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

自动机 接受语言的字符串

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

  通过计算去判断输入的语言是否属于该语言

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

  正则化表达 描述语言的字符串

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

  Describe the general shape of all strings in the language.

  classify

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

  定义语言的种类

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

  提供一个步骤来列出语言中所有strings

Context-Free Languages 上下文无关语言

Applications of CFG

定义句法

- Programming languages: CFGs are used to define the syntax of programming languages, allowing parsers to analyze code structure. 分析代码框架

  分析程序

  自然语言处理

- NLP: CFGs help in parsing sentences, enabling applications like machine translation and speech recognition

  解析句子

  编译器 方便

- Compilers: CFGs facilitate syntax analysis, ensuring that the source code adheres to the language's grammatical rules.

  确保源代码贴合语言语法规则

Context-Free Grammar

Example

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

$L = \{a^m \mid b^m : m \geq 1\}$

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

- Following these rules, we can yield ?

We can infer a language from given rule

$S \Rightarrow AB \Rightarrow aAB \Rightarrow aAbB \Rightarrow aaAbB \Rightarrow$
$aaAbB \Rightarrow aaaAbbB \Rightarrow aaaa\,bbb.$
aaa bb
aa bbb ...

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**, 有限集合 变量
2. Σ is a finite set, whose elements are called **terminals**, 有限集合 末端 (注意和 DFA/NFA 的 Σ 区别)
3. $V \cap \Sigma = \emptyset$, variable ∩ terminal 没有元素相交
4. S is an element of V ; it is called the **start variable**, 开始
5. R is a finite set, whose elements are called **rules**. Each rule has the form $A \to w$, where $A \in V$ and $w \in (V \cup \Sigma)^*$.

A is a variable in V   w is the strings constructed from $(V \cup \Sigma)^*$

Context-Free Grammar

Example

Language $L = \{0^k 1^k : k \geq 0\}$ has CFG $G = (V, \Sigma, R, S)$,

variable set $V = \{S\}$
Terminal set $\Sigma = \{0, 1\}$
Start variable S
Rule set R: $S \to 0S1$
$S \to \epsilon$

$0^k 1^k$
11

$S \to 0S1 \to 00S11 \to 000S111 \Rightarrow 0 \cdots 0 S 1 \cdots 1$

Deriving strings and languages using CFG

$\Rightarrow$: **yeild** 产出

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

- $A \in V$
- $u, v, w \in (V \cup \Sigma)^*$,
- $A \to w$ 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 右由左得到

Let G = (V, Σ, R, S) be a context free grammar with

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

得到

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 $u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k$.

$u_1, u_2, \ldots u_k \in (V \cup \Sigma)^*$

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

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

Language of CFG

Definition

The language of CFG G = (V, Σ, 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, Σ, R, S) with

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

L(G) = ?

$S \to 0\overset{0S}{S} \to 00S \to \cdot\cdot$

$\to 0 \cdots 0S$

$\because S \to \varepsilon$

$\therefore S \to 0 \ldots 0 \Rightarrow L(G) = \{0^n : n \geq 0\}$

Example (Palindrome) 回文

CFG G = (V, Σ, R, S) with

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

$\begin{cases} S \to aSa \\ S \to bSb \\ S \to a \\ S \to b \\ S \to \varepsilon \end{cases}$

Language of this CFG ?

$S \Rightarrow aSa \Rightarrow aaSaa \overset{*}{\Rightarrow} a \cdots a\, S\, a \cdots a$

$\Rightarrow \begin{cases} a \ldots aaa \ldots a & S \to a \\ a \cdots aba \ldots a & S \to b \\ a \ldots a\varepsilon a \cdots a & S \to \varepsilon \end{cases}$

$S \Rightarrow bSb \Rightarrow bbSbb \cdots\cdots$ same measure as above

$L(G) = \{ w \in \Sigma^* \mid w = w^R \}$ R: reverse

简单的算术表达

Example (Simple Arithmetic Expressions)

CFG G = (V, Σ, R, S) with

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

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

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

$S \Rightarrow S \times S \Rightarrow S \times (S) \Rightarrow S \times (S + S) \Rightarrow 3 \times (S+S) \Rightarrow 3 \times (5+S)$

$\Rightarrow 3 \times (5+6)$

Regular Languages are context-free

Theorem (if) Regular Language (can say) => Context free

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 (general idea)

L是正则语言，有个DFA M接受，L是上下文无关的需要有个上下文无关语法G满足 $L=L(M)=L(G)$

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:

$w \in L(M) \Leftrightarrow w \in L(G)$

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 $S \overset{*}{\Rightarrow} w$.

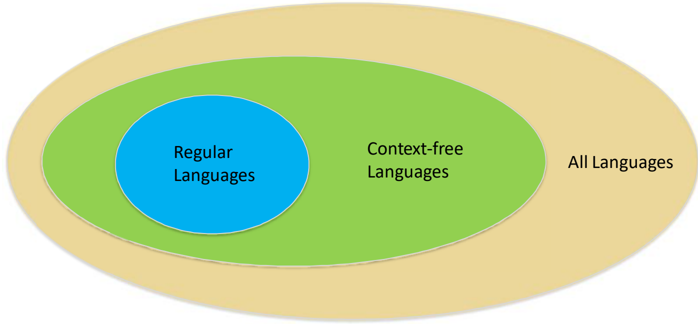
G的V就是M的所有Q

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 $R_i \to aR_j$. For every accepting state $q_i \in F$ add a rule $R_i \to \varepsilon$. Finally, make the start variable $S = R_0$.

R_0 is the initial state of the machine

Regular Languages are context-free

L is regular => L is context free
$\not\Leftarrow$



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

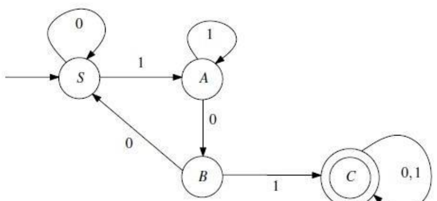
Regular Languages are context-free

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



将DFA转换为 CFG

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

Regular Languages are context-free

Example

- $G = \{V, \Sigma, R, S\}$

$V = \{S, A, B, C\}$

$\Sigma = \{0, 1\}$

Start variable : S (initial state of M)

Rules:

$S \to 0S \mid 1A$

$A \to 0B \mid 1A$

$B \to 0S \mid 1C$

$C \to 0C \mid 1C \mid \varepsilon$

Chomsky Normal Form (CNF) 乔姆斯基公式

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: 如果Rules满足下面三点条件

- $A \rightarrow BC$, 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 → 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 ($A \rightarrow B$, $A \rightarrow aBc$, $A \rightarrow BCDE$)

context free grammar → chomsky normal form

Converting CFG into CNF

Transformation steps

//右手边消除 start variable

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

In final,
All rules must be satisfied with above 3 requirements.

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, Σ, R, S), where V = {A, B}, Σ = {0, 1}, A is the start variable, and R consists of the rules:

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

ε- rules:
$A \to \varepsilon$
$B \to \varepsilon$

Convert this G to CNF:

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

$S \to A$
$A \to BAB \mid B \mid \varepsilon$
$B \to 00 \mid \varepsilon$

Converting CFG into CNF

Example

$S \to A$
✓ $A \to BAB \mid B \mid \varepsilon$
✓ $B \to 00 \mid \varepsilon$

Step 2. Remove **ε-rules**.

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

$S \to A \mid \varepsilon$
$A \to BAB \mid B \mid BB$
$B \to 00 \mid \varepsilon$

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

$S \to A \mid \varepsilon$
$A \to BAB \mid B \mid BB \mid AB \mid BA \mid A$
$B \to 00$

Converting CFG into CNF

Example

$S \to A$. $A \to A$
$A \to B$

$S \to A \mid \varepsilon$
$A \to BAB \mid B \mid BB \mid AB \mid BA \mid A$
$B \to 00$

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

(1) Remove $A \to A$:

$S \to A \mid \varepsilon$
$A \to BAB \mid B \mid BB \mid AB \mid BA$
$B \to 00$

(2) Remove $S \to A$:

$S \to B$
$A \to B$

$S \to \varepsilon \mid BAB \mid B \mid BB \mid AB \mid BA$
$A \to BAB \mid B \mid BB \mid AB \mid BA$
$B \to 00$

Converting CFG into CNF

Example

$S \to \varepsilon \mid BAB \mid B \mid BB \mid AB \mid BA$
$A \to BAB \mid B \mid BB \mid AB \mid BA$
$B \to 00$

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

(3) Remove $S \to B$:

$S \to \varepsilon \mid BAB \mid BB \mid AB \mid BA$
✓$A \to BAB \mid B \mid BB \mid AB \mid 3A$
$B \to 00$

(4) Remove $A \to B$:

$S \to \varepsilon \mid BAB \mid BB \mid AB \mid BA \mid 00$
$A \to BAB \mid BB \mid AB \mid BA$
$B \to 00$

Converting CFG into CNF

✓ $S \to \varepsilon \mid BAB \mid BB \mid AB \mid BA \mid 00$

$A \to BAB \mid BB \mid AB \mid BA \mid 00$

$B \to 00$

Example

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

(1) Remove S → BAB: 想办法把 BAB 变为两个 symbol

$BAB \xrightarrow{A_1} BA_1$

$S \to \varepsilon \mid BB \mid AB \mid BA \mid 00 \mid BA_1$

$A \to BAB \mid BB \mid AB \mid BA \mid 00$

$B \to 00$

assume $A_1 \to AB$

(2) Remove A → BAB:

$S \to \varepsilon \mid BB \mid AB \mid BA \mid 00 \mid BA_1$ replace $00 \to A_3A_3$

$A \to BB \mid AB \mid BA \mid 00 \mid BA_2$ replace $00 \to A_4A_4$

$B \to 00$

$A_1 \to AB$

$A_2 \to AB$

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 → 00:

(2) Remove A → 00:

$S \to \varepsilon \mid BB \mid AB \mid BA \mid BA_1 \mid A_3A_3$

$A \to BB \mid AB \mid BA \mid BA_2 \mid A_4A_4$

✓ $B \to 00$ => replace to $B \to A_5A_5$

$A_1 \to AB$

$A_2 \to AB$

$A_3 \to 0$

$A_4 \to 0$

$A_5 \to 0$

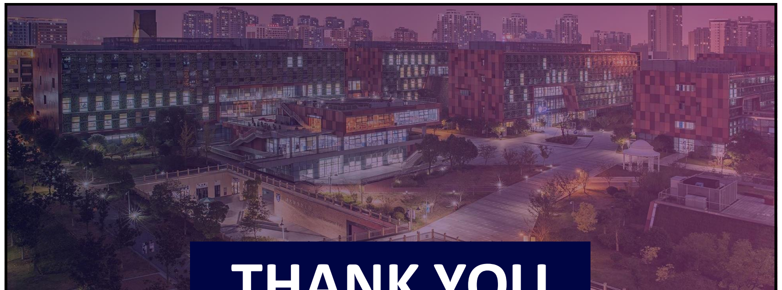
(3) Remove $B \to 00$

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 → 00:



THANK YOU

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

XJTLU | SCHOOL OF FILM AND TV ARTS