由G构造M的一般步骤：

(1) 令 $\Sigma = V_T$, $Q = V_N \cup \{T\}$, $q_0 = S$, 其中 T是一个新增加的非终结符

(2) 如果在P中有产生式 $S \to \varepsilon$, 则 $F = \{S, T\}$, 否则 $F = \{T\}$

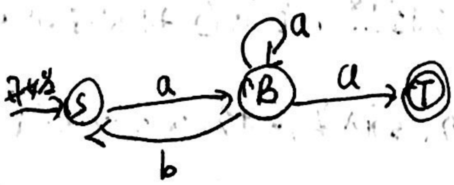
(3) 如果在P中有产生式 $B \to a$, $B \in V_N$, $a \in V_T$, 则 $T \in \delta(B, a)$

(4) 如果在P中有产生式 $B \to aC$, $B, C \in V_N$, $a \in V_T$, 则 $C \in \delta(B, a)$

(5) 对于每一个 $a \in V_T$, 有 $\delta(T, a) = \phi$

例：将正则文法 $G = (V_N, V_T, P, S)$, $V_N = \{S, B\}$, $V_T = \{a, b\}$, $P = \{S \to aB,$
$B \to bS \mid aB \mid a\}$

转化为与G等价的NFA



(1) $M = (\Sigma, Q, \delta, q_0, F)$

$\Sigma = V_T = \{a, b\}$ $Q = V_N \cup \{T\} = \{S, B, T\}$ $q_0 = S$ $F = \{T\}$

(2) $\delta(S, a) = \{B\}$ $\delta(B, a) = \{B, T\}$ $\delta(T, a) = \phi$

$\delta(S, b) = \phi$ $\delta(B, b) = \{S\}$ $\delta(T, b) = \phi$

定理：若 $M = (\Sigma, Q, \delta, q_0, F)$ 是一有限自动机，则存在正则文法 $G = (V_N, V_T, P, S)$ 使 $L(G) = T(M)$。

由M构造G的一般步骤：

(1) 令 $V_N = Q$, $V_T = \Sigma$, $S = q_0$

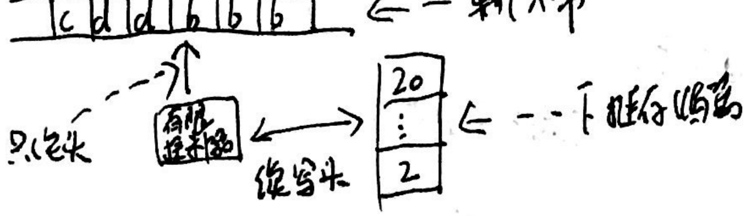
(2) 如果 $C \in \delta(B, a)$, 则 $B \to aC$ $(B, C \in Q, a \in \Sigma)$

(3) 如果 $C \in \delta(B, a)$, $C \in F$, 则 $B \to a$

下推自动机与CFG

(PDA)

PDA可以看成是一个带有附加的下推存储器的有限自动机，下推存储器是一个栈



7元组：$M=(\Sigma, Q, \Gamma, \delta, q_0, z_0, F)$

Γ：下推存储器符号的有穷集合

$z_0\in\Gamma$：最初出现在下推存储器顶端的符号

δ：从 $Q\times(\Sigma\cup\{\varepsilon\})\times\Gamma$ 到 $Q\times\Gamma^*$ 子集的映射

$\delta(q,a,z)=\{(q_1,\gamma_1),(q_2,\gamma_2),\cdots,(q_m,\gamma_m)\}$

其中，$q_1,q_2\cdots q_m\in Q$，$a\in\Sigma$，$z\in\Gamma$，$\gamma_1,\gamma_2\cdots\gamma_m\in\Gamma^*$

γ_i 代替 z，γ_i 的符号按照从左到右的顺序依次从下向上推入到存储器

特殊情况：ε 移动，输入头不移动只修改栈

$\delta(q,\varepsilon,z)=\{(q_1,\gamma_1),(q_2,\gamma_2),\cdots,(q_m,\gamma_m)\}$

符号约定：

$a:(q,z\gamma)\vdash_M(q',\beta\gamma)$

(改栈中)(状态由 $q\to q'$)

0次/多次合法转移：$a:(q,z\gamma)\overset{*}{\vdash}_M(q',\beta\gamma)$

下推自动机接受的语言：

$T(M)=\{x\mid x:(q_0,z_0)\overset{*}{\vdash}_M(q,\gamma),\ \gamma\in\Gamma^*,\ q\in F\}$

例：PDA：$M=(\Sigma,Q,\Gamma,\delta,q_0,z_0,F)$ 接受语言 $L=\{wcw^R\mid w\in\{a,b\}^*\}$

其中，$Q=\{0,1\}$，$\Sigma=\{a,b,c\}$，$\Gamma=\{A,B\}$，$q_0=0$，$z_0=\#$，$F=\{1\}$，

δ：1) $\delta(0,a,\varepsilon)\vdash_M\{(0,A)\}$ ---

状态　输入　栈

语言类型	识别器类型
0型	图灵机
1型	线性带限自动机（确定的单带图灵机）
☆ 2型	下推自动机
☆ 3型	有限自动机

存储空间的差异

英语单词拼写检查

MAIN POINTS

设X为拼写错误的字符串，其长度为m，Y为X对应的正确的单词（答案），其长度为n，则X和Y的编辑距离 ed(X[m], Y[n]) 定义为：从字符串X转换为Y所需的插入、删除、替换和交换两个相邻的基本单位（字符）的最小个数。

递归计算

METHODS

构造一个确定的有限状态机

$R=(Q, A, \delta, q_0, F)$

Q表示状态集，A表示输入字母集

$\delta: Q \times A \rightarrow Q$

$q_0 \in Q$ 为初始状态

$F \subseteq Q$ 为终止状态集

SHORTCOMINGS

$L \subseteq A^*$ 表示R接受的语言

所有合法单词都是有限状态机的一条路径

寻找路径

（键树

便于查找树）

FIGURES

QUESTIONS

1）如果 $x_{i+1}=y_{j+1}$，则 $ed(X[i+1], Y[j+1]) = ed(X[i], Y[j])$

最后一个

2）如果 $x_i = y_{j+1}$，且 $x_{i+1} = y_j$，则

$$ed(X[i+1], Y[j+1]) = 1+\min\{ed(X[i-1], Y[j-1]),\ ed(X[i], Y[j+1]),\ ed(X[i+1], Y[j])\}$$

交换

删

加

3）else

$$ed(X[i+1], Y[j+1]) = 1+\min\{ed(X[i], Y[j]),\ ed(X[i], Y[j+1]),\ ed(X[i+1], Y[j])\}$$

其中 $ed(X[0], Y[j]) = j$　X长为0

$ed(X[i], Y[0]) = i$　Y长为0

$ed(X[-1], Y[j]) = ed(X[i], Y[-1])$

$= \max\{m, n\}$　（边界约定）

KEYWORDS