形式语言与自动机

语言描述：文法、正则表达式、自动机

形式语言（代数语言）：精确地描述语言 $\alpha \to \beta$（α, β 均为字符串）

4元组 $G=(V, \Sigma, P, S)$　V：非终结符　P：一组重写规则的有限集合

Σ：终结符　$S \in V$：句子符号（初始符）

推导：设 $G=(V, \Sigma, P, S)$ 是一个文法，在 $(V \cup \Sigma)^*$ 上定义关系 $\Rightarrow_G$ 如下：

↓ 闭包，即 V 和 Σ 的任意组合　↓ 直接推生或推导

如果 $\alpha\beta\gamma$ 是 $(V \cup \Sigma)^*$ 中的符号串，且 $\beta \to \delta$ 是 P 的产生式，则

$$\alpha\beta\gamma \Rightarrow_G \alpha\delta\gamma$$

$\overset{+}{\Rightarrow_G}$：至少要推导一次，称为按非平凡方式推出　$\beta_i \to \beta_{i+1}$ 一步 $i \geq 1$

$\overset{*}{\Rightarrow_G}$：称为推生，表示 $\Rightarrow_G$ 的自反和传递闭包

最左推导：只改写最左边的那个非终结符

最右推导：只改写最右边的 ……（规范推导）

例：$G=(\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

P：$E \to E+T \mid T$　$T \to T*F \mid F$　$F \to (E) \mid a$

字符串 $a+a*a$ 的推导过程

最右：$E \Rightarrow E+T \Rightarrow E+T*F \Rightarrow E+T*a \Rightarrow E+F*a \Rightarrow E+a*a$

$\Rightarrow T+a*a \Rightarrow F+a*a \Rightarrow a+a*a$.

句型与句子（推导过程中位于中间过程）

定义：一些特殊类型的符号串为文法 $G=(V, \Sigma, P, S)$ 的句子形式（句型）：

1) S 是一个句子形式

2) 如果 $\alpha\beta\gamma$ 是一个句子形式，且 $\beta \to \delta$ 是 P 的产生式，则 $\alpha\delta\gamma$ 也是一个句子形式

文法 G 的不含非终结符的句子形式称为 G 生成的句子。由文法 G 生成的语言，记作 $L(G)$

$L(G)=\{w \mid x\in\Sigma^*, S \overset{+}{\Rightarrow}_G x\}$ $G=(V,\Sigma,P,S)$

~~正则语法~~：

正则文法：P：$A\to Bx$，或 $A\to x$，其中 $A,B\in V$，$x\in\Sigma^*$，也称为右线性正则文法

正则文法或写为 $A\to xB$ 右线性正则文法

例：$G=(V,\Sigma,P,S)$

$V=\{S,A,B\}$ $\Sigma=\{a,b\}$

P：(a) $S\to aA$ (b) $A\to aA$ (c) $A\to bbB$ (d) $B\to bB$ (e) $B\to b$

求 $L(G)$

可以将 (c) $A\to bbB$ 改写为 $A\to bB'$ $B'\to bB$

则满足右线性正则文法

$L(G)=\{a^n b^m\}$ ~~$n\geq 1, m\geq 1$~~ $n\geq 1, m\geq 3$

上下文无关文法：Context-free grammar，CFG 也称为2型文法

$A\to\alpha$，其中 $A\in V$，$\alpha\in(V\cup\Sigma)^*$.

例：$G=(V,\Sigma,P,S)$ $V=\{S,A,B,C\}$ $\Sigma=\{a,b,c\}$

P：(a) $S\to ABC$ (b) $A\to aA\mid a$ (c) $B\to bB\mid b$ (d) $C\to BA\mid C$

$L(G)=\{a^n b^m a^k c^\alpha\}$，$n\geq 1$，$m\geq 1$，$k\geq 0$，$\alpha\in\{0,1\}$

$(k>0,\alpha=0$，或 $\alpha=1\}$

上下文有关文法：CSG 也称1型文法

P：$\alpha AB\to\alpha\gamma\beta$，其中 $A\in V$，$\alpha,\beta,\gamma\in(V\cup\Sigma)^*$，且 γ 至少包含一个字符.

即：$\{u\to v$，$u\in(V\cup\Sigma)^+$，$v\in(V\cup\Sigma)^*$，并且 $|v|\geq|u|$

$|\cdot|$ 表示长度

例：$G=(V,\Sigma,P,S)$ $V=\{S,A,B,C\}$ $\Sigma=\{a,b,c\}$.

P：(a) $S\to ABC$ (b) $A\to aA\mid a$ (c) $B\to bB\mid b$ (d) $BC\to BCC$

$L(G)=\{a^n b^m c^k\}$，$n\geq 1$，$m\geq 1$

无约束文法：0型文法

$P: \alpha \to \beta$

$L(G0) \supseteq L(G1) \supseteq L(G2) \supseteq L(G3)$

0型 ⊇ 有长 ⊇ 无长 ⊇ 正则

语言与文法类型的约定：

如果一个语言能由几种文法所产生，则把这种语言称为在这几种文法中当限制最为的那种文法所产生的语言。

例：$G=(\{S, A, B\}, \{a, b\}, P, S)$

$P: S \to aB \quad S \to bA \quad A \to aS \quad A \to bAA$

$A \to a \quad B \to bS \quad B \to aBB \quad B \to b$

G为0型文法 $L(G)=\{$等数量的a和b构成的串$\}$

CFG产生的语言句子的派生树表示：

1) 对于每个 $x \in N \cup \Sigma$ 给一个标记作为节点，S作为树的根节点

2) 如果一个节点的标记为A，并且它是非叶子节点，则 $A \in N$

3) 如果一个节点的标记为A，它的 $k(k>0)$ 个直接后裔节点按从左到右的次序依次标记为 $A_1, A_2, \cdots, A_k$，则 $A \to A_1 A_2 \cdots A_k$ 一定是P中的一个产生式

例：$G=(\{S, A\}, \{a, b\}, P, S)$

$P: S \to bA \quad A \to bAA \quad A \to a$

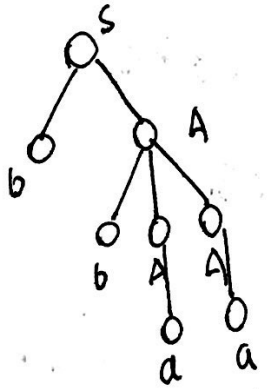
G所产生的句子bbaa可以由下面的生成树表示：

$S \Rightarrow bA$

$\Rightarrow bbAA$

$\Rightarrow bbaA$

$\Rightarrow bbaa$



上下文无关文法的二义性：一个文法G，如果存在某个句子有不只一棵分析树与之对应，则称这个文法是二义的。 计算机无法执行 歧义的产生(不同分支)

例： P：介词　NP：名词短语　PP：介词短语　AUX：助词　N：名词

S：句子/短语

有限自动机与正则文法

确定的有限自动机（DFA）：五元组　$M=(\Sigma, Q, \delta, q_0, F)$

Σ：输入符号的有穷集合

Q：Q是状态的有限集合

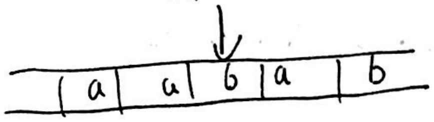
q_0：$q_0 \in Q$是初始状态

F：F是终止状态集合，$F \subseteq Q$

δ：是Q与Σ的直积 $Q \times \Sigma$ 到Q（下一个状态）的映射。它支配着有限状态控制的行为，有时也称为状态转移函数

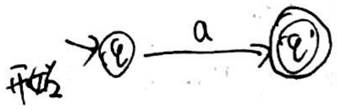
示意图：

有限控制器



判断，接收，终止

状态图：　$\delta(q, a) = q'$



a为句子

语言：如果一个句子x使得有限自动机M有 $\delta(q_0, x) = P$，$P \in F$，那么称句子x被M接受，T(M)即为被M接受的句子的全集，即

$T(M) = \{x \mid \delta(q_0, x) \in F\}$

不确定的有限自动机：　五元组 $(\Sigma, Q, \delta, q_0, F)$ M　　NFA

δ：Q与Σ的直积 $Q \times \Sigma$ 到Q的幂集 2^Q 的映射.　　输出不确定

区别 { NFA　$\delta(q, a)$是一个状态集合

　　　 DFA　$\delta(q, a)$是一个状态

NFA与DFA的关系：（化为FA）　（因此不再区分）

定理3.1：设L是一个被NFA所接受的句子的集合，则存在一个DFA，它能够接受L

正则文法与有限自动机的关系：

若 $G=(V_N, V_T, P, S)$ 是一个正则文法，则存在一个有限自动机 $M=(\Sigma, Q, \delta, q_0, F)$

使得 $T(M)=L(G)$