(DEEMED TO BE UNIVERSITY)

# 23MT2014

# THEORY OF COMPUTATION

Topic:

# POSITIVE PROPERTIES OF CONTEXT-FREE LANGUAGE

Session - 1

# AIM OF THE SESSION

The aim of the session is to introduce participants to the positive properties of Context-Free Languages (CFLs) and their significance in computer science and formal language theory.

# INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Understand and identify the positive properties of CFLs, such as closure under concatenation, closure under union, and the ability to generate nested structures.

# LEARNING OUTCOMES

At the end of this session, you should be able to:

1. recognize and analyze the positive properties of CFLs in different contexts, apply them to construct grammars and parse trees, and comprehend their relevance in programming language design and compiler construction.

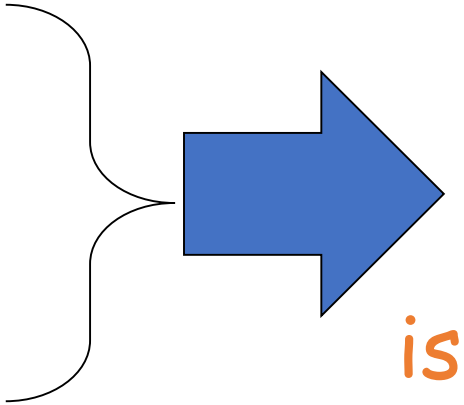
# Positive Properties of Context-Free languages

# Union

Context-free languages
are closed under: **Union**

$L_1$ is context free

$L_2$ is context free



$L_1 \cup L_2$ is context-free

# Example

## Language

$$L_1 = \{a^n b^n\}$$

$$L_2 = \{ww^R\}$$

## Grammar

$$S_1 \rightarrow aS_1b \mid \lambda$$

$$S_2 \rightarrow aS_2a \mid bS_2b \mid \lambda$$

## Union

$$L = \{a^n b^n\} \cup \{ww^R\}$$

$$S \rightarrow S_1 \mid S_2$$

# In general:

For context-free languages $L_1,\ L_2$

with context-free grammars $G_1,\ G_2$

and start variables $S_1,\ S_2$

The grammar of the **union** $L_1 \cup L_2$

has new start variable $S$

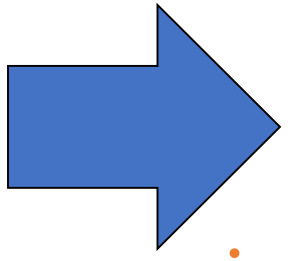
and additional production $S \to S_1 \mid S_2$

# Concatenation

Context-free languages are closed under: **Concatenation**

$L_1$ is context free

$L_2$ is context free



$L_1 L_2$ is context-free

# Example

## Language

$$L_1 = \{a^n b^n\}$$

$$L_2 = \{ww^R\}$$

## Grammar

$$S_1 \rightarrow aS_1b \mid \lambda$$

$$S_2 \rightarrow aS_2a \mid bS_2b \mid \lambda$$

## Concatenation

$$L = \{a^n b^n\}\{ww^R\}$$

$$S \rightarrow S_1 S_2$$

# In general:

For context-free languages $L_1,\ L_2$

with context-free grammars $G_1,\ G_2$

and start variables $S_1,\ S_2$

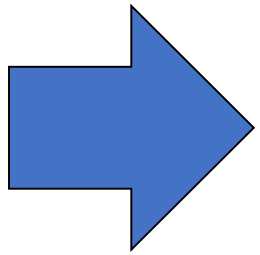
The grammar of the **concatenation** $L_1 L_2$

has new start variable $S$

and additional production $S \to S_1 S_2$

# Star Operation

Context-free languages
are closed under: **Star-operation**

$L$ is context free  $L^*$ is context-free

# Example

Language

$$L = \{a^n b^n\}$$

Grammar

$$S \rightarrow aSb \mid \lambda$$

## Star Operation

$$L = \{a^n b^n\}*$$

$$S_1 \rightarrow SS_1 \mid \lambda$$

# In general:

For context-free language $L$

with context-free grammar $G$

and start variable $S$

The grammar of the **star operation** $L^*$

has new start variable $S_1$

and additional production $S_1 \rightarrow SS_1 \mid \lambda$

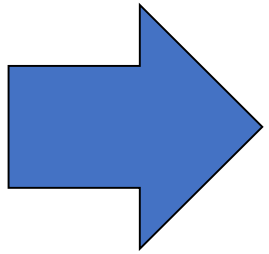
# Negative Properties of Context-Free Languages

# Intersection

Context-free languages are **not** closed under: **intersection**

$L_1$ is context free

$L_2$ is context free



$L_1 \cap L_2$

**not** necessarily context-free

# Example

$$L_1 = \{a^n b^n c^m\}$$

Context-free:

$$S \rightarrow AC$$

$$A \rightarrow aAb \mid \lambda$$

$$C \rightarrow cC \mid \lambda$$

$$L_2 = \{a^n b^m c^m\}$$

Context-free:

$$S \rightarrow AB$$

$$A \rightarrow aA \mid \lambda$$

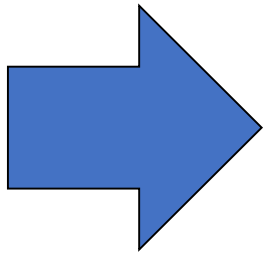
$$B \rightarrow bBc \mid \lambda$$

**Intersection**

$$L_1 \cap L_2 = \{a^n b^n c^n\}$$ **NOT** context-free

# Complement

Context-free languages are **not** closed under: **complement**

$L$ is context free  $\overline{L}$ **not** necessarily context-free

# Example

$$L_1 = \{a^n b^n c^m\}$$

Context-free:

$$S \to AC$$

$$A \to aAb \mid \lambda$$

$$C \to cC \mid \lambda$$

$$L_2 = \{a^n b^m c^m\}$$

Context-free:

$$S \to AB$$

$$A \to aA \mid \lambda$$

$$B \to bBc \mid \lambda$$

**Complement**

$$\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2 = \{a^n b^n c^n\}$$

**NOT** context-free

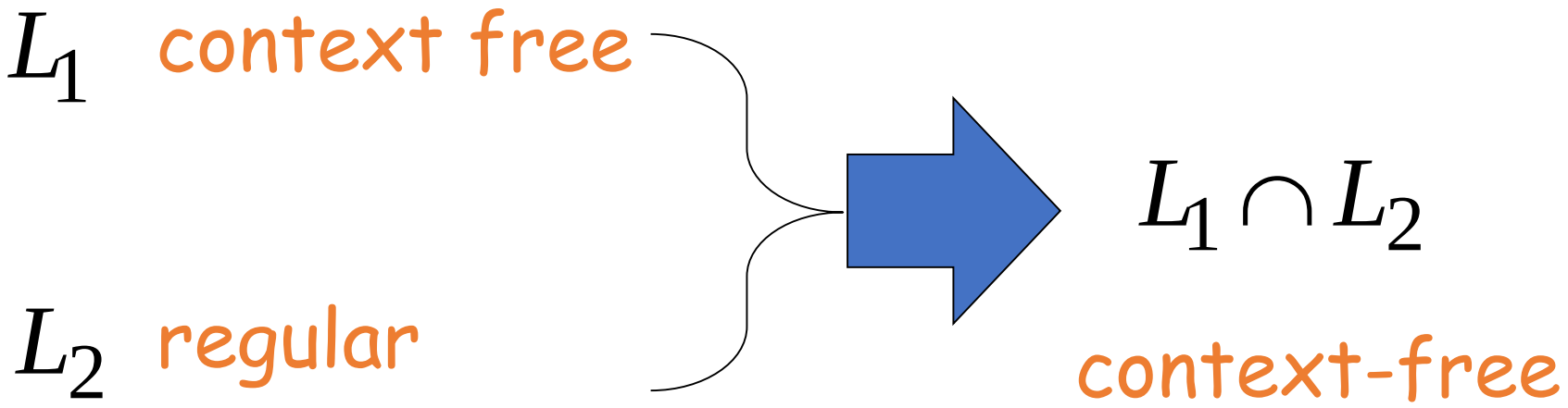
# Intersection of Context-free languages and Regular Languages

The intersection of

a context-free language and

a regular language

is a context-free language



Machine $M_1$

NPDA for $L_1$

context-free

Machine $M_2$

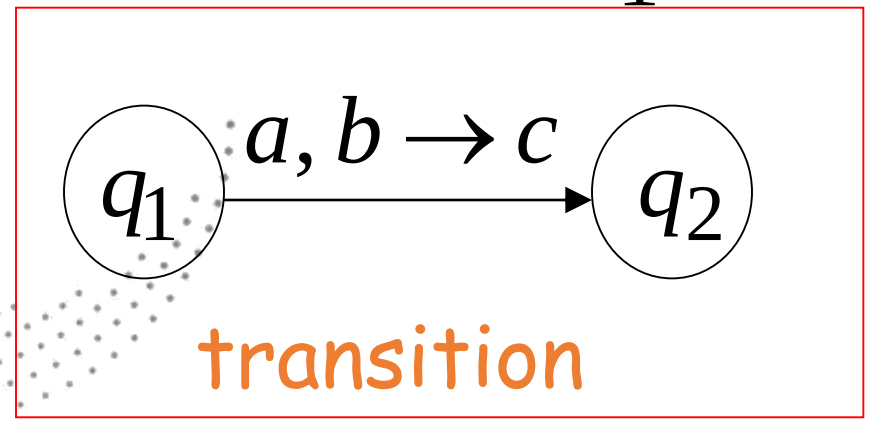
DFA for $L_2$

regular

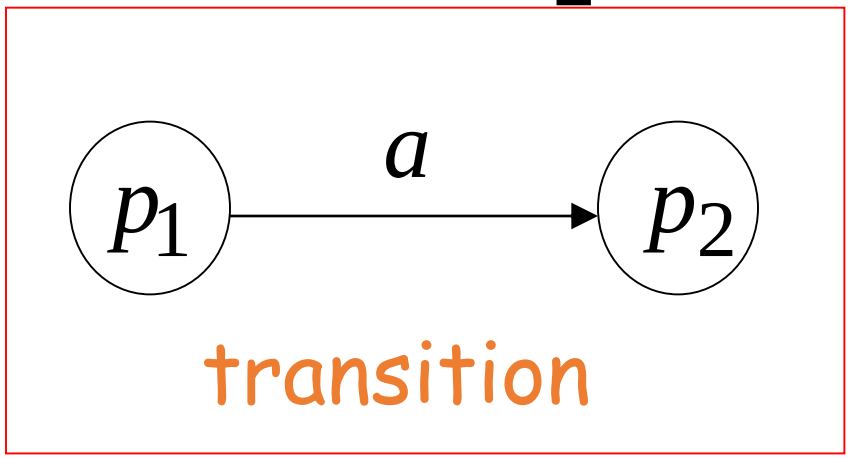
Construct a new NPDA machine $M$ that accepts $L_1 \cap L_2$

$M$ simulates in parallel $M_1$ and $M_2$

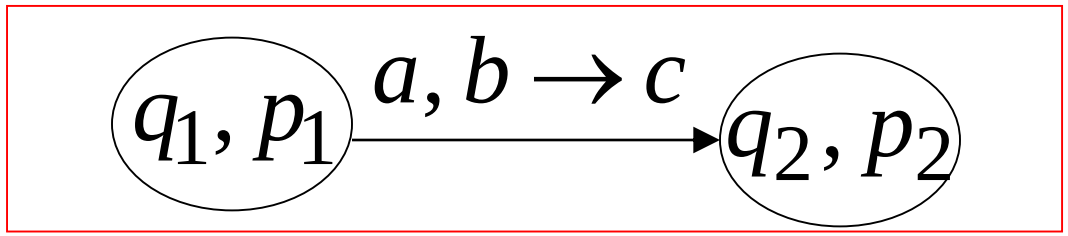
NPDA $M_1$



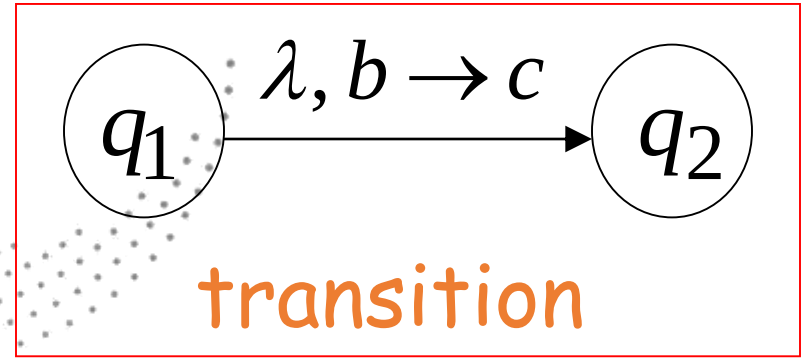

DFA $M_2$




NPDA $M$

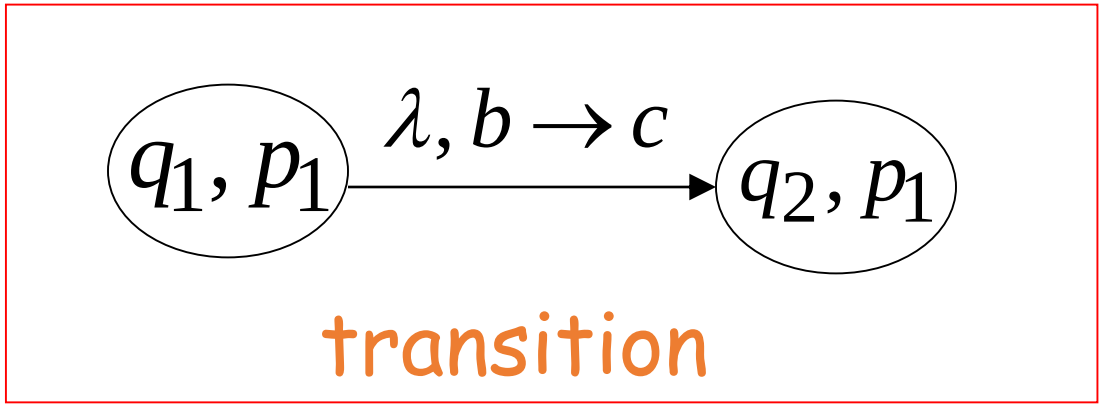


NPDA $M_1$

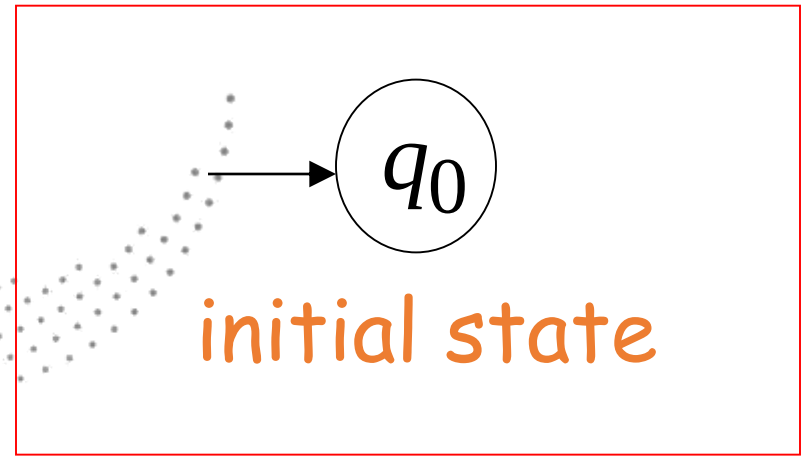



DFA $M_2$

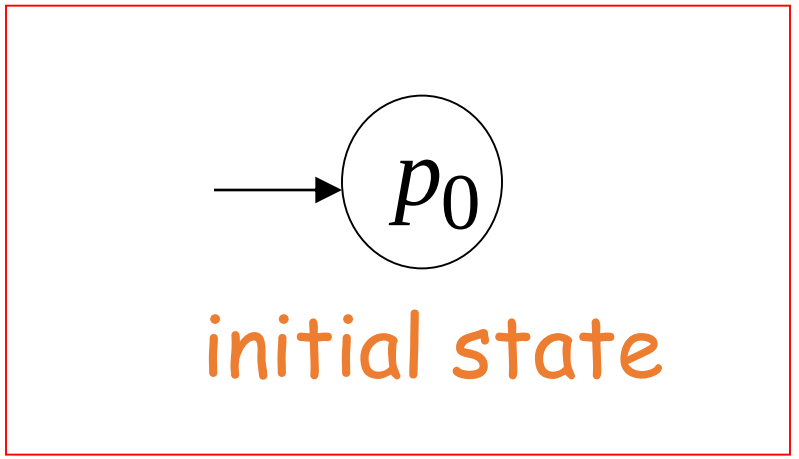

NPDA $M$



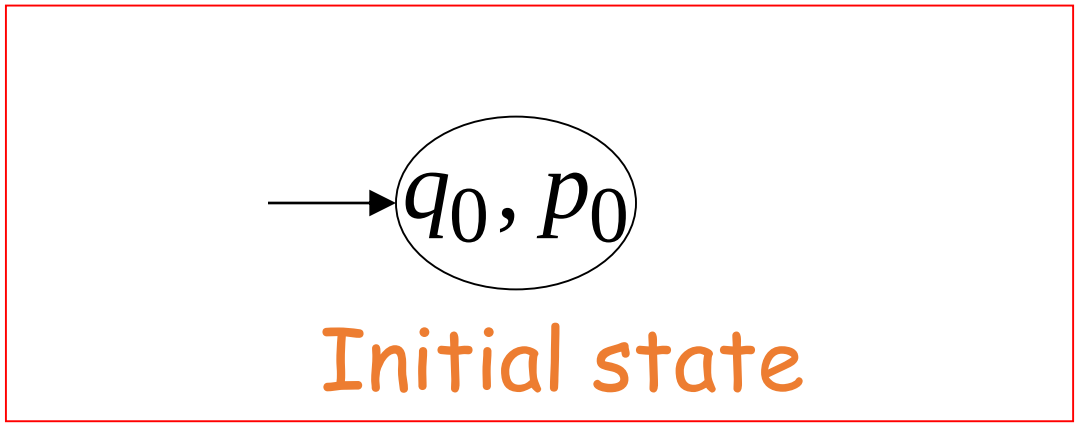
NPDA $M_1$



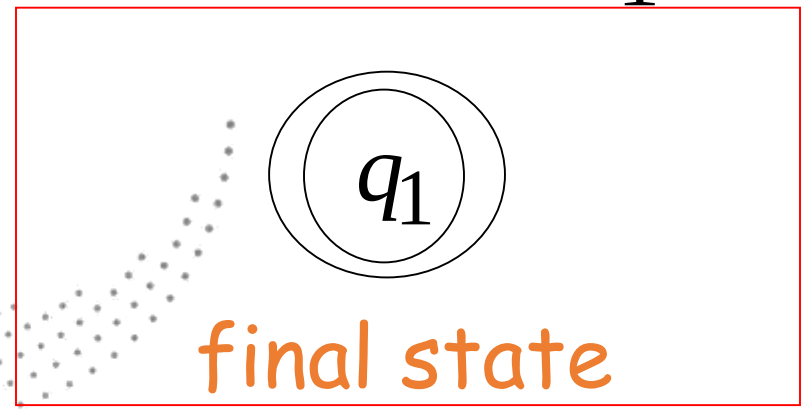

DFA $M_2$



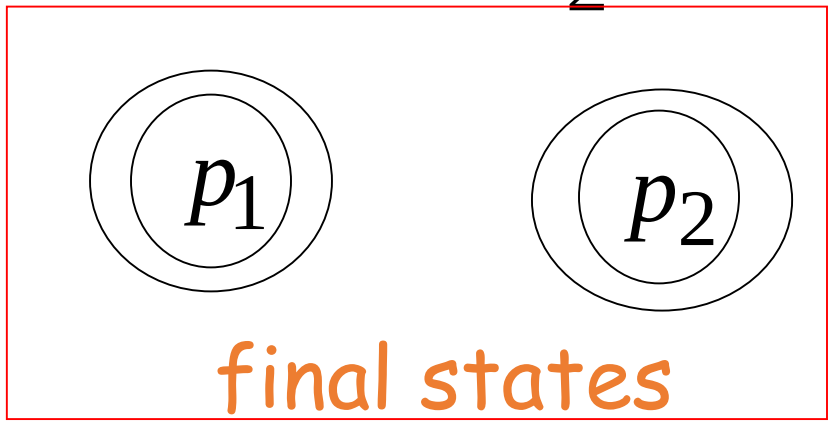

NPDA $M$



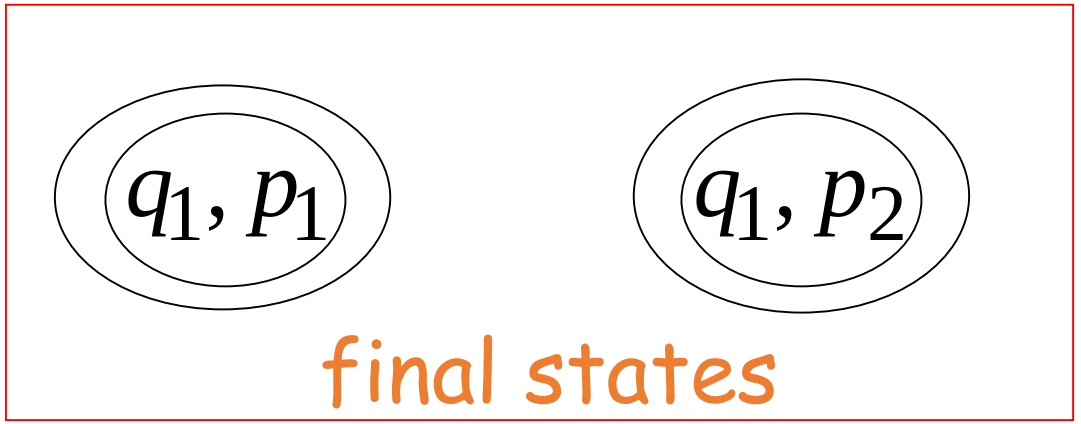
NPDA $M_1$




DFA $M_2$




NPDA $M$

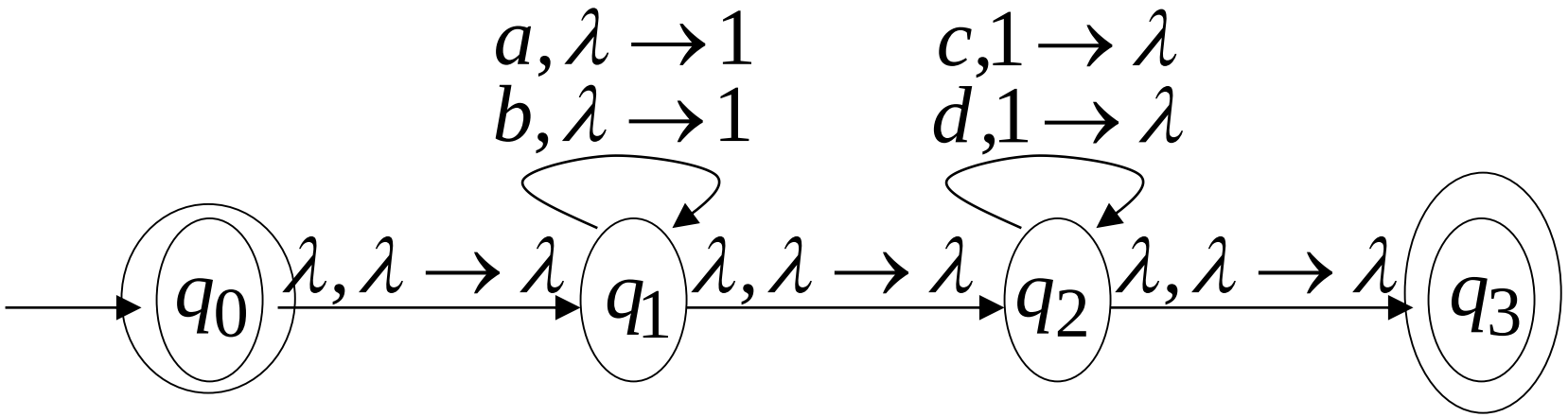


# Example:

context-free

$$L_1 = \{w_1 w_2 : |\, w_1 \,| = |\, w_2 \,|,\ w_1 \in \{a,b\}^*,\ w_2 \in \{c,d\}^*\}$$

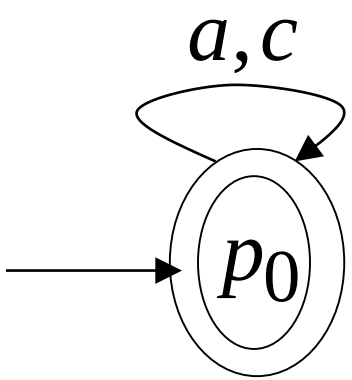
NPDA $M_1$



regular

$$L_2 = \{a, c\}^*$$

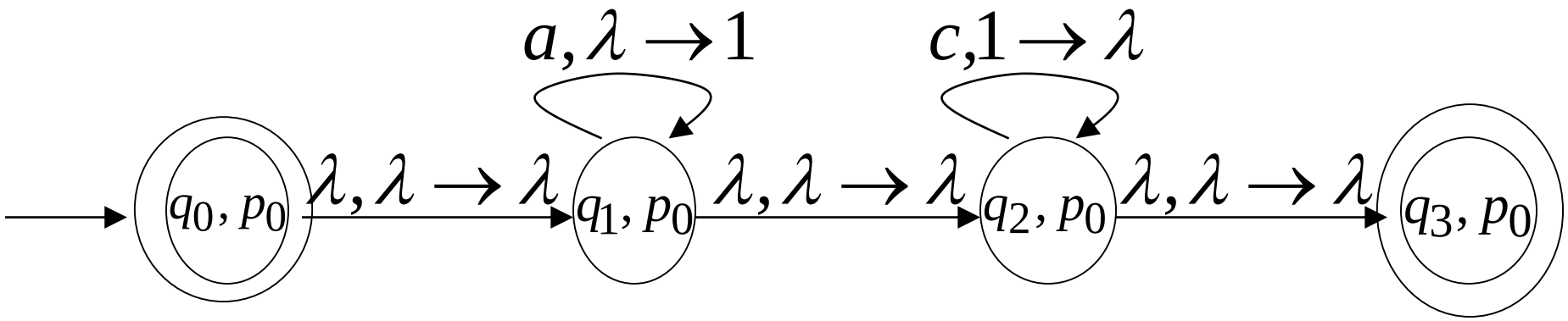
DFA $M_2$



context-free

Automaton for: $L_1 \cap L_2 = \{a^n c^n : n \geq 0\}$

NPDA $M$



# In General:

$M$ simulates in parallel $M_1$ and $M_2$

$M$ accepts string $w$ if and only if

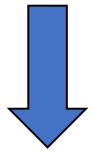
$M_1$ accepts string $w$ and

$M_2$ accepts string $w$

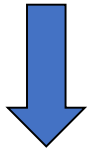
$$L(M) = L(M_1) \cap L(M_2)$$

# Therefore:

$M$ is NPDA



$L(M_1) \cap L(M_2)$ is context-free



$L_1 \cap L_2$ is context-free

# SELF-ASSESSMENT QUESTIONS

Q.1. Which of the following is a positive property of Context-Free Languages?

A) Inability to generate palindromes

B) Closure under union

C) Limited expressive power compared to Regular Languages

D) Lack of nested structures

# SELF-ASSESSMENT QUESTIONS

Q.2. Which of the following is a negative property of Context-Free Languages?

A) Ability to generate palindromes

B) Closure under intersection

C) Ability to recognize Regular Languages

D) Lack of recursion

# SELF-ASSESSMENT QUESTIONS

Q.3. Which property makes Context-Free Languages suitable for designing programming languages?

A) Closure under intersection

B) Lack of nesting

C) Closure under complement

D) Closure under concatenation

# TERMINAL QUESTIONS

Q.1 Question 1: Define a Context-Free Language (CFL) in your own words.

Question 2: Explain one positive property of Context-Free Languages and provide an example.

Question 3: Describe a negative property or limitation of Context-Free Languages.

Question 4: How are Context-Free Languages relevant in the field of compiler construction?

# THANK YOU


## Team – TOC