INT201 Decision, Computation and Language

Lecture 6 – Context-Free Languages (1)

Dr Yushi Li

Xi'an Jiaotong-Liverpool University
西交利物浦大学

Context-Free Languages

- Context-Free Grammar (CFG)
- Chomsky Normal Form (CNF)

Context-Free Languages

- **Finite automata** accept precisely the strings in the language.

  Perform a computation to determine whether a specific string is in the language.

- **Regular expressions** describe precisely the strings in the language

  Describe the general shape of all strings in the language.

- **Context-free grammar (CFG)** is an entirely different formalism for defining a class of languages.

  Give a procedure for listing off all strings in the language.

Context-Free Languages

Applications of CFG

- Programming languages: CFGs are used to define the syntax of programming languages, allowing parsers to analyze code structure.
- NLP: CFGs help in parsing sentences, enabling applications like machine translation and speech recognition
- Compilers: CFGs facilitate syntax analysis, ensuring that the source code adheres to the language's grammatical rules.

Context-Free Grammar

Example

- Start variable S with rules:
  $S \rightarrow AB$
  $A \rightarrow a$
  $A \rightarrow aA$
  $B \rightarrow b$
  $B \rightarrow bB$

variables: S, A, B terminals: a, b

- Following these rules, we can yield ?

Context-Free Grammar

Definition

A context-free grammar is a 4-tuple $G = (V, \Sigma, R, S)$, where

1. V is a finite set, whose elements are called **variables**,
2. Σ is a finite set, whose elements are called **terminals**,
3. $V \cap \Sigma = \emptyset$,
4. S is an element of V ; it is called the **start variable**,
5. R is a finite set, whose elements are called **rules**. Each rule has the form $A \to w$, where $A \in V$ and $w \in (V \cup \Sigma)^*$.

Context-Free Grammar

Example

Language $L = \{0^k 0^k : k \geq 0\}$ has CFG $G = (V, \Sigma, R, S)$,

Deriving strings and languages using CFG

$\Rightarrow$: **yeild**

Let $G = (V, \Sigma, R, S)$ be a context free grammar with

- $A \in V$
- $u, v, w \in (V \cup \Sigma)^*$,
- $A \rightarrow w$ is a rule of the grammar

The string uwv can be derived in one step from the string uAv, written as

$$uAv \Rightarrow uwv$$

Example: $aaAbb \Rightarrow aaaAbb$

Deriving strings and languages using CFG

$\overset{*}{\Rightarrow}$: **derive**

Let $G = (V, \Sigma, R, S)$ be a context free grammar with

- $u, v \in (V \cup \Sigma)^*$

The string v can be derived from the string u, written as $u \overset{*}{\Rightarrow} v$, if one of the following conditions holds:

1. $u = v$
2. there exist an integer $k \geq 2$ and a sequence $u_1, u_2, \ldots, u_k$ of strings in $(V \cup \Sigma)^*$, such that

(a) $u = u_1$,

(b) $v = u_k$, and $u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k$.

Example: With the rules $A \rightarrow B1 \mid D0C$

$$0AA \overset{*}{\Rightarrow} 0D0CB1$$

Language of CFG

Definition

The language of CFG $G = (V, \Sigma, R, S)$ is

$$L(G) = \{ w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w \}.$$

Such a language is called **context-free**, and satisfies $L(G) \subseteq \Sigma^*$.

Example

CFG $G = (V, \Sigma, R, S)$ with

1. $V = \{S\}$
2. $\Sigma = \{0, 1\}$
3. Rules R: $S \to 0S \mid \varepsilon$

$L(G) = ?$

Example (Palindrome)

CFG $G = (V, \Sigma, R, S)$ with

1. $V = \{S\}$
2. $\Sigma = \{a, b\}$
3. Rules R: $S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon$

Language of this CFG ?

Example (Simple Arithmetic Expressions)

CFG $G = (V, \Sigma, R, S)$ with

1. $V = \{S\}$
2. $\Sigma = \{+, -, \times, /, (,), 0, 1, 2, \ldots, 9\}$
3. Rules R:

   $S \to S + S \mid S - S \mid S \times S \mid S/S \mid (S) \mid -S \mid 0 \mid 1 \mid \cdot \quad \cdot \quad \cdot \mid 9$

$L(G)$: valid arithmetic expressions over single-digit integers

S derives string $3 \times (5 + 6)$?

Regular Languages are context-free

Theorem

Let Σ be an alphabet and let $L \subseteq \Sigma^*$ be a regular language. Then L is a context-free language (Every regular language is context-free).

Proof

Since L is a regular language, there exists a deterministic finite automaton $M = (Q, \Sigma, \delta, q, F)$ that accepts L. To prove that L is context-free, we have to define a context-free grammar $G = (V, \Sigma, R, S)$, such that $L = L(M) = L(G)$. Thus, G must have the following property:

For every string $w \in \Sigma^*$,

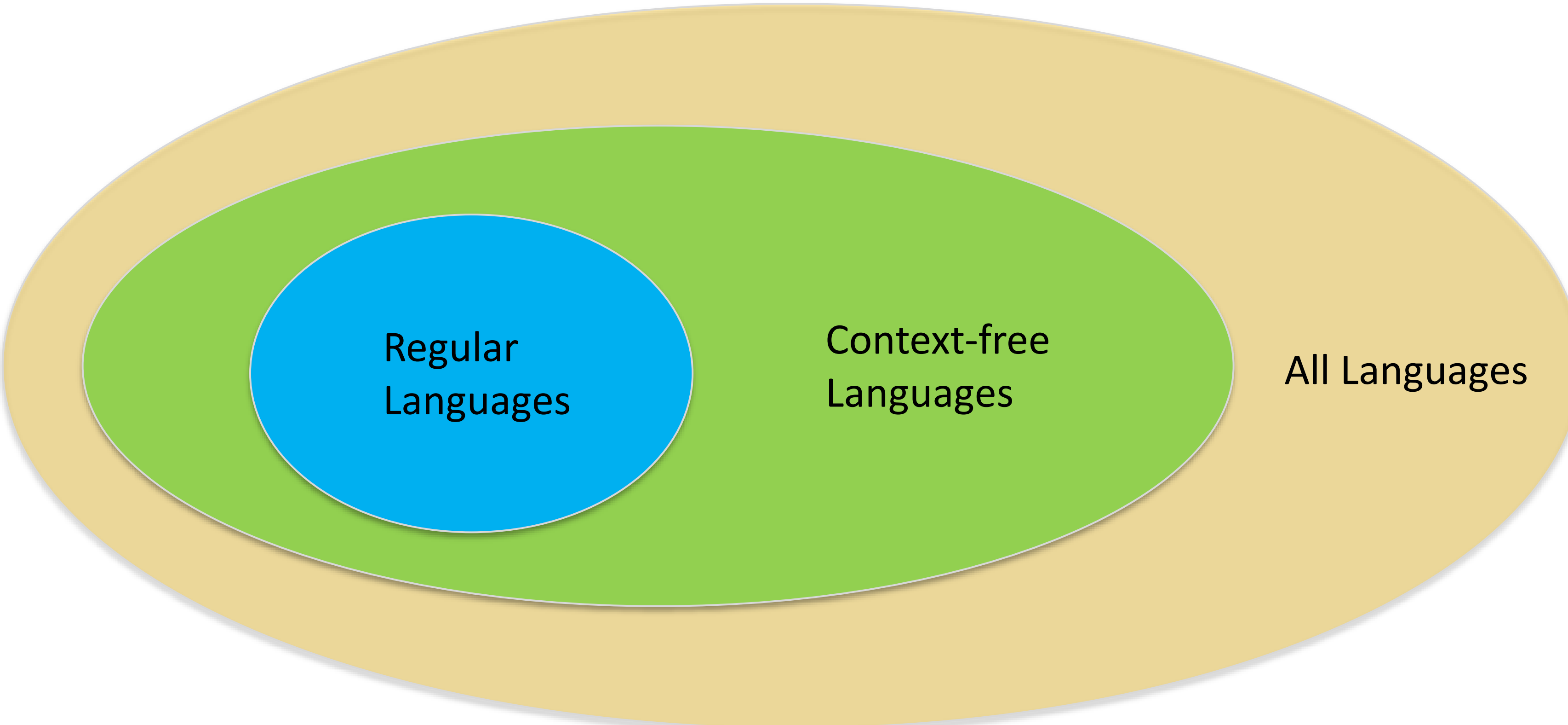
$w \in L(M)$ if and only if $w \in L(G)$,

which can be reformulated as

M accepts w if and only if $S \overset{*}{\Rightarrow} w$.

Set $V = \{R_i \mid q_i \in Q\}$ (that is, G has a variable for every state of M). Now, for every transition $\delta(q_i, a) = q_j$ add a rule $R_i \rightarrow aR_j$. For every accepting state $q_i \in F$ add a rule $R_i \rightarrow \varepsilon$. Finally, make the start variable $S = R_0$.

Regular Languages are context-free



Closure properties of CFLs: CFLs are closed under operations like union and concatenation but not under intersection or complementation.

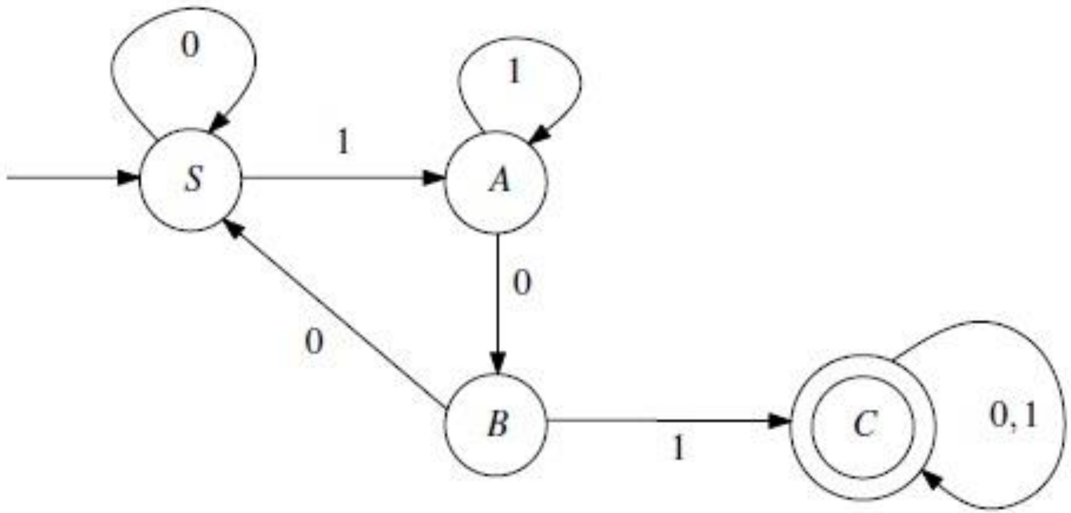
Regular Languages are context-free

Example

Let L be the language defined as

$$L = \{w \in \{0, 1\}^*: 101 \text{ is a substring of } w\}.$$

The DFA M that accepts L



How can we convert M to a context-free grammar G whose language is L?

Regular Languages are context-free

Example

Chomsky Normal Form (CNF)

Definition

A context-free grammar $G = (V, \Sigma, R, S)$ is said to be in **Chomsky normal form**, if every rule in R has one of the following three forms:

- $A \rightarrow BC$, where A, B, and C are elements of V , $B \neq S$, and $C \neq S$.
- $A \rightarrow a$, where A is an element of V and a is an element of Σ.
- $S \rightarrow \varepsilon$, where S is the start variable.

Why CNF?

Grammars in Chomsky normal form are far easier to analyze.

Example

Rules of CFG in Chomsky normal form with $V = \{S, A, B\}$, $\Sigma = \{a, b\}$:

$G_1: S \rightarrow AB, S \rightarrow c, A \rightarrow a, B \rightarrow b$ (CNF)

$G_1: S \rightarrow aA, A \rightarrow a, B \rightarrow c$ (not CNF)

Chomsky Normal Form (CNF)

Theorem

Let Σ be an alphabet and let $L \subseteq \Sigma^*$ be a context-free language. There exists a context-free grammar in Chomsky normal form, whose language is L (Every CFL can be described by a CFG in CNF).

CFL $\rightarrow$ CNF

Given CFG $G = (V, \Sigma, R, S)$. Replace, one-by-one, every rule that is not "Chomsky".

- Start variable (not allowed on RHS of rules)
- ε-rules ($A \rightarrow \varepsilon$ not allowed when A isn't start variable)
- all other violating rules ($A \rightarrow B$, $A \rightarrow aBc$, $A \rightarrow BCDE$)

Converting CFG into CNF

Transformation steps

Step 1. Eliminate the start variable from the right-hand side of the rules.

- New start variable S_0
- New rule $S_0 \rightarrow S$

Step 2. Remove **ε-rules** $A \rightarrow \varepsilon$, where $A \in V - \{S\}$.

- Before: $B \rightarrow xAy$ and $A \rightarrow \varepsilon \mid \cdot \cdot \cdot$
- After: $B \rightarrow xAy \mid xy$ and $A \rightarrow \cdot \cdot \cdot$

When removing $A \rightarrow \varepsilon$ rules, insert all new replacements:

- Before: $B \rightarrow AbA$ and $A \rightarrow \varepsilon \mid \cdot \cdot \cdot$
- After: $B \rightarrow AbA \mid bA \mid Ab \mid b$ and $A \rightarrow \cdot \cdot \cdot$

Converting CFG into CNF

Transformation steps

Step 3. Remove **unit rules** $A \rightarrow B$, where $A \in V$.

- Before: $A \rightarrow B$ and $B \rightarrow xCy$
- After: $A \rightarrow xCy$ and $B \rightarrow xCy$

Step 4. Eliminate all rules having more than two symbols on the right-hand side.

- Before: $A \rightarrow B_1B_2B_3$
- After: $A \rightarrow B_1A_1$, $A_1 \rightarrow B_2B_3$

Step 5. Eliminate all rules of the form $A \rightarrow ab$, where a and b are not both variables.

- Before: $A \rightarrow ab$
- After: $A \rightarrow B_1B_2$, $B_1 \rightarrow a$, $B_2 \rightarrow b$.

Converting CFG into CNF

Example

Given a CFG $G = (V, \Sigma, R, S)$, where $V = \{A, B\}$, $\Sigma = \{0, 1\}$, A is the start variable, and R consists of the rules:

$$A \rightarrow BAB \mid B \mid \varepsilon$$
$$B \rightarrow 00 \mid \varepsilon$$

Convert this G to CNF:

Step 1. Eliminate the start variable from the right-hand side of the rules.

Converting CFG into CNF

Example

Step 2. Remove **ε-rules**.

(1) Remove $A \rightarrow \varepsilon$: $S \rightarrow A$, $A \rightarrow BAB$

(2) Remove $B \rightarrow \varepsilon$: $A \rightarrow BAB$, $A \rightarrow B$, $A \rightarrow BB$

Converting CFG into CNF

Example

Step 3. Remove **unit-rules**.

(1) Remove $A \rightarrow A$:

(2) Remove $S \rightarrow A$:

Converting CFG into CNF

Example

Step 3. Remove **unit-rules**.

(3) Remove $S \rightarrow B$:

(4) Remove $A \rightarrow B$:

Converting CFG into CNF

Example

Step 4. Eliminate all rules having more than two symbols on the right-hand side.

(1) Remove $S \rightarrow BAB$:

(2) Remove $A \rightarrow BAB$:

Converting CFG into CNF

Example

Step 5. Eliminate all rules, whose right-hand side contains exactly two symbols, which are not both variables.

(1) Remove $S \to 00$:

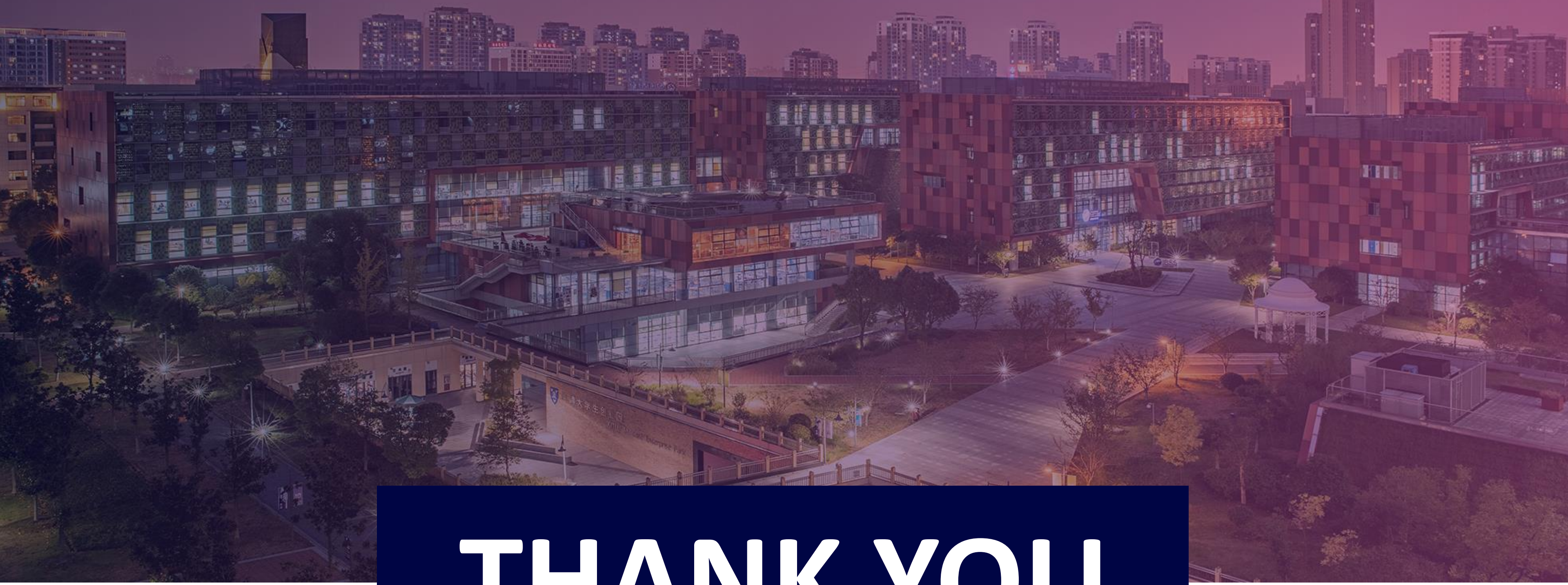
(1) Remove $A \to 00$:

Converting CFG into CNF

Example

Step 5. Eliminate all rules, whose right-hand side contains exactly two symbols, which are not both variables.

(3) Remove $S \rightarrow 00$:



THANK YOU

Xi'an Jiaotong-Liverpool University
西交利物浦大学

XJTLU | SCHOOL OF FILM AND TV ARTS