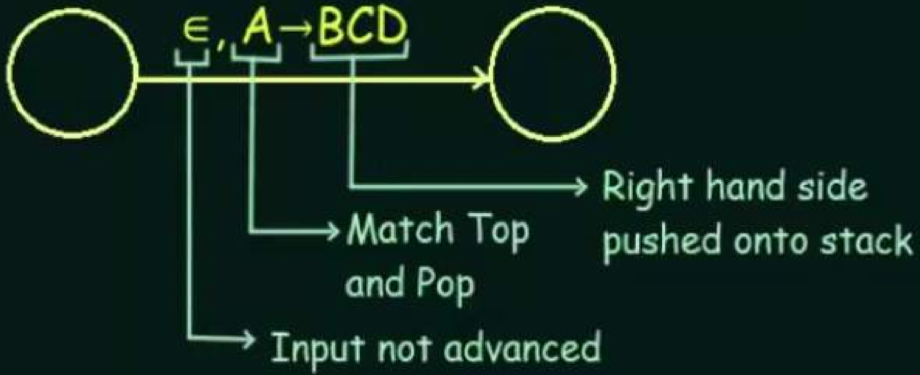
Right hand side pushed onto stack

Match Top and Pop

Input not advanced

Rule: A → BCD

Add this to the PDA



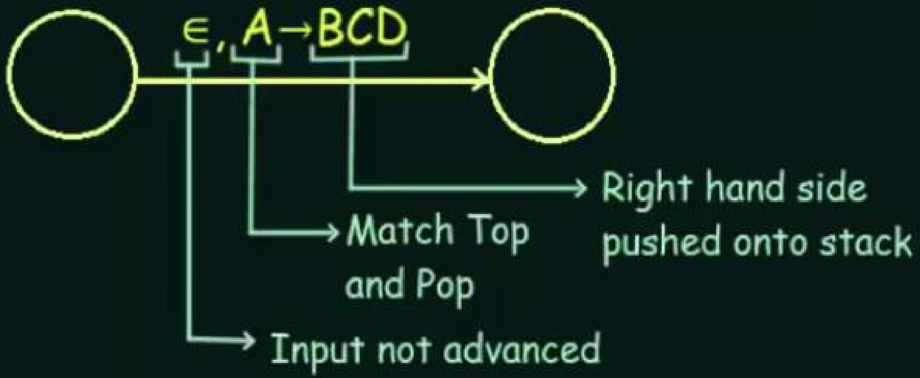
Right hand side pushed onto stack

Match Top and Pop

Input not advanced

Rule: A → BCD

Add this to the PDA

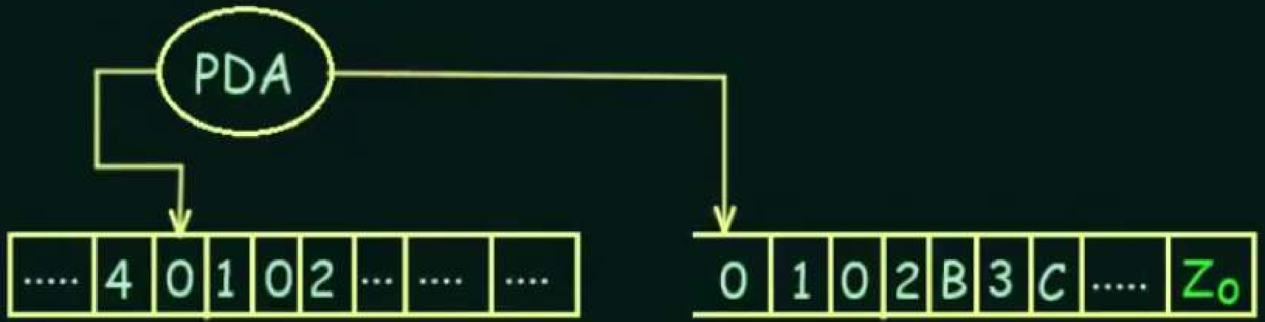


Right hand side pushed onto stack

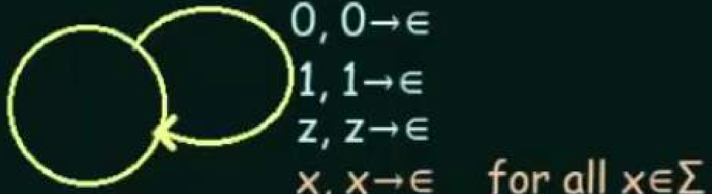
Match Top and Pop

Input not advanced

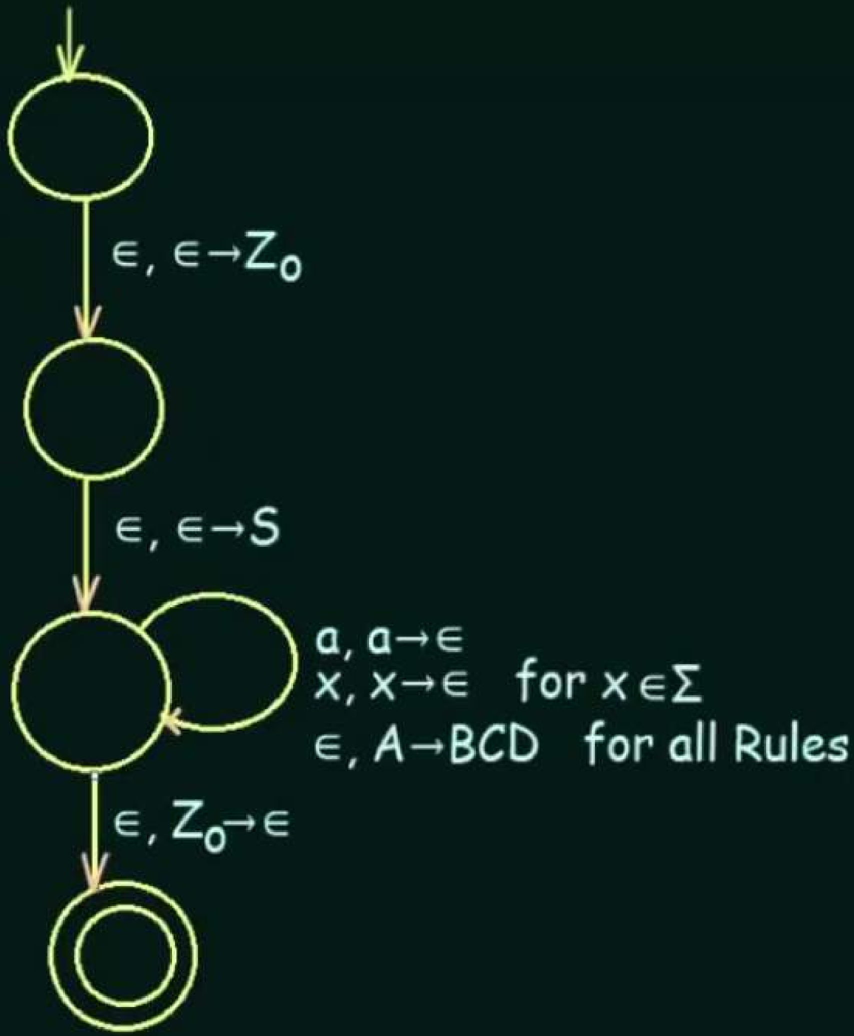
Rule: $A \rightarrow 0\ 1\ 0\ 2\ B\ 3\ C$



MATCH TERMINAL SYMBOLS TO THE STACK TOP



The Final PDA



The Final PDA