第7章 上下文无关语言的性质

7.1 上下文无关文法的范式

例7.1 给定文法：

G_1： $S\rightarrow 0 \mid 0A \mid E$

$A\rightarrow\varepsilon\mid 0A \mid 1A \mid B$

$B\rightarrow \_C$

$C\rightarrow 0 \mid 1 \mid 0C \mid 1C$

$D\rightarrow 1 \mid 1D \mid 2D$

$E\rightarrow 0E2 \mid E02$

定义的语言为

$L(G_1)=\{ 0x \mid x\in\{0,1\}^*\} \cup\{0x\_y \mid x\in\{0,1\}^*, y\in\{0,1\}^+\}$

7.1 上下文无关文法的范式

G_1： $S \rightarrow 0 \mid 0A \mid E$

$A \rightarrow \varepsilon \mid 0A \mid 1A \mid B$

$B \rightarrow \_C$

$C \rightarrow 0 \mid 1 \mid 0C \mid 1C$

$D \rightarrow 1 \mid 1D \mid 2D$

$E \rightarrow 0E2 \mid E02$

去掉无用符号后的文法

G_2： $S \rightarrow 0 \mid 0A$

$A \rightarrow \varepsilon \mid 0A \mid 1A \mid B$

$B \rightarrow \_C$

$C \rightarrow 0 \mid 1 \mid 0C \mid 1C$

去掉产生式 $A \rightarrow \varepsilon$后的文法

G_3： $S \rightarrow 0 \mid 0A$

$A \rightarrow 0 \mid 1 \mid 0A \mid 1A \mid B$

$B \rightarrow \_C$

$C \rightarrow 0 \mid 1 \mid 0C \mid 1C$

去掉产生式$A \rightarrow B$后的文法

G_4： $S \rightarrow 0 \mid 0A$

$A \rightarrow 0 \mid 1 \mid 0A \mid 1A \mid \_C$

$C \rightarrow 0 \mid 1 \mid 0C \mid 1C$

文法化简：去掉文法中的无用符号、ε产生式和单一产生式。

消除无用符号

定义7.1 **有用符号**和**无用符号**

CFG G=(V, T, P, S)，$X\in V\cup T$，如果存在 $w\in T^*$，$\alpha,\beta\in(V\cup T)^*$：

① $S\Rightarrow^*\alpha X\beta$，称X是可达的；

② $\alpha X\beta\Rightarrow^* w$，称X是可产生的；

③ 如果X既是可产生的，又是可达的，即 $S\Rightarrow^*\alpha X\beta\Rightarrow^* w$，称$X$是有用的，否则，称 X 是无用符号。

注意： 当X是无用的时候，它既可能是终极符号，也可能是语法变量。

消除无用符号

可产生的符号集

①每个终结符都是可产生的；

②$A \to \alpha \in P$且α中的符号都是可产生的，则A是可产生的；

可达的符号集

①起始变元S是可达的；

②$A \to \alpha \in P$且A是可达的，则α中符号都是可达的；

消除无用符号

例7.2 消除文法中的无用符号

$S \to AB \mid a$

$A \to b$

第一步：消除全部非“可达的”符号	第一步：消除全部非“可产生的”符号
$S \to AB \mid a$ $A \to b$	$S \to a$ $A \to b$
第二步：消除全部非“可产生的”符号	第二步：消除全部非“可达的”符号
$S \to a$ $A \to b$	$S \to a$

注意：

① 先消除非“可产生的”符号；

② 再消除非“可达的”符号；

消除无用符号

定理 7-1 对于任意CFL L，L≠Φ，则存在不含无用符号的CFG G，使得L(G)=L。

例 7-3 设有如下文法，消除无用符号

$$S \to AB \mid a \mid BB，A \to a，C \to b \mid ABa$$

第一步：消除全部非“可产生的”符号

$S \to a$

$A \to a$

$C \to b$

第二步：消除全部非“可达的”符号

$S \to a$

消除ε-产生式

- ε-产生式(ε-production)

  形如 $A \to \varepsilon$的产生式叫做ε-产生式。

  ε-产生式又称为空产生式(null production)。

- 可空(nullable)变量

  对于文法 $G=(V, T, P, S)$中的任意变量 A，如果 $A \Rightarrow^{+} \varepsilon$，则称 A 为可空变量。

消除ε-产生式

例7.4 有如下文法，求可空变量集。

$$S \rightarrow ABS \mid AB0$$

$$A \rightarrow CA \mid CBC$$

$$B \rightarrow 2 \mid \varepsilon$$

$$C \rightarrow 1C \mid \varepsilon$$

因为有 B→ε和 C→ε，变量 B 和 C 可以直接派生出ε。

作如下派生：

$$A \Rightarrow CBC \Rightarrow BC \Rightarrow C \Rightarrow \varepsilon$$

所以 A, B, C 都是可空变量。

但是不能简单地将 S→ABS 中的A 删去，而是要考虑表达出 A 产生ε和A 不产生ε的情况。

消除ε-产生式

消除ε-产生式的方法：

- 对形如 $A \to X_1X_2\ldots X_m$ 的产生式进行考察，

  找出文法的可空变量集U，

  然后对于 $\forall H \subseteq U$，从 $A \to X_1X_2\ldots X_m$ 中删除 H 中的变量。

  对于不同的 H，得到不同的 A 产生式，用这组 A 产生式替代产生式 $A \to X_1X_2\ldots X_m$。

- 必须避免在这个过程中产生新的ε-产生式：

  当 $\{ X_1, X_2, \ldots, X_m \} \subseteq U$ 时，不可将 $X_1, X_2, \ldots, X_m$ 同时从产生式 $A \to X_1X_2\ldots X_m$ 中删除。

消除ε-产生式

- 这样，可以得到其如下不含ε产生式的等价的文法：

  $S \rightarrow ABS \mid BS \mid AS \mid S \mid AB0 \mid B0 \mid A0 \mid 0$

  $A \rightarrow CA \mid C \mid A \mid CBC \mid BC \mid CB \mid CC \mid C \mid B$

  $B \rightarrow 1C \mid 1$

  $C \rightarrow 1C \mid 1$

消除ε-产生式

定理 7.2 对于任意 CFG G，存在不含ε-产生式的 CFG G′使得L(G′)=L(G)-{ε}。

例7.5 消除下列文法中的ε-产生式

S →AB

A →AaA | ε

B →BbB | ε

$$S \to AB \mid A \mid B$$

$$A \to AaA \mid Aa \mid aA \mid a$$

$$B \to BbB \mid Bb \mid bB \mid b$$

消除单一产生式

考虑如下的关于算术表达式的文法：

G_{exp1}：

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T*F \mid T/F \mid F$

$F \rightarrow F\uparrow P \mid P$

$P \rightarrow (E) \mid N(L) \mid id$

$N \rightarrow sin \mid cos \mid exp \mid abs \mid log \mid int$

$L \rightarrow L,E \mid E$

增加了句子的分析步骤：

$$E \Rightarrow T \Rightarrow F \Rightarrow P \Rightarrow id$$

原因：由形如 $A\rightarrow B$ 的单一产生式造成的。

消除单一产生式

单一产生式(unit production)

形如 $A \rightarrow B$ 的产生式称为单一产生式。

定理 7.3 对于任意 CFG G，$\varepsilon \notin L(G)$，存在等价的 CFG G_1，G_1不含无用符号、ε-产生式和单一产生式。

满足本定理的 CFG 为化简过的文法。

消除单一产生式

确定单元对

①如果有A→B，则称[A, B]为单元对；
②若[A, B]和[B, C]是单元对，则[A, C]是单元对。

消除单元对

①删除全部形如A→B的单元产生式；
②对每个单元对[A, B]，将B复制给A。

消除单一产生式

例7.5 消除下列文法的单一产生式

$S \to A \mid B \mid 0S1$

$A \to 0A \mid 0$

$B \to 1B \mid 1$

第一步：确定单元对

单元对有：[S, A]，[S, B]

第二步：消除单元对

$S \to 0A \mid 0 \mid 1B \mid 1 \mid 0S1$

$A \to 0A \mid 0$

$B \to 1B \mid 1$

建议的文法化简顺序

1. 消除ε -产生式
2. 消除单一产生式
3. 消除非可产生的无用符号
4. 消除非可达的无用符号

7.1 上下文无关文法的范式

乔姆斯基范式文法(Chomsky normal form， CNF)简称为Chomsky文法，或 Chomsky 范式。

CFG G=(V, T, P, S) 中的产生式形式：

$A \rightarrow BC$

$A \rightarrow a$

其中，A, B, C∈V，a∈T。

① CNF 中，不允许有 ε-产生式、单一产生式；

② 利用CNF派生长度为n的串，需要2n-1步；

③ 存在算法判定字符串w是否属于CFL；

④ 利用CNF的多项式时间解析算法---CYK算法；

7.1 上下文无关文法的范式

例子7.6 试将下列文法转换成等价的 CNF。

$S \rightarrow bA \mid aB$

$A \rightarrow bAA \mid aS \mid a$

$B \rightarrow aBB \mid bS \mid b$

1. 先引入变量B_a，B_b和产生式$B_a \rightarrow a$, $B_b \rightarrow b$，完成第一步变换。

   $S \rightarrow B_bA \mid B_aB$

   $A \rightarrow B_bAA \mid B_aS \mid a$

   $B \rightarrow B_aBB \mid B_bS \mid b$

   $B_a \rightarrow a$

   $B_b \rightarrow b$

2. 引入新变量 B_1, B_2

   $S \rightarrow B_bA \mid B_aB$

   $A \rightarrow B_bB_1 \mid B_aS \mid a$

   $B \rightarrow B_aB_2 \mid B_bS \mid b$

   $B_a \rightarrow a$

   $B_b \rightarrow b$

   $B_1 \rightarrow AA$

   $B_2 \rightarrow BB$

7.2 上下文无关语言的泵引理

The language $L = \{ a^n b^n c^n \mid n \geq 0 \}$ does not appear to be context-free.

Informal: The problem is that every variable can (only) act 'by itself' (*context-free*).

The problem of $A \Rightarrow^* vAy$:

If $S \Rightarrow^* uAz \Rightarrow^* uvAyz \Rightarrow^* uvxyz \in L$,

then $S \Rightarrow^* uAz \Rightarrow^* uvAyz \Rightarrow^* \ldots \Rightarrow^* uv^iAy^iz$

$\Rightarrow^* uv^ixy^iz \in L$ as well, for all $i=0,1,2,\ldots$

7.2 上下文无关语言的泵引理

Idea: If we can prove the existence of derivations for elements of the CFL L that use the step $A \Rightarrow^* vAy$, then a new form of ‘v-y pumping’ holds: $A \Rightarrow^* vAy \Rightarrow^* v^2Ay^2 \Rightarrow^* v^3Ay^3 \Rightarrow^* \ldots$)

要点：证明存在如上的“中递归”派生，反复调用

Observation: We can prove this existence if the parse-tree is tall enough.

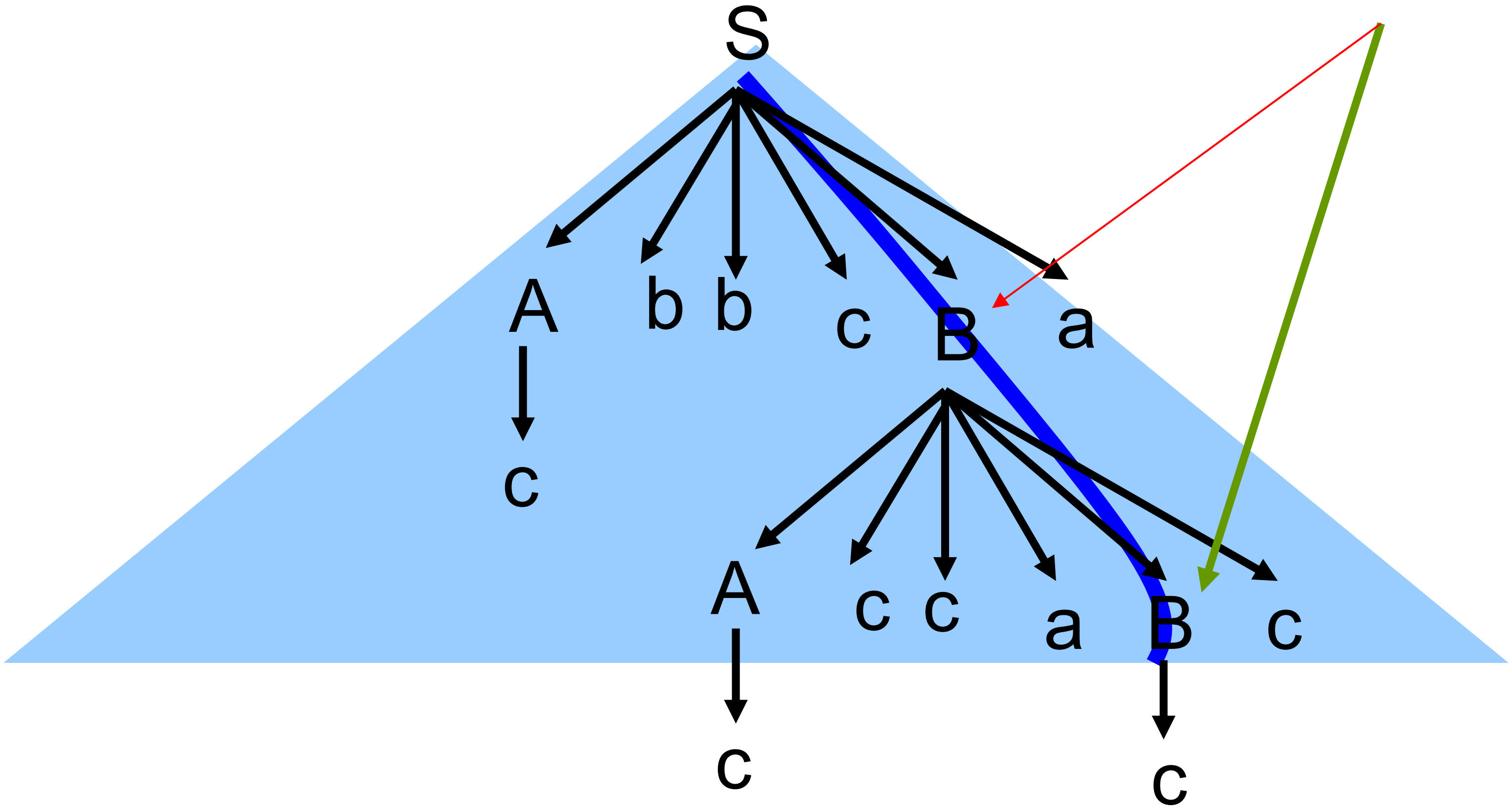
当派生树足够高，用尽了资源，就会出现重复

Pumping a Parse Tree

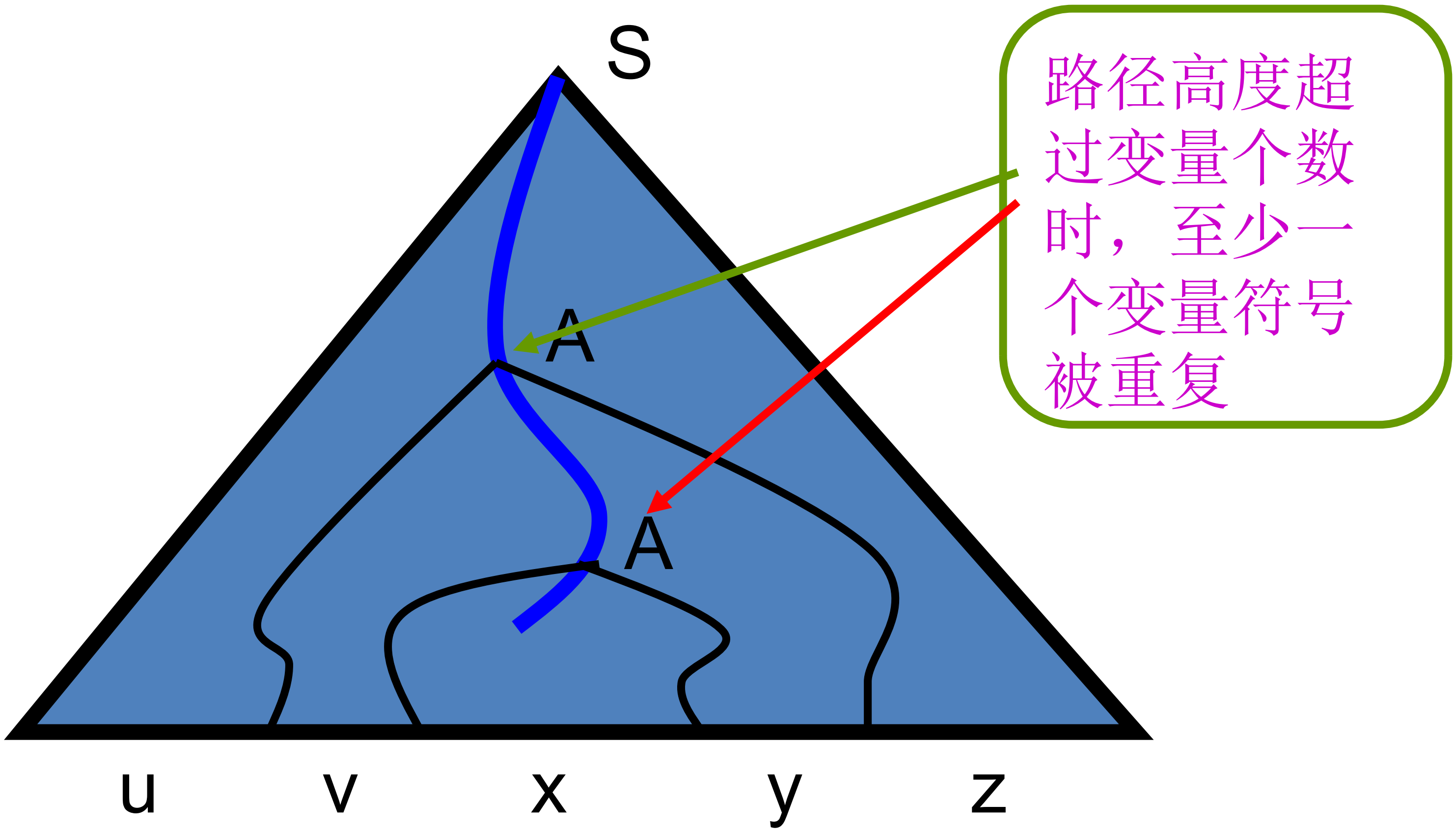
Parse tree for S ⇒ AbbcBa ⇒* cbbccccaBca

⇒ cbbccccacca

当树足够高时，有限的变量符就会被重复使用



Pumping a Parse Tree



If s = uvxyz $\in$ L is long, then its parse-tree is tall. Hence, there is a path on which a variable A repeats itself. We can pump this A–A part.

A Tree Tall Enough

Let L be a context-free language, and let G be its grammar with maximal b symbols on the right side of the rules: $A \rightarrow X_1 \ldots X_b$

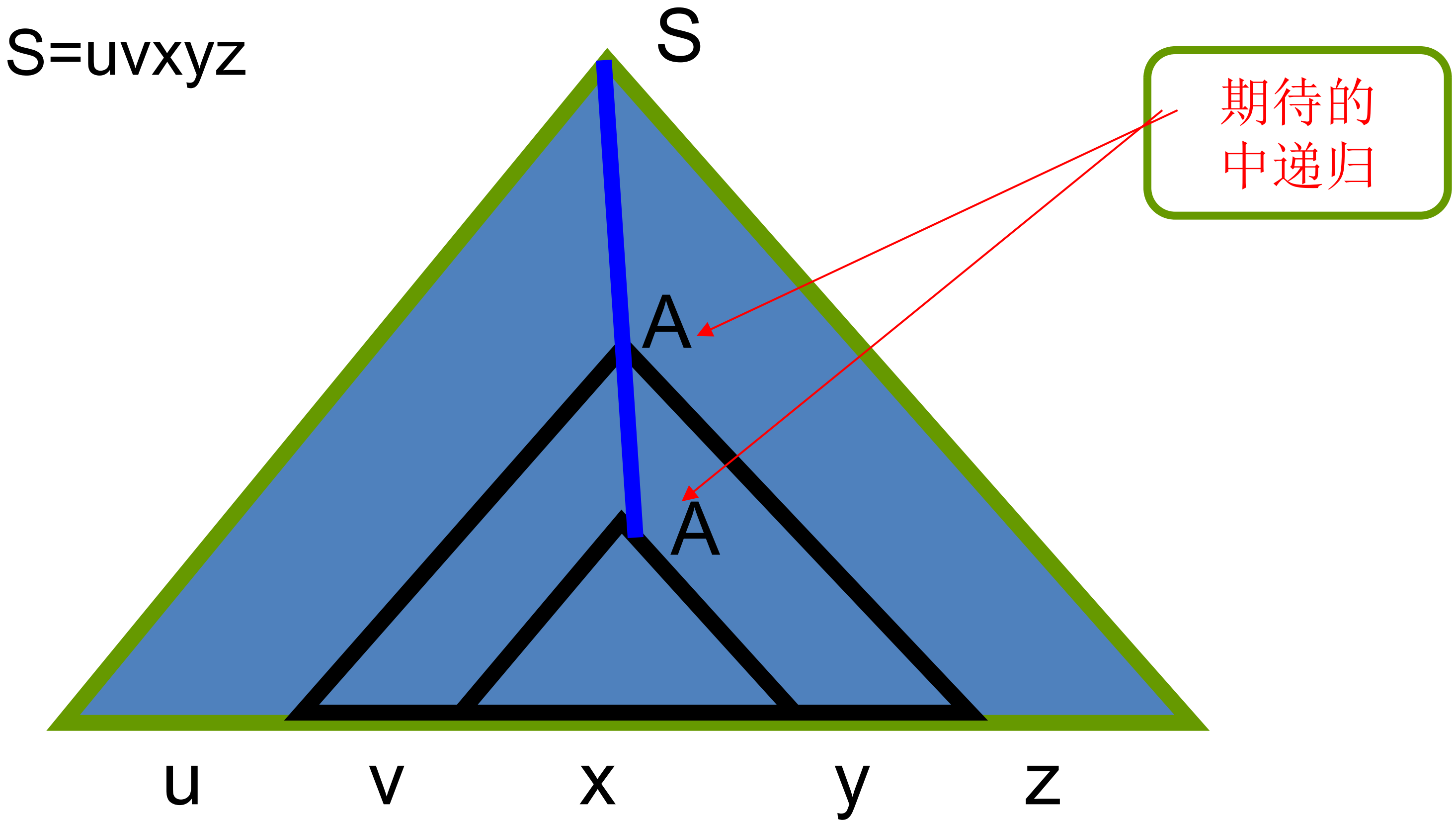
A parse tree of depth h produces a string with maximum length of b^h.
Long strings implies tall trees.

变量个数V

树高h=v+2时

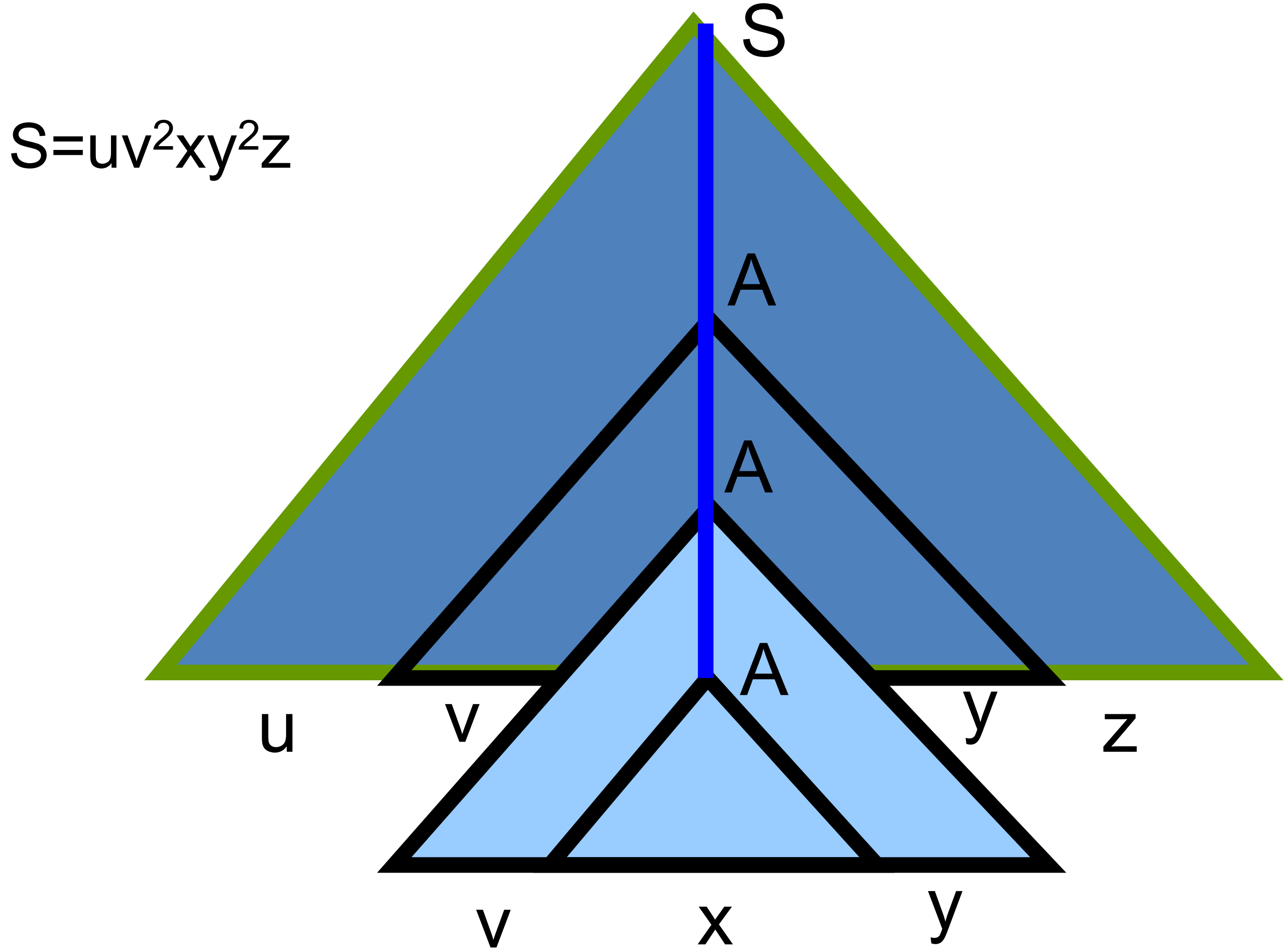
Let $|V|$ be the number of variables of G.
If $h = |V|+2$ or bigger, then there is a variable on a 'top-down path' that occurs more than once.

uvxyz ∈L 派生树足够高，分段记号



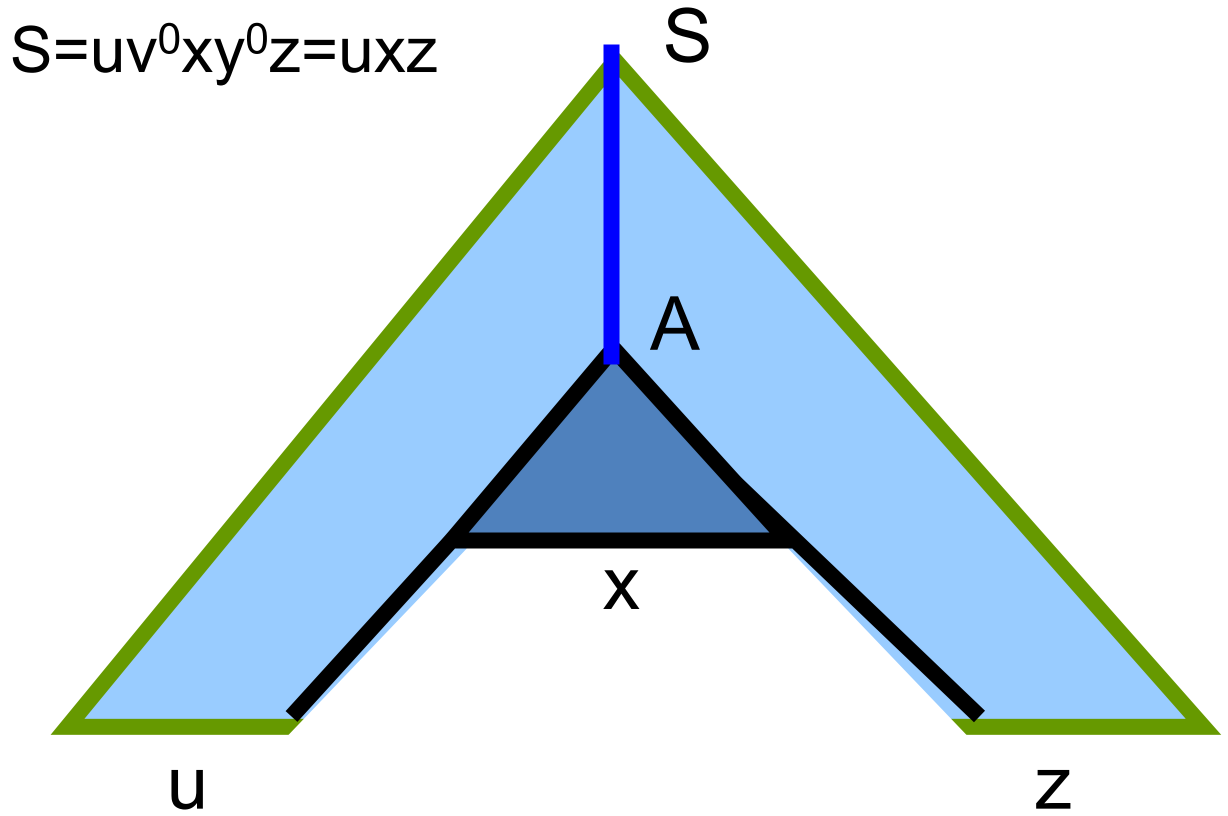
By repeating the A–A part we get…

$uv^2xy^2z \in L$, i=2



… while removing the A–-A gives…

$uxz \in L$, $i=0$



In general $uv^ixy^iz \in L$ for all $i=0,1,2,\ldots$

上下文无关语言的泵引理

引理7.1 For every context-free language L, there is a <u>pumping length</u> p, such that for every string $s \in L$ and $|s| \geq p$, we can write $s=uvxyz$ with

一分为五 +
一或二处 重复

1) $uv^ixy^iz \in L$ for every $i \in \{0,1,2,\ldots\}$ //两处打圈
2) $|vy| \geq 1$ //真圈
3) $|vxy| \leq p$
//扇出度小于泵长（变量未重复，不超过变量个数）

Note that 1) implies that $uxz \in L$
2) says that vy cannot be the empty string ε
Condition 3) is not always used

7.2 上下文无关语言的泵引理

Let $G=(V,\Sigma,R,S)$ be the grammar of a CFL.
Maximum size of rules is $b\geq 2$: $A \to X_1\ldots X_b$
A string s requires a <u>minimum</u> tree-depth $\geq \log_b|s|$.
If $|s| \geq p=b^{|V|+2}$, then tree-depth $\geq |V|+2$, hence there is a path and variable A where A repeats itself: $S \Rightarrow^* uAz \Rightarrow^* uvAyz \Rightarrow^* uvxyz$
It follows that $uv^ixy^iz \in L$ for all $i=0,1,2,\ldots$

Furthermore:

$|vy| \geq 1$ because tree is minimal
$|vxy| \leq p$ because bottom tree with $\geq p$ leaves has no 'repeating path'

7.2 上下文无关语言的泵引理

例7-1 证明 $L=\{a^n b^n c^n \mid n\geq 1\}$不是 CFL。

取 $z = a^p b^p c^p \in L$ ，设 $z = uvwxy$，

注意到 $|vwx|\leq p$，所以 v, w 和 x 并在一起不能同时有 3 种字符。

可能出现以下几种情况：

(1) v 和 x 只包含 a，取 $i=2$，则在 uv^2wx^2y 中，

a 的个数明显大于 b, c 的个数，因此它不在 L 中。

(2) v 和 x 只包含 b 或只包含 c，理由与 (1) 同，uv^2wx^2y 也不在 L 中。

(3) v 只包含 a, x 只包含 b，取 $i=2$，则在 uv^2wx^2y 中，

a, b 的个数将超过 c 的个数，它不在 L 中。

(4) v 只包含 b, x 只包含c，理由与 (3) 同，uv^2wx^2y 也不在 L 中。

(5) v 或 x 包含两种不同的符号，例如，v 包含 a 和 b，则在 uv^2wx^2y 中将呈现 a 和 b 交错出现的情况，显然它不在 L 中。

所以，L 不是 CFL。

7.2 上下文无关语言的泵引理

例7.2 求证 $C = \{a^i b^j c^k \mid 0 \le i \le j \le k\}$ is not context-free.

Proof Let p be the pumping length, and $s = a^p b^p c^p \in C$

P.L.: $s = uvxyz$, such that $uv^i xy^i z \in C$ for every $i \ge 0$

Two options for $1 \le |vxy| \le p$: 只有2种可能，分别讨论

1) $vxy = a^*b^*$, then the string uv^2xy^2z has not enough c's, hence $uv^2xy^2z \notin C$ //在前面打圈，c的个数少

2) $vxy = b^*c^*$, then the string $uv^0xy^0z = uxz$ has too many a's, hence $uv^0xy^0z \notin C$

Contradiction: C is not a context-free language.

例7.3 求证 $D = \{ ww \mid w \in \{0,1\}^* \}$不是CFL

Carefully take the strings $s \in D$.

Let p be the pumping length, take $s=0^p1^p0^p1^p$.

Three options for $s=uvxyz$ with $1 \le |vxy| \le p$:

1) If a part of y is to the left of | in $0^p1^p|0^p1^p$, then second half of uv^2xy^2z starts with “1”
2) Same reasoning if a part of v is to the right of middle of $0^p1^p|0^p1^p$, hence $uv^2xy^2z \notin D$
3) If x is in the middle of $0^p1^p|0^p1^p$, then uxz equals $0^p1^i0^j1^p \notin D$ (because i or $j < p$)

Contradiction: D is not context-free.

7.3 上下文无关语言的封闭性

主要讨论：

- CFL 在并、乘积、闭包、补、交等运算下的封闭性。
- **定理** 8-1 CFL 在并、乘积、闭包运算下是封闭的。
- 定理 8-2 CFL 在交运算下不封闭的。
- 推论8-1 CFL在补运算下是不封闭的。
- **定理**8-3 CFL 与 RL 的交是 CFL。

7.4 上下文无关语言的判定算法

- 不存在判断算法的问题：

① CFG G，是不是二义性的？

② CFL L 的补是否确实不是CFL？

③ 任意两个给定 CFG 是否等价？

- 存在判断算法的问题：

④ CFG L 是非空语言么？

⑤ CFG L 是有穷的么？

⑥ 一个给定的字符串 x 是 L 的句子么？

7.4 上下文无关语言的判定算法

- CFL 空否的判定
- 基本思想：

  设 L 为一个CFL，则存在 CFG G，使得 L(G)=L。

  由算法 6-1，可以求出等价的 CFG G′，G′ 中不含派生不出终极符号行的变量。

  显然，如果 NewV 中包含 G 的开始符号，则 L 就是非空的。否则，L 就是空的。

  因此，通过改造算法 6-1，可得到判定 L 是否为空的算法 8-1。

7.4 上下文无关语言的判定算法

- x 是否为 L 的句子的判定
- 判断 x 是否为给定文法生成的句子的根本方法是看 G 能否派生出 x。
- 一种最简单的算法是用穷举法，这种方法又称为“试错法”,是“带回溯”的，所以效率不高。其时间复杂度为串长的指数函数。
- 典型的自顶向下的分析方法：递归子程序法、LL(1) 分析法、状态矩阵法等。
- 典型的自底向上的分析方法：LR 分析法、算符优先分析法。
- 这些基本的方法均只可以分析 CFG 的一个真子类。

7.4 上下文无关语言的判定算法

- CYK算法
- 基本思想是从 1 到 $|x|$，找出 x 的相应长度的子串的派生变量。
- 效率较高的根本原因是它在求 x 的长度为 i 的子串 y 的“派生变量”时，是根据相应的 CNF 中的形如 $A \rightarrow BC$ 的产生式，使用已经求出的 B 是 y 的前缀...的“派生变量”，而 C 是相应的后缀的“派生变量”的结果。
  - 使用 CNF，对于任给的字符串 $x=x_1x_2\ldots x_n$，
  - 若 $B \rightarrow x_k$，$C \rightarrow x_{k+1}$，$A \rightarrow BC$，则 $A \Rightarrow^* x_k x_{k+1}$。
  - 若 $B \Rightarrow^* x_i\ldots x_k$，$C \Rightarrow^* x_{k+1}\ldots x_j$，$A \rightarrow BC$，则 $A \Rightarrow^* x_i x_j$。
  - 用 $x_{i,k}$表示 $x_i\ldots x_{i+k}$，$V_{i,k}$表示能派生出$x_{i,k}$的变量集合。
  - 求$V_{1,n}$并检查 S 是否 是 $V_{1,n}$ 中的变量。
- 时间复杂度为 $O(n^3)$。
- 由 Cocke, Younger 和 Kasami 在20世纪60年代分别独立提出。

7.4 上下文无关语言的判定算法

算法8-3 CYK算法

输入：CNF $G=(V,T,P,S)$，x；

输出：$x \in L(G)$ 或者 $x \notin L(G)$；

主要数据结构：

集合 $V_{i,j}$——可以派生出子串 $x_{i,k}$ 的变量的集合。这里，$x_{i,k}$ 表示 x 的从第 i 个字符开始的，长度为 k 的字串。

(1) for $i=1$ to $|x|$ do

(2) $V_{i,1}=\{A|A \to x_{i,1} \in P\}$；

(3) for $k=2$ to $|x|$ do

(4) for $i=1$ to $|x|-k+1$ do

begin

(5) $V_{i,k}=\Phi$；

(6) for $j=1$ to $k-1$ do

(7) $V_{i,k}= V_{i,k} \cup \{A \mid A \to BC \in P$ 且 $B \in V_{i,j}$ 且 $C \in V_{i+j,k-j}\}$；

end

7.4 上下文无关语言的判定算法

- CYK算法举例
- 给出Chomsky范式文法 G:
  $S \to AB \mid BC \qquad A \to BA \mid a$
  $B \to CC \mid b \qquad C \to AB \mid a$
  和$x = baaba$，判断 x 是否属于L(G)。

$i \to$

$j \downarrow$

x =	b	a	a	b	a
	1	2	3	4	5
1	B	A, C	A, C	B	A, C
2	S, A	B	S, C	A, S	
3	Φ	B	B		
4	Φ	S, A, C			
5	S, A, C				

7.4 上下文无关语言的判定算法

本章小结

(1) 泵引理：与RL的泵引理类似，CFL的泵引理用来证明一个语言不是 CFL。

(2) CFL 在并、乘、闭包、代换、同态映射、逆同态映射等运算下是封闭的。

(3) CFL在交、补运算下是不封闭。

(4) 存在判定CFG产生的语言是否为空、是否有穷、是否无穷，以及一个给定的符号串是否为该文法产生的语言的一个句子的算法。