上下文无关语言的性质

CFL的性质

- 本章讨论CFL的性质
- 主要内容
  - CFL的泵引理及其应用
  - CFL的封闭性
    - 封闭运算：并、乘、闭包、代换、同态映射、逆同态映射
    - 不封闭运算：交、补

上下文无关语言的性质

- CFL的判定算法。
  - 判定CFG产生的语言是否为空、有穷、无穷。
  - 一个给定的符号串是否为该文法产生的语言的一个句子等问题。
- 重点
  - CFL的泵引理、 CFL 的封闭性。
- 难点
  - CFL的泵引理的应用、CFL的同态原象是CFL。

上下文无关语言的泵引理

启发：

- **RG G=(V，T，P，S)，使得L(G)=L，当x足够长时，如|x|≥|V|+1时，存在u、v、w∈T*，|v|≥1，使得x=uvw，当G为右线性文法时，必定存在语法变量A，使得如下派生成立：**

$$S \Rightarrow^* uA \Rightarrow^* uvA \Rightarrow^* \ldots\ldots \Rightarrow^* uv^iA \Rightarrow^* uv^iw$$

v是可以被重复任意多次的字串！

CFL也有类似性质？

上下文无关语言的泵引理

- **CFL**的自嵌套特性：如果**L**是一个**CFL**，**CFG G=(V，T，P，S)**是产生**L**的文法。当**L**是一个无穷语言时，必存在**w**∈**L**，**A**∈**V**，α，β∈**(V**∪**T)***，且α和β中至少有一个不为ε，使得如下派生成立

$$S\Rightarrow^{*} \gamma A \delta \Rightarrow^{+} \gamma \alpha A \beta \delta \Rightarrow^{+} z$$

- 文法**G**中存在有如下形式的派生

$$A\Rightarrow^{+} \alpha A \beta$$

上下文无关语言的泵引理

这种类型的派生预示着

$S \Rightarrow^{*} \gamma A \delta \Rightarrow^{+} \gamma \alpha A \beta \delta \Rightarrow^{+} z$

并且

$S \Rightarrow^{*} \gamma A \delta \Rightarrow^{+} \gamma \alpha^{n} A \beta^{n} \delta \Rightarrow^{+} z'$

设 $\alpha \Rightarrow^{*} v$， $\beta \Rightarrow^{*} x$， $\gamma \Rightarrow^{*} u$， $A \Rightarrow^{*} w$， $\delta \Rightarrow^{*} y$

可以得到如下派生

上下文无关语言的泵引理

$S \Rightarrow^* \gamma A \delta$

$\Rightarrow^* u \alpha A \beta \delta$

$\Rightarrow^* u \alpha A \beta y$

…

$\Rightarrow^* u \alpha^n A \beta^n y$

$\Rightarrow^* uv^nAx^ny$

$\Rightarrow^* uv^nwx^ny$

上下文无关语言的泵引理

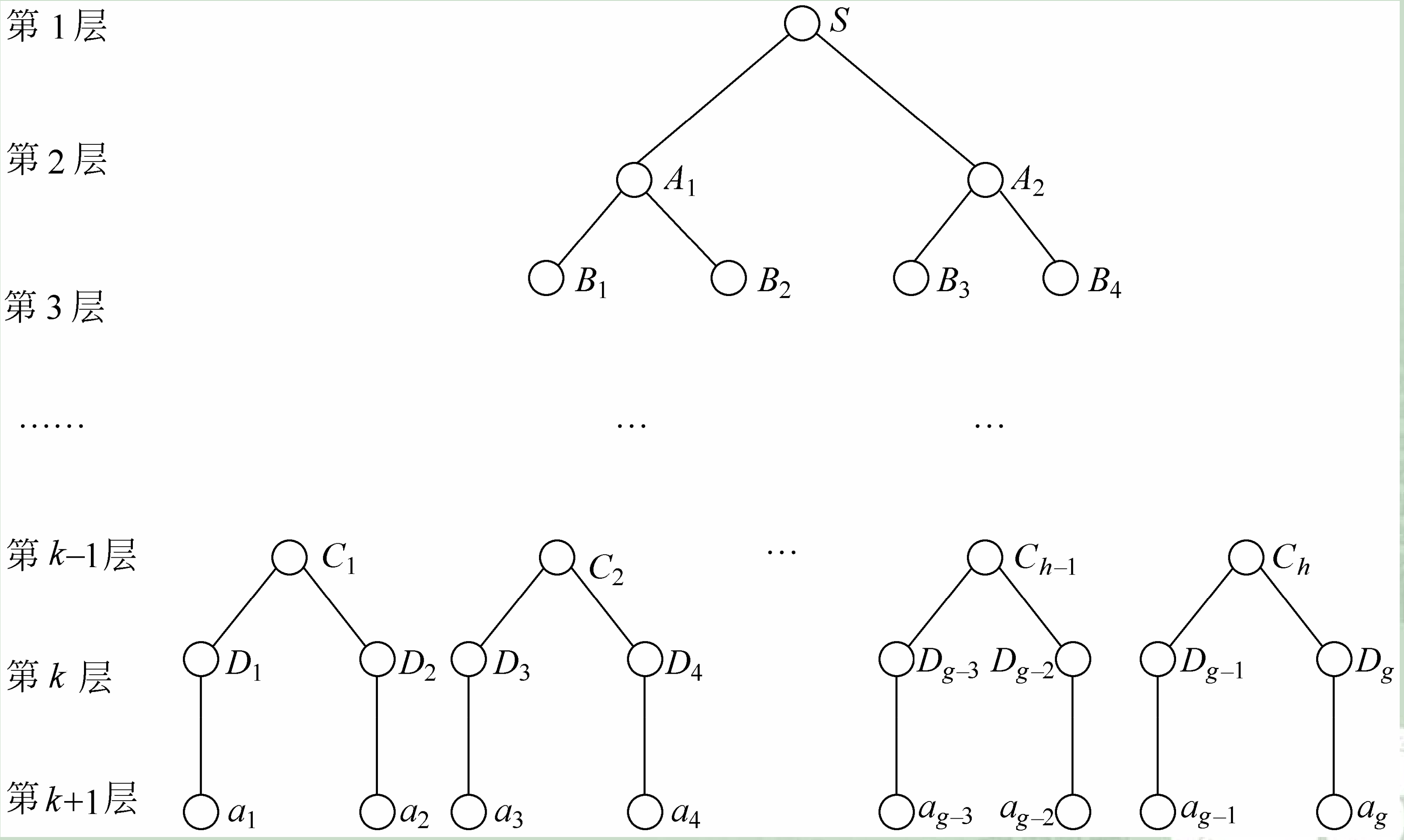
引理 （**CFL**的泵引理）对于任意的**CFL L**，存在仅仅依赖于**L**的正整数**N**，对于任意的**z**∈**L**，当|**z**|≥**N**时，存在**u**，**v**，**w**，**x**，**y**，使得**z=uvwxy**，同时满足：

(1) |**vwx**|≤**N**；

(2) |**vx**|≥**1**；

(3) 对于任意的非负整数**i**，$uv^iwx^iy \in L$。

上下文无关语言的泵引理



上下文无关语言的泵引理

- 证明要点：

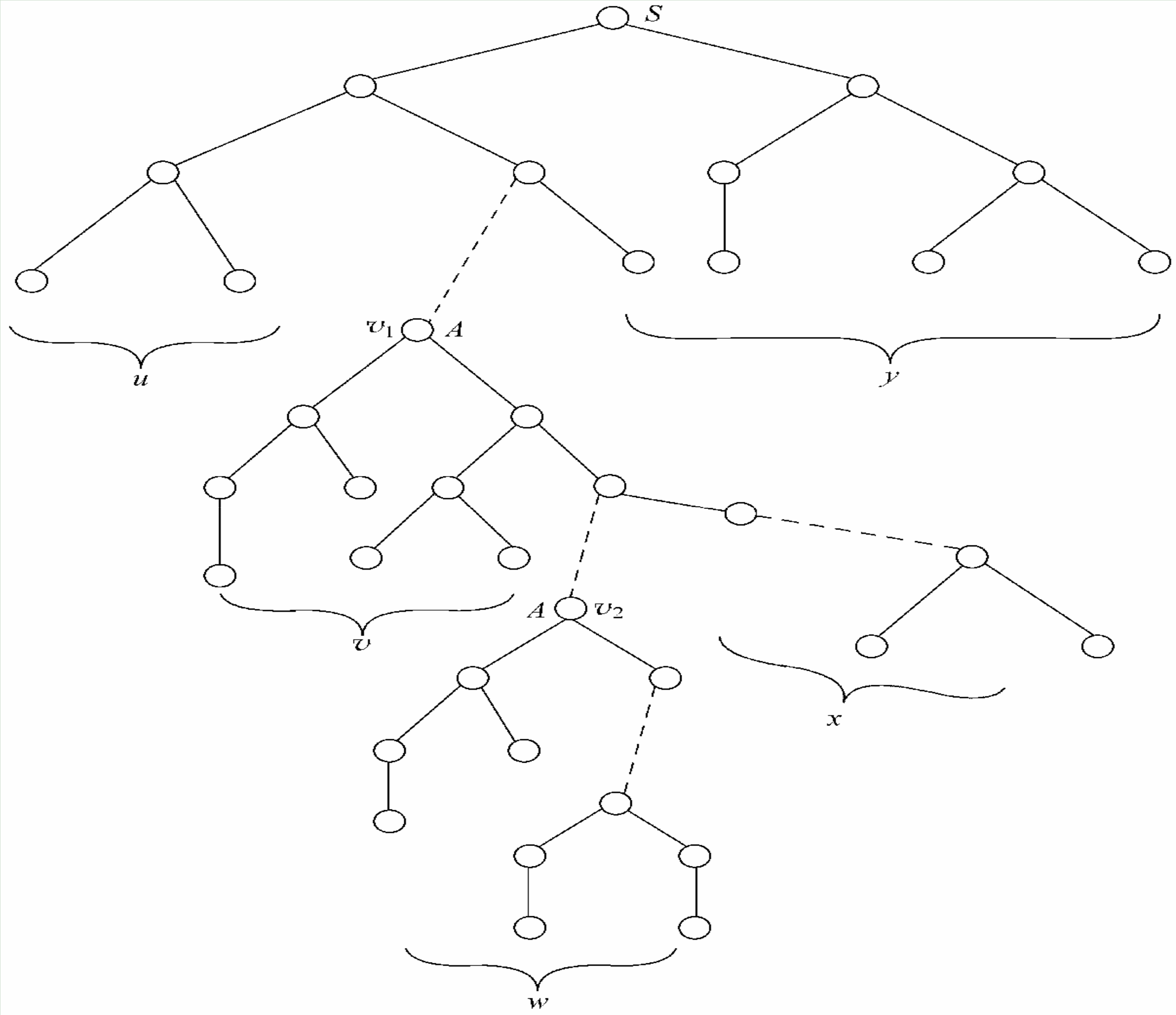
（1）用CNF作为CFL的描述工具。

（2）对于任意的$z \in L$，当k是z的语法树的最大路长时，必有$|z| \leq 2^{k-1}$成立。

（3）仅当z的语法树呈上图所示的满二元树时，才有$|z|=2^{k-1}$，其他时候均有$|z|<2^{k-1}$。

（4）取$N=2^{|V|}=2^{|V|+1-1}$，$z \in L$，$|z| \geq N$。

上下文无关语言的泵引理



上下文无关语言的泵引理

（5） 取z的语法树中的最长的一条路p，p中的非叶子结点中必定有不同的结点标有相同的语法变量。

（6） p中最接近叶子且都标有相同的语法变量A的两个结点为v_1、v_2，如上图所示。

上下文无关语言的泵引理

- 结点v_1左边的所有叶子结点的标记从左到右构成的字符串为u；
- 结点v_1的为根的子树中，v_2左边的所有叶子结点的标记从左到右构成的字符串为v；
- 结点v_2为根的子树的结果为w；
- 结点v_1为根的子树中v_2右边的所有叶子结点的标记从左到右构成的字符串为x；
- 结点v_1右边的所有叶子结点的标记从左到右构成的字符串为y。

上下文无关语言的泵引理

- z=uvwxy
- 注意到v_1-子树的最大路长小于等于|V|+1，所以，v_1的结果vwx满足：$|vwx| \leqslant 2^{(|V|+1)-1}=2^{|V|}=N$
- v_1的后代v_2标记为变量A，所以，$|vx| \geqslant 1$。
- 此时有
- $S \Rightarrow^* uAy \Rightarrow^+ uvAxy \Rightarrow^+ uvwxy$
- 显然，对于i=0，1，2，3，…
- $A \Rightarrow^* v^iAx^i \Rightarrow^+ v^iwx^i$

上下文无关语言的泵引理

- 总结上述推导

（7） $S \Rightarrow^{*} uAy \Rightarrow^{+} uvAxy \Rightarrow^{+} uvwxy=z$，$|vwx| \leqslant 2^{(|V|+1)-1}=2^{|V|}=N$，$|vx| \geqslant 1$。

（8）对于任意非负整数i，$S \Rightarrow^{*} uAy \Rightarrow^{+} uv^{i}Ax^{i}y \Rightarrow^{+} uv^{i}wx^{i}y$。

上下文无关语言的泵引理

- **例** 证明$L=\{a^nb^nc^n|n\geqslant 1\}$不是CFL。

证明：

取$z=a^Nb^Nc^N\in L$

(1) $v=a^h$，$x=b^f$，$h+f\geqslant 1$。

- $uv^iwx^iy=a^{N+(i-1)h}b^{N+(i-1)f}c^N$，
  - 当$i\neq 1$时，$N+(i-1)h\neq N$和$N+(i-1)f\neq N$中至少有一个成立。
  - $uv^iwx^iy=a^{N+(i-1)h}b^{N+(i-1)f}c^N\notin L$。

上下文无关语言的泵引理

(2) $v=b^h$，$x=c^f$，$h+f\geqslant 1$。

- $uv^iwx^iy=a^Nb^{N+(i-1)h}\,c^{N+(i-1)f}$
  - 当$i\neq 1$时，$N+(i-1)h\neq N$和$N+(i-1)f\neq N$中至少有一个成立。
  - $uv^iwx^iy=a^{Nh}b^{N+(i-1)}c^{N+(i-1)f}\notin L$。
- 这些都与泵引理矛盾，所以，L不是CFL。

上下文无关语言的泵引理

- **例** 证明$L=\{a^n b^m c^n d^m|n，m\geq 1\}$不是CFL。

  证明:

  取$z= a^N b^N c^N d^N$

  $v=a^h$、$x=b^f$；$v=b^h$、$x=c^f$；$v=c^h$、$x=d^f$，$h+f\geq 1$等3种情况。

上下文无关语言的泵引理

- 设$v=a^h$、$x=b^f$，并且$h+f\geq 1$
- $uv^iwx^iy=a^{N+(i-1)h}b^{N+(i-1)f}c^Nd^N$
- 当$i\neq 1$时，$N+(i-1)h\neq N$和$N+(i-1)f\neq N$至少有一个成立。
- $uv^iwx^iy=a^{N+(i-1)h}b^{N+(i-1)f}c^Nd^N\notin L$

上下文无关语言的泵引理

- 同理可以证明，当$v=b^h$、$x=c^f$或者$v=c^h$、$x=d^f$，$h+f\geqslant 1$时，
  $uv^iwx^iy=a^{N+(i-1)h}b^{N+(i-1)f}c^Nd^N\notin L$
  对$i\neq 1$成立。
- 由泵引理，L不是CFL。

CFL的封闭性

定理 CFL 在并、乘积、闭包运算下是封闭的。

证明要点：

令CFG

- $G_1=(V_1, T_1, P_1, S_1)$，$L(G_1)=L_1$，
- $G_2=(V_2, T_2, P_2, S_2)$，$L(G_2)=L_2$，
- $V_1 \cap V_2=\Phi$，且S_3，S_4，$S_5 \notin V_1 \cup V_2$。

CFL的封闭性

$G_3=(V_1\cup V_2\cup\{S_3\},T_1\cup T_2,P_1\cup P_2\cup\{ S_3 \rightarrow S_1|S_2\},S_3)$

$G_4=(V_1\cup V_2\cup\{S_4\},T_1\cup T_2,P_1\cup P_2\cup\{ S_4 \rightarrow S_1S_2\},S_4)$

$G_5=(V_1\cup\{S_5\}$，T_1，$P_1\cup\{ S_5 \rightarrow S_1S_5|\varepsilon \}$，$S_5)$

显然，G_3、G_4、G_5都是CFG，并且

$L(G_3)=L_1\cup L_2$

$L(G_4)= L_1L_2$

$L(G_5)= L_1^*$

CFL的封闭性

定理 CFL 在交运算下是不封闭的。

证明要点：

设$L_1=\{0^n1^n2^m|n,m\geqslant 1\}$，$L_2=\{0^n1^m2^m|n,m\geqslant 1\}$。

G_1： $S_1\to AB$

$A\to 0A1|01$

$B\to 2B|2$

G_2： $S\to AB$

$A\to 0A|0$

$B\to 1B2|12$

CFL的封闭性

- 显然，$L(G_1)=L_1$、$L(G_2)=L_2$ 。
- $L= L_1 \cap L_2=\{0^n1^n2^n|n\geqslant 1\}$ 不是CFL。
- 所以，CFL在交运算下是不封闭的。

CFL的封闭性

推论 **CFL**在补运算下是不封闭的。

证明要点：

由于**CFL**对并运算的封闭性、对交运算的不封闭性和下列式子，可以得出**CFL**对补运算是不封闭的结论。

$$L_1 \cap L_2 = \overline{\overline{L_1 \cap L_2}} = \overline{\overline{L_1} \cup \overline{L_2}}$$

CFL的封闭性

定理　CFL与RL的交是CFL。

证明：

（1）构造

设PDA $M_1=(Q_1, \Sigma, \Gamma, \delta_1, q_{01}, Z_0, F_1)$

$L_1=L(M_1)$

DFA $M_2=(Q_2, \Sigma, \delta_2, q_{02}, F_2)$

$L_2=L(M_2)$

CFL的封闭性

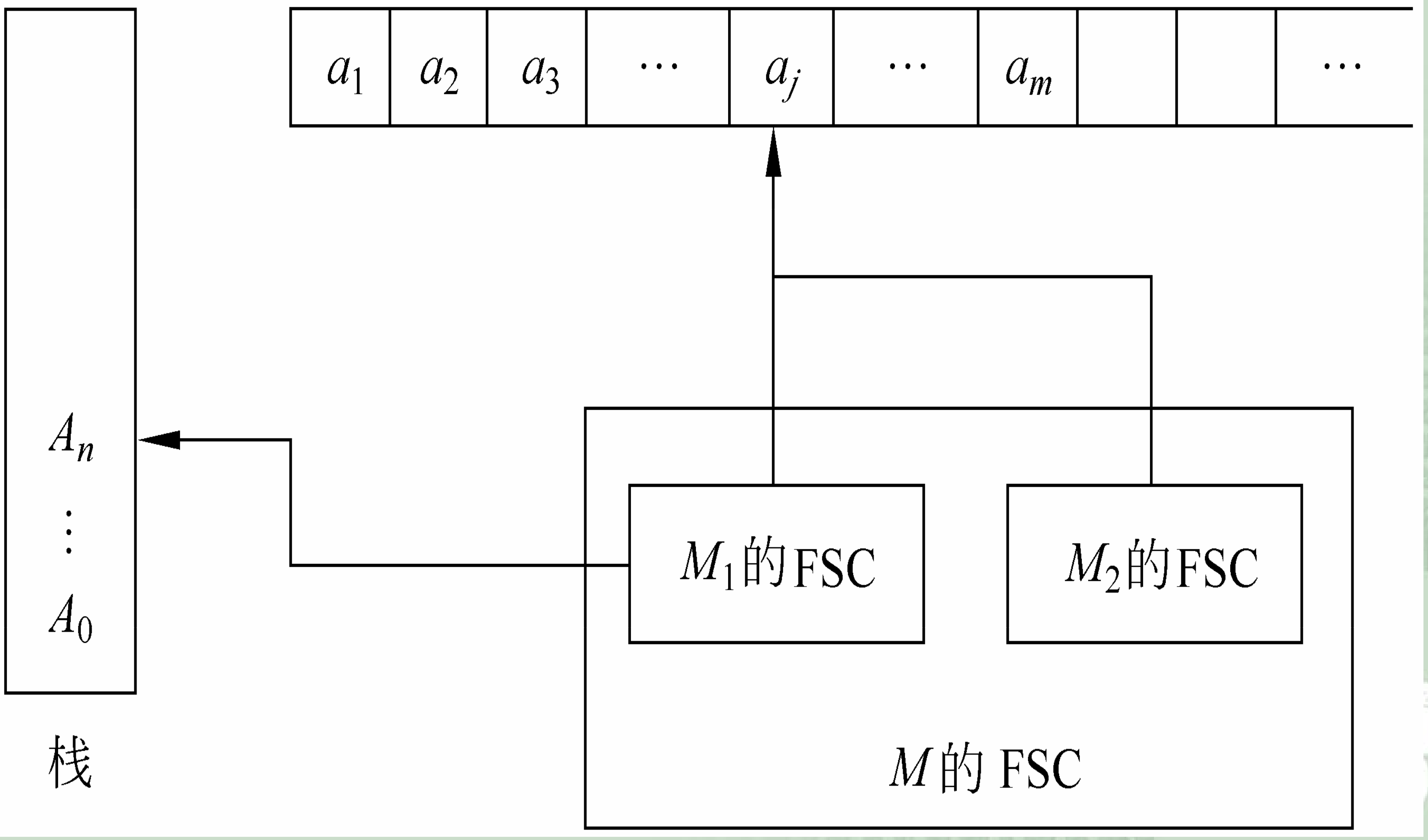
- 取 PDA $M=(Q_1\times Q_2, \Sigma, \Gamma, \delta, [q_{01}, q_{02}], Z_0, F_1\times F_2)$

对$\forall([q, p], a, Z)\in(Q_1\times Q_2)\times(\Sigma\cup\{\varepsilon\})\times\Gamma$

$\delta([q, p], a, Z)=\{([q', p'], \gamma) \mid (q', \gamma)\in\delta_1(q, a, Z)$ 且 $p'=\delta(p, a)\}$

- 可以用下图表示构造的基本思想。

CFL的封闭性



CFL的封闭性

① M使用的栈就是M_1的栈。

② M的状态包括两个分量：一个为M_1的状态，用来使M的动作能准确地模拟M_1的动作；另一个为M_2的状态，用来使M的动作能准确地模拟M_2的动作。

③ 当M_1执行ε-移动时，M_2不执行动作。

CFL的封闭性

(2) 证明构造的正确性。

施归纳于n证明：

$([q_{01}，q_{02}]，w，Z_0) \vdash_M^n ([q，p]，\varepsilon，\gamma)$

$\Leftrightarrow (q_{01}，w，Z_0) \vdash_{M1}^n (q，\varepsilon，\gamma)$ & $\delta(q_{02}，w)=p$

再注意到$[q，p] \in F_1 \times F_2 \Leftrightarrow q \in F_1$ & $p \in F_2$

这就是说，对于$\forall x \in \Sigma^*$,

$x \in L(M) \Leftrightarrow x \in L(M_1)$ & $x \in L(M_2)$

所以，

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

CFL的封闭性

- 根据本定理证明中给出的方法，可以用N个正则语言L_1、L_2、…、L_N的识别器(DFA)的有穷状态控制器构成这N个正则语言的交的识别器(DFA)：

$M_{\cap}=(Q_1 \times Q_2 \times \ldots \times Q_N, \Sigma, \delta, [q_{10}, q_{20}, \ldots, q_{N0}], F_1 \times F_2 \times \ldots \times F_N)$

CFL的封闭性

定理 CFL在代换下是封闭的。

① **CFG G=(V，T，P，S)**，使得**L=L(G)**。对于 $\forall$**a**$\in$**T**，**f(a)**是Σ上的**CFL**，且**CFG** $G_a=(V_a，\Sigma，P_a，S_a)$，使得$f(a)=L(G_a)$。

② **f(a)**的所有句子都是由S_a派生出来的，所以，构造的基本思想是用S_a替换产生**L**的**CFG**的产生式中出现的终极符号**a**。

CFL的封闭性

③产生**f(L)**的文法

$$G' = (\bigcup_{a\in T} V_a \bigcup V, \sum, \bigcup_{a\in T} P_a \bigcup P', S)$$

$$P' = \{A \to A_1A_2\cdots A_n \mid$$

$$A \to X_1X_2\cdots X_n \in P \text{且}$$

$$\text{如果} X_i \in V, \text{则} A_i = X_i, \text{否则} A_i = S_{X_i}\}$$

CFL的封闭性

④ 证明L(G′)=f(L)

设x∈L(G′)，从而

$S \Rightarrow^*_{G'} S_a S_b \ldots S_c$

$\Rightarrow^+_{G'} x_a S_b \ldots S_c$

$\Rightarrow^+_{G'} x_a x_b \ldots S_c$

…

$\Rightarrow^+_{G'} x_a x_b \ldots x_c$

$= x$

S_a，S_b，…，$S_c \in \{S_d \mid d \in T\}$。且

$S_a \Rightarrow^*_{G'} x_a$

$S_b \Rightarrow^*_{G'} x_b$

…

$S_c \Rightarrow^*_{G'} x_c$

CFL的封闭性

由G′的定义知，

$S \Rightarrow^*_{G'} S_a S_b \ldots S_c \Leftrightarrow S \Rightarrow^*_G ab\ldots c$

并且，对于$\forall d \in T$，$S_d \Rightarrow^*_{G'} x_d \Leftrightarrow S_d \Rightarrow^*_{Gd} x_d$

所以，$ab\ldots c \in L$，$x_a \in L(G_a)$，$x_b \in L(G_b)$，…，$x_c \in L(G_c)$

故：$x= x_a x_b \ldots x_c = f(a)f(b)\ldots f(c)=f(ab\ldots c)$

即：$x \in f(L)$。从而$L(G') \subseteq f(L)$

用类似的方法，不难证明$f(L) \subseteq L(G')$。

综上所述，定理得证。

CFL的封闭性

推论 CFL的同态像是CFL。

- 注意到对任意的**CFG G=(V，T，P，S)，L=L(G)**。对于∀**a**∈**T**，**|f(a)|=1**，所以它是Σ上的**CFL**，根据这个定理，得到此推论。

CFL的封闭性

定理 CFL L的同态原像是CFL。

基本思路:

① 描述工具**PDA**。

② **PDA** $M_2=(Q_2,\ \Sigma_2,\ \Gamma,\ \delta_2,\ q_0,\ Z_0,\ F)$接受**L**

③ 设$T=\{a_1,\ a_2,\ \ldots,\ a_n\}$根据 $f(a_1)=x_1$，$f(a_2)=x_2$，…，$f(a_n)=x_n$，M_1有穷控制器中的缓冲区存放$f(a_1)$，$f(a_2)$，…，$f(a_n)$的任一后缀。

CFL的封闭性

④ 对于任意的$x\in\Sigma_1^*$，设$x=b_1b_2\ldots b_n$，M_1是否接受x，完全依据于M_2是否接受$f(b_1)f(b_2)\ldots f(b_n)$。

⑤ M_1的形式定义为

$M_1=(Q_1, \Sigma_1, \Gamma, \delta_1, [q_0, \varepsilon], Z_0, F\times\{\varepsilon\})$

$Q_1=\{[q, x] \mid q\in Q_2$，存在$a\in\Sigma_1$，x是f(a)的后缀}

当M_1扫描到$a\in\Sigma_1$时，将f(a)存入有穷控制器，

$([q, f(a)], A) \in \delta_1([q, \varepsilon], a, A)$

CFL的封闭性

然后，M_1模拟M_2处理存在缓冲区中的f(a)。

- M_1用ε-移动模拟M_2的非ε-移动：

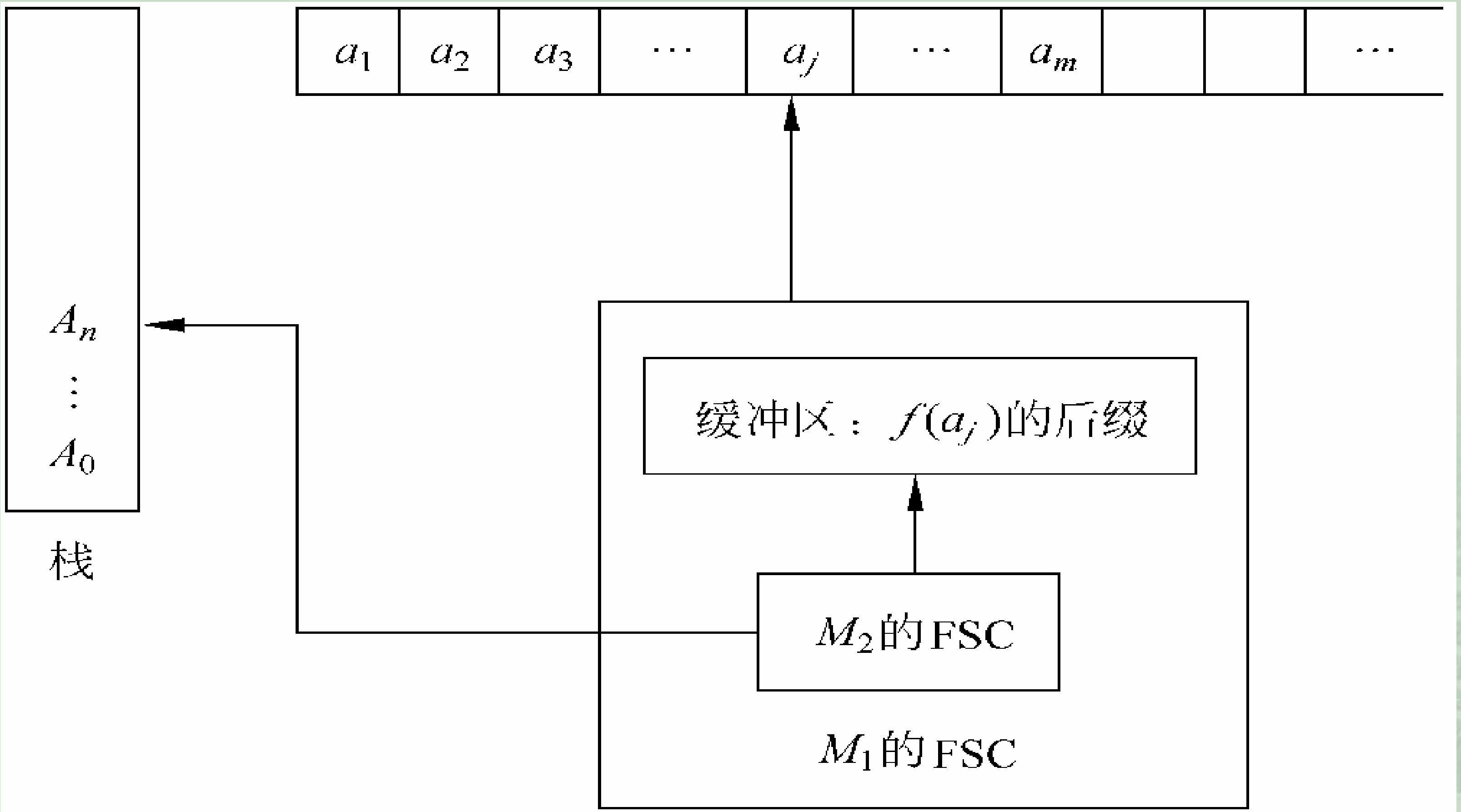
$(p, \gamma) \in \delta_2(q, a, A) \Leftrightarrow$

$([p, x], \gamma) \in \delta_1([q, ax], \varepsilon, A)$

- M_1用ε-移动模拟M_2的ε-移动

$(p, \gamma) \in \delta_2(q, \varepsilon, A) \Leftrightarrow$

$([p, x], \gamma) \in \delta_1([q, x], \varepsilon, A)$



a_1
a_2
a_3
$\cdots$
a_j
$\cdots$
a_m
$\cdots$
A_n
$\vdots$
A_0
栈
缓冲区：$f(a_j)$的后缀
M_2的FSC
M_1的FSC

图 M的构造示意图

CFL的封闭性

- 构造的正确性证明。
- 证明$x=a_1a_2\ldots a_n \in L(M_1)$的充分必要条件是$f(a_1)f(a_2)\ldots f(a_n) \in L(M_2)$。

CFL的判定算法

对CFL存在判定算法的问题：

- L是否为空。
- L是否无穷。
- 一个给定的字符串x是否为L的句子。

L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。由前述算法，我们可以求出等价的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，通过改造前述算法，我们可得到判定L是否为空的算法。

L空否的判定

算法 判定CFL L是否为空。

输入：CFG G=(V，T，P，S)。

输出：G是否为空的判定；CFG G′=(V′，T，P′，S)，V′中不含派生不出终极符号行的变量，并且有L(G′)=L(G)。

主要步骤：

(1) $OLDV=\Phi$;

(2) $NEWV=\{A|A\rightarrow w\in P$ 且 $w\in T^*\}$

L空否的判定

(3) **while OLDV$\neq$NEWV do**

begin

(4) **OLDV=NEWV;**

(5) **NEWV=OLDV$\cup$\{A$\rightarrow\alpha\in$P且 $\alpha\in$(T$\cup$OLDV)*\};**

end

(6) **V′ =NEWV;**

(7) **P′ =\{ A$\rightarrow\alpha$ | A$\rightarrow\alpha\in$P且 A$\in$V′ 且 $\alpha\in$(T$\cup$V′)*\};**

(8) **if S$\in$NEWV then L(G)非空 else L(G)为空**

L是否无穷的判定

- **可派生性图表示**(Derivability Graph of G——DG)

设**CFG G=(V，T，P，S)**，**G**的**可派生性图表示**是满足下列条件的有向图：

(1) 对于$\forall$**X**$\in$**V**$\cup$**T**，图中有且仅有一个标记为**X**的顶点；

(2) 如果$\mathbf{A \rightarrow X_1X_2 \ldots X_n} \in \mathbf{P}$，则图中存在从标记为**A**的顶点分别到标记为$\mathbf{X_1}$、$\mathbf{X_2}$、…、$\mathbf{X_n}$的弧；

(3) 图中只有满足条件**1**和**2**的顶点和弧。

L是否无穷的判定

- G的可派生性图表示中，任意两个顶点之间最多有一条相同方向的弧。
- G的可派生性图表示表达了文法G中的语法变量之间的派生关系。
- 派生$A \Rightarrow^{+} \alpha A \beta$存在的充分必要条件是G的可派生性图表示中存在一条从标记为A的顶点到标记为A的顶点的长度非0的有向回路。

L是否无穷的判定

定理 设**CFG G=(V，T，P，S)**已化简，**L(G)**为无穷语言的充分必要条件是**G**的可派生性图表示中存在一条有向回路。

证明要点：

对应某一条有向回路，寻找一条从**S**出发，到达此回路的路，可以构造出下列形式的派生

$$S \Rightarrow^{+} \gamma A \rho \Rightarrow^{+} \gamma \alpha A \beta \rho$$

从而对任意的非负整数**i**，

$$S \Rightarrow^{+} \gamma A \rho \Rightarrow^{+} \gamma \alpha^{i} A \beta^{i} \rho$$

L是否无穷的判定

- **DG中标记为终极符号的结点是无用的**
- **简化的可派生性图表示(simplified derivability graph of G，SDG)**
- 设**CFG G=(V，T，P，S)，G的简化的可派生性图表示**是从G的可派生性图表示中删除所有标记为终极符号的顶点后得到的图。

L是否无穷的判定

定理 设**CFG G=(V，T，P，S)**已化简，**L(G)**为无穷语言的充分必要条件是**G**的简化的可派生性图表示中存在一条有向回路。

L是否无穷的判定

算法 判定CFL L是否为无穷语言。

- 输入：CFG G=(V，T，P，S)。
- 输出：G是否为无穷的判定；CFG G′=(V′，T，P′，S)，G′化简，并且有L(G′)=L(G)。

L是否无穷的判定

(1) 调用文法化简算法;

(2) if S$\notin$V′ then L(G)为有穷的语言

else

begin

(3) 构造G′的简化的可派生性图表示SDG;

(4) if SDG中含有回路 then L(G′)为无穷语言

(5) else L(G′)为有穷有语言。

end

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。

x是否为L的句子的判定

- CYK算法的基本思想是从1到|x|，找出x的相应长度的子串的派生变量，x是L的句子当且仅当S是x 的长度为|x|的子串(即x)的一个派生变量。
- 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→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。

x是否为L的句子的判定

算法 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$V_{i,k}$——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的子串。

x是否为L的句子的判定

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

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

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

end

x是否为L的句子的判定

x∈L(G)当且仅当S ∈ $V_{1,|x|}$。

其中：语句(1)和(2)完成长度为1的子串的派生变量的计算。其时间复杂度为 |P|。语句(3)控制算法依次完成长度是2、3、…、|x|的子串的派生变量的计算；语句(4)控制完成对于串x中的所有长度为k的子串的派生变量的计算。这里的计算顺序依次为从第1个字符开始的长度为k的子串、从第2个字符开始的长度为k的子串、…、从第|x|-k+1个字符开始的长度为k的子串。语句(6)控制实现长度为k的子串的不同切分方式下的派生的可能性。

小结

本章讨论了**CFL** 的性质和**CFL**的一些判定问题。

(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**。它不能证明一个语言是 **CFL**。

小结

(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、补运算下是不封闭的。

(4) 存在判定**CFG**产生的语言是否为空、是否有穷，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。

习题

1 用泵引理证明下列语言不是CFL。

(1) $\{0^n1^m \mid n=m^2\}$。

(2) $\{0^n1^n2^n \mid n\geq 0\}$。

2 证明上下文无关语言在逆运算下是封闭的。

习题

3 判定符号串w=aabbb是否属于由文法

$S \to AB$

$A \to BB \mid a$

$B \to AB \mid b$

生成的语言。