第三章　词法分析

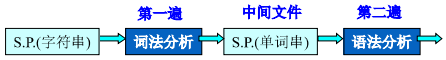
结构框架

词法分析程序的设计

两种实现方案

1.词法分析单独作为一遍

优点：结构清晰、各遍功能单一
缺点：效率低



2.词法分析程序作为子程序

词法分析的功能

（1）分析和识别单词及属性

（2）跳过各种分隔符，如空格，回车，制表符等

（3）删除注释

（4）进行词法检查，报告所发现的错误

（5）建立符号表

词法分析程序的输入、输出

int x=10,y=20,sum;

单词类别	编码
关键字	1
标识符	2
常数	3
运算符	4
分界符	5

单词种别	单词自身的值
1	int
2	指向x的符号表入口指针
4	=
3	10
5	,
2	指向y的符号表入口指针
4	=
3	20
5	,
2	指向sum的符号表入口指针
5	;

如何识别单词

逐个读取字符 → 词法规则 → 拼接字符形成单词

状态转换图
EBNF
有穷状态自动机
正规表达式
正规文法

词法分析程序的自动构造工具

3.2 PL/0编译程序中词法分析程序的设计和实现

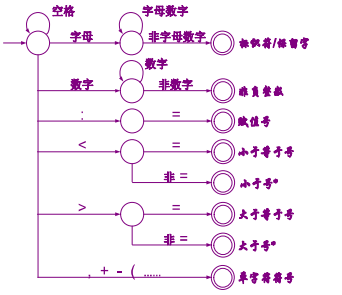
词法分析器的设计
1、给出程序设计语言的单词规范——单词表
2、对照单词表设计识别该语言所有单词的状态转换图
3、根据状态转换图编写词法分析程序

请大家课后自主学习！

单词的形式化描述工具

- 状态转换图
- EBNF
- 正规文法
- 正规表达式
- 有限状态自动机

适合于正规语言的描述及处理的形式模型



单词的形式化描述工具

- 状态转换图
- EBNF
- 正规文法
- 正规式
- 有限状态自动机

<无符号整数> ::= <数字>{<数字>}
<标识符> ::= <字母>{<字母>|<数字>}
<字母> ::= a | b | … | X | Y | Z
<数字> ::= 0 | 1 | 2 | … | 8 | 9
<保留字> ::= const | var | procedur | begin | end | odd
| if | then | call | while | do | read | write
<运算符> ::= + | - | * | / | = | # | < | <= | > | >= | :=
<界符> ::= (|) | , | ; | .

单词的形式化描述工具

- 状态转换图
- EBNF
- 正规文法
- 正规式
- 有限状态自动机

<标识符> → l | l <字母数字>
<字母数字> → l | d | l <字母数字> | d <字母数字>
<无符号整数> → d | d <无符号整数>
<运算符> → + | - | * | / | < <等号> | > <等号> | .....
<等号> → =
<界符> → , | ; | (|) | .....
其中l表示a~z的英文字母，d表示0~9的数字

单词的形式化描述工具

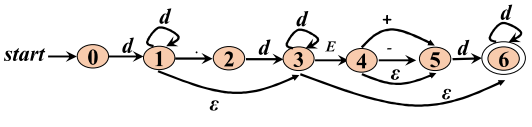
- 状态转换图
- EBNF
- 正规文法
- 正规式
- 有限状态自动机

(1|...|9)(0|...|9)*|0

d * (. dd * | ε)(e (+ | - | ε) dd * | ε)

单词的形式化描述工具

- 状态转换图
- EBNF
- 正规文法
- 正规式
- 有限状态自动机



正规集

程序语言中的合法单词的集合——正规集

表示正规集的工具——正则式

可由正规文法产生

正规文法

Chomsky 3型文法，是描述正规集的文法，可用于描述程序设计语言的词法规则。

<标识符> → l|l<字母数字>

<字母数字> → l|d|l<字母数字>|d<字母数字>

<无符号整数> → d|d<无符号整数>

<运算符> → +|-|*|/|<<等号>|><等号>|……

<等号> → =

<界符> → ,|;|(|)|……

其中l表示a~z中的任何一个英文字母，d表示0~9中的任何一个数字

正规式、正则集的递归定义

1、ε和Φ都是Σ上的正规式，它们所表示的正规集分别为：{ε}和Φ

2、任何a∈Σ，a是Σ上的正规式，它所表示的正规集为：{a}

3、假定 e_1 和 e_2 是Σ上的正规式，它们所表示的正规集分别记为$L(e_1)$和$L(e_2)$，则

$e_1|e_2$ 是Σ上的正规式，表示的正规集为 $L(e_1)\cup L(e_2)$

$e_1 \cdot e_2$ 是Σ上的正规式，表示的正规集为 $L(e1)\cdot L(e_2)$

e_1^* 是Σ上的正规式，表示的正规集为 $(L(e_1))^*$

4、任何Σ上的正规式和正规集均由1、2和3产生

正规式中的符号

正规式中的运算符

| | ----或（选择）
| • | ---连接
| * | ----闭包
| ( ) | ---括号

运算符的优先级

* > • > |

• 在正规式中可以省略

正规式相等 ⇔ 两个正规式表示的正规集相同

正规式服从的代数规律

设r，s，t均是正规式

（1）交换律：r|s = s|r

（2）结合律：r|(s|t) = (r|s)|t

(rs)t = r(st)

（3）分配律：(r|s)t = rt|st

（4）同一律：εr = rε = r

【例】设Σ={a,b}

正规式	正规集
ba*	所有以b为首后跟任意多个a的符号串
a(a|b)*	所有以a为首的a，b符号串
(a|b)*abb	所有以abb为尾的a，b符号串
(a|b)*(aa|bb) (a|b)*	所有含有两个相继的a或相继的b的符号串
(aa|ab|ba|bb) *	空串和任何长度为偶数的a，b符号串
(a|b)(a|b)(a|b) *	任何长度大于等于2的a，b符号串

思 考

$(a^*b^*)^*$ 是否等价 $(a|b)^*$ ✓

$a^*|b^*$ 是否等价 $(a|b)^*$ ✗

思 考

令 $\Sigma=\{d, ., e, +, -\}$，d 表示数字，则 Σ 上的正规式 r 表示什么？

$r = d^*(.\ dd^*|\varepsilon)(e(+|-|\varepsilon)\ dd^*|\varepsilon)$

6

6.66

6.6e6

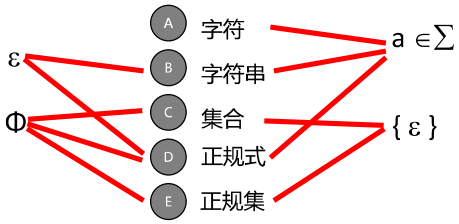
6.6e-6

正浮点数

单选题 1分 设置

此题未设置答案，请点击右侧设置按钮

思 考



提交

正规文法和正规式的等价性

1. 正规式 r ⇒ 正规文法 G

step1. 构造 S → r

step2. 不断利用下表中规则做变换，直到每个产生式最多含有一个终结符为止

	正规式	文法产生式
规则1	A→ xy	A→xB, B→y
规则2	A→ x*y	A→xA|y
规则3	A→ x|y	A→x, A→y

【例】求正规式 r = a(a|d)* 对应的正规文法

S→a(a|d)* ⟹ S→aA, A→(a|d)*

↓

A→(a|d)A | ε

↓

G[S]:
S→aA
A→aA|dA|ε

【例】求正规式 r=(a|b)(a|b|0|1)* 对应的正规文法

S→(a|b)(a|b|0|1)*

S→(a|b)A
A→(a|b|0|1)*

↓

A→aA|bA|0A|1A|ε

↓

G_1[S]:
S→aA|bA
A→aA|bA|0A|1A|ε

S→(a|b) | S (a|b|0|1)

↓

G_2[S]:
S→ a | b | Sa | Sb | S0 | S1

正规文法和正规式的等价性

2.正规文法⇒正规式

step1. 将每条产生式改写为正规式；

step2. 用代入法解正规式方程组，最后只剩下开始符号定义的正规式，其中不含V_N。

	文法产生式	正规式
规则1	A→xB, B→y	A=xy
规则2	A→xA|y	A=x*y
规则3	A→x, A→y	A=x|y

南京中医药大学 | 智信学院

	文法产生式	正规式
规则1	A→xB, B→y	A=xy
规则2	A→xA|y	A=x*y
规则3	A→x, A→y	A=x|y

$G_1[S]$:
S→aA|a
A→dA|d

⇒ S=aA|a, A= d*d ⇒ S=ad*d|a ⇒ S=ad*

$G_2[S]$:
S→aA|a
A→aA|dA|a|d

⇒ S=a(A|ε), A= (a|d)*|(a|d) ⇒ S=a(a|d)*

单选题 1分 设置

用正规式表示的变量声明：(int | float) id (, id)* 构造等价的文法

D→(int | float)L
L→ id (, id)*

⇓

D→ int L | float L
L→ L, id | id

⇓

$G_1[D]$:
D→ int L | float L
L→ L, id | id

$G_2[D]$:
D → int L | float L
L → id T
T → , L| ε

与正规式 a*b* 等价的文法是（ ）。

A G_1: S→ aS | bS |ε

B G_2: S→ aSb |ε

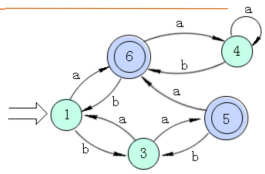
C G_3: S→ aS | Sb |ε

D G_4: S→ abS |ε

提交

有穷自动机（Finite Automata，FA）

- 由两位神经物理学家于1948年首先提出，是对一类处理系统建立的数学模型，具有一系列离散的输入输出信息和有穷数目的内部状态。



- 只要根据当前所处的**状态**和当前面临的**输入信息**就可以决定系统的**后继行为**。每当系统处理了当前的输入后，系统的内部状态也将发生改变。
- 引入有穷自动机这个理论，正是为词法分析程序的自动构造寻找特殊的方法和工具。

南京中医药大学 | 智信学院

有穷自动机的分类

有穷自动机
- **确定的有穷自动机（DFA）**
  (Deterministic Finite Automata)
- **不确定的有穷自动机（NFA）**
  (Nondeterministic Finite Automata)

确定的有穷自动机（DFA）

M=（K，Σ，f，S，Z）

K：有穷集，它的每个元素称为一个状态

Σ：有穷字母表，它的每个元素称为一个输入符号

S∈K，是唯一的初态

Z⊆K是一个终态集，终态也称可接受状态或结束状态

f：转换函数，是K×Σ→K上的单值映射：

$f(q_1,a)=q_2$

DFA中转移函数的表示

M:({0,1,2,3},{a,b},f, 0, {3})

f (0, a) =1　　f (0, b) =2
f (1, a) =3　　f (1, b) =2
f (2, a) =1　　f (2, b) =3
f (3, a) =3　　f (3, b) =3

状态转移函数 f 可用一矩阵来表示：

状态 \ 输入字符	a	b
0	1	2
1	3	2
2	1	3
3	3	3

DFA的确定性表现在状态转移函数是单值函数！

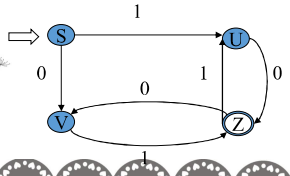
状态图

为识别单词而专门设计的有向图，是设计词法分析程序的一种好途径。

结点代表状态（有限个），用圆圈表示，对应文法的V_N

有向弧表示状态转移

弧上的标记表示在射出弧的结点状态下可能会出现的输入字符，为V_T

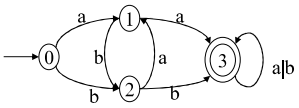


一张状态图只能有一个初态，至少有一个终态（用双圈表示）

DFA的状态转换图表示

状态 \ 输入字符	a	b
0	1	2
1	3	2
2	1	3
3	3	3

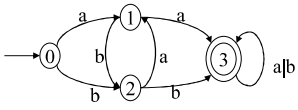
对应的状态图：



思 考

假定DFA有m个状态， 字母表 Σ 包含n个输入符号，

那么每个状态最多有几条有向弧?



DFA功能

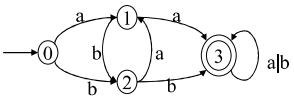
若存在一条从初始状态到某一终止状态的路径，且这条路径上所有弧的标记符号连接成符号串α，则称α为DFA M（接受）**识别**。

- 若DFA的初态结点同时为终态，则ε为DFA所识别。

DFA M 所识别的符号串的集合即为该M的语言L(M)

$L(M)=\{\alpha \mid f(S,\alpha)=S_n, S_n \in Z\}$

DFA功能



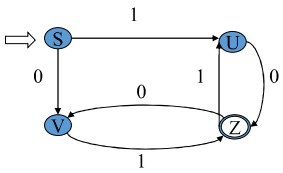
符号串abaab

δ（0，abaab）
=δ（1，baab）
=δ（2，aab）
=δ（1，ab）
=δ（3，b）
=3 (接受)

符号串abab

δ（0，abab）
=δ（1，bab）
=δ（2，ab）
=δ（1，b）
=2 (拒绝)

DFA功能



DFA能识别什么样的符号串?

M 所接受的语言（正规式） $R=(10|01)(10|01)^*$

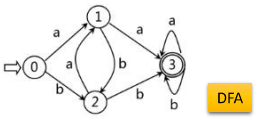
思考，下图两个DFA分别识别什么?

⇨ Ⓢ　　{ ε }

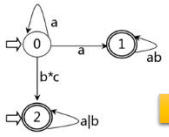
⇨ Ⓢ　　Φ

NFA

DFA是NFA的特例



DFA



NFA

1、NFA可以有多个初态
2、NFA弧上的标记可以是一个正规式
3、同一个正规式可能出现在同一个状态出发的多条弧上

不确定的有穷自动机（NFA）

N=（K，Σ，f，S，Z）

K：有穷集，它的每个元素称为一个状态
Σ：有穷字母表，它的每个元素称为一个输入符号
S ⊆ K，非空初态集
Z ⊆ K，终态集
f：转换函数，是 $K\times\Sigma^* \to 2^K$ 的映射，多值函数

某状态对于某个输入字符存在多个后继状态，即状态的转向是不确定的

NFA

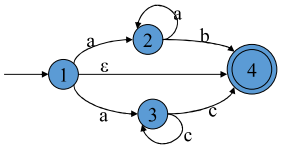
- 对于Σ*上的任何符号串α，若存在一条从某一初态到某一终态的通路，且该通路上所有弧的标记字符依次连接成的串等于α，则称α可以被NFA N所识别或接受。
- 若N的初态结点同时为终态，或者存在一条从初态到某个终态结点的ε通路，则ε为N所识别。

NFA N所能识别的符号串的全体记为L(N)，称为NFA N所识别的语言。

请画出该NFA对应的状态转换图

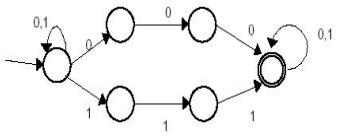
例： NFA N=({a,b,c} ,{1,2,3,4}, f,{1},{4})

状态 \ 符号	ε	a	b	c
1	{4}	{2，3}	Φ	Φ
2	Φ	{2}	{4}	Φ
3	Φ	Φ	Φ	{3，4}
4	Φ	Φ	Φ	Φ



NFA所接受的语言（正规式）： r= aa*b | ac*c | ε

请写出该自动机对应的正规式



能接受
000
111
1010001
110000001

不能接受
00
01100

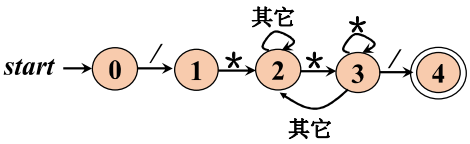
(0|1)*(000|111)(0|1)*

请构造满足条件的有穷自动机

假设字母表为{0,1}，字符串以1开头以0结尾

假设字母表为{0,1}，字符串中包含001子串

能够识别C语言注释/* */的DFA



有穷自动机的其他应用

❑用于某些重要软件的设计和构造

- 设计和检查数字电路行为的软件;
- 扫描如网页族等大规模文本以发现字、词或其它结构的出现频率的软件;
- 验证所有只有有限多个不同状态的系统的软件，这类系统包括通信协议和信息安全交换协议。

相关论文

[1]张居晓.非确定有穷自动机在盲文转码中的应用[J].计算机科学,2017,44(01):271-276.

[2]S. Prithi,S. Sumathi. LD 2 FA-PSO: A novel Learning Dynamic Deterministic Finite Automata with PSO algorithm for secured energy efficient routing in Wireless Sensor Network[J]. Ad Hoc Networks,2020,97.

3.4.3 NFA转换为等价的DFA

L为一个由NFA接受的集合，则存在一个接受L的DFA

NFA N —构造成一个→ DFA M

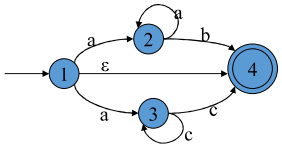
使得 L(M)=L(N)

DFA是NFA的特例。

将NFA转换成接受同样语言的DFA的算法——子集法

与某一NFA等价的DFA不唯一

NFA的确定化



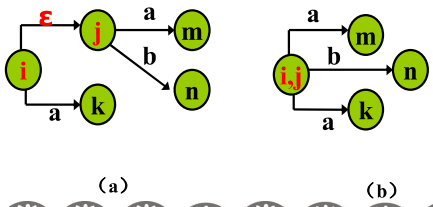
状态 \ 符号	ε	a	b	c
1	{4}	{2, 3}	Φ	Φ
2	Φ	{2}	{4}	Φ
3	Φ	Φ	Φ	{3, 4}
4	Φ	Φ	Φ	Φ

- 从NFA的矩阵表示中可以看出，表项通常是一状态的集合，而DFA的矩阵表中每个表项仅是一个状态。
- NFA到相应的DFA的构造的基本思路是：DFA的每一个状态对应NFA的一组状态。

转换需解决的问题

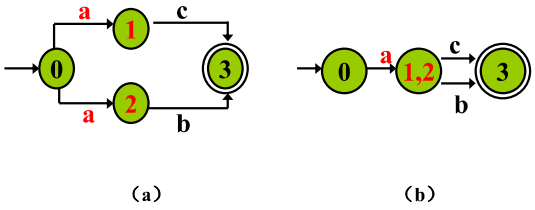
(1) ε 合并

如果有 $S_1 \xrightarrow{\varepsilon} S_2$ ，则把 S_2 合并到 S_1



转换需解决的问题

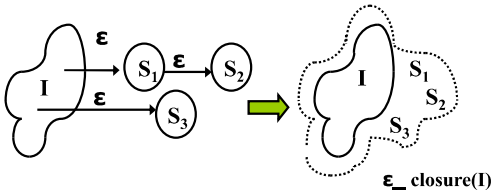
(2) 状态合并



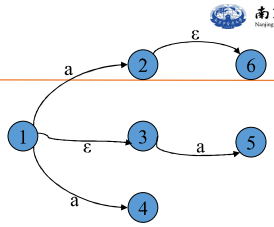
定义1 状态集合 I 的 ε-闭包，记为 ε-closure(I)

① 若q∈I，则q∈ε-closure(I)；

② 若q∈I，则从q出发经过任意条ε弧而能到达的任何状态q'都属于ε- closure(I) 。



右图所示NFA中，令I={1}，则 ε-closure(I)=?



根据定义：

ε-closure(I)={1，3}

课堂练习：

令I={1,2}，

求ε-closure(I)= ？

定义2 I_a 子集

✓ 由 I 中的状态出发，经过一条a弧可能到达的状态的集合称为 move(I,a)

✓ I_a= ε-closure (move(I,a))

课堂练习：I={1,2,3}

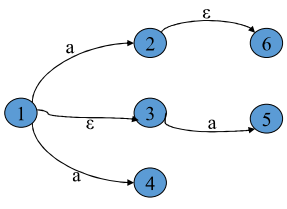
求I_a =?

令 I={1}

则 I_a=ε-closure(move(I,a))

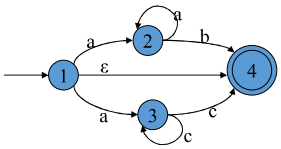
=ε-closure(f(1,a))

=ε-closure({2,4}) = {2, 4, 6}



NFA转换为DFA的思想

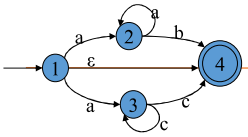
- 将从NFA的初态S出发经过任意条ε弧所能到达的状态作为DFA的初态S'
- 从S'出发，把遇到输入符号a所转移到的后继状态集作为DFA的新状态
- 如此重复，直到不再有新的状态出现为止



课堂练习：

令I={1}，设S'=ε-closure(I)

求 S'=？ S'_a =?



1. 构造一张表，它共有 | Σ | +1列
2. 第一行第一列为 ε-closure({S})
3. 求 I_a
4. 重复步骤3
5. 将状态子集重新命名

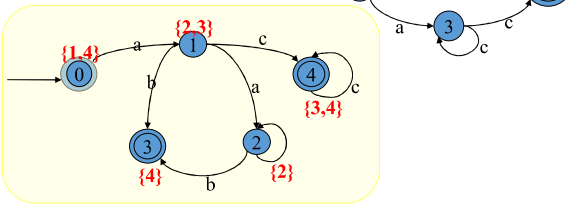
I	I_a	I_b	I_c
{1,4}	{2,3}	Φ	Φ
{2,3}	{2}	{4}	{3,4}
{2}	{2}	{4}	Φ
{4}	Φ	Φ	Φ
{3,4}	Φ	Φ	{3,4}

对状态转换矩阵重新编号

DFA M状态转换矩阵：

状态 \ 符号	a	b	c
0	1	–	–
1	2	3	4
2	2	3	–
3	–	–	–
4	–	–	4

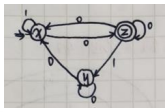
DFA M的状态图



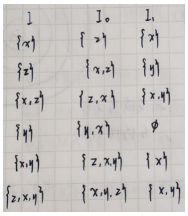
☆ 注意：包含原初始状态1的状态子集为DFA M的初态
包含原终止状态4的状态子集为DFA M的终态。

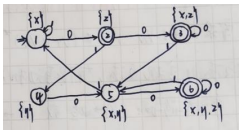
作业

2. 已知 NFA=({x,y,z},{0,1},M,{x},{z}) 其中：
$M(x,0)=\{z\}, M(y,0)=\{x,y\}, M(z,0)=\{x,z\}, M(x,1)=\{x\}, M(y,1)=\varnothing, M(z,1)=\{y\}$，构造相应的 DFA。



标清初态、终态





作业

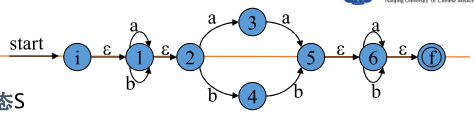
8. 给出下述正规文法所对应的正规式：

$S \to 0A \mid 1B$

$A \to 1S \mid 1$

$B \to 0S \mid 0$

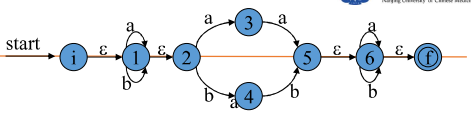
(01|10)(01|10)*



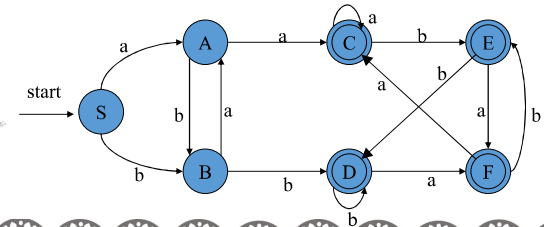
找出等价的DFA的初态S

		Ia		I_b	
{i,1,2}	S	{1,2,3}	A	{1,2,4}	B
{1,2,3}	A	{1,2,3,5,6,f}	C	{1,2,4}	B
{1,2,4}	B	{1,2,3}	A	{1,2,4,5,6,f}	D
{1,2,3,5,6,f}	C	{1,2,3,5,6,f}	C	{1,2,4,6,f}	E
{1,2,4,5,6,f}	D	{1,2,3,6,f}	F	{1,2,4,5,6,f}	D
{1,2,4,6,f}	E	{1,2,3,6,f}	F	{1,2,4,5,6,f}	D
{1,2,3,6,f}	F	{1,2,3,5,6,f}	C	{1,2,4,6,f}	E

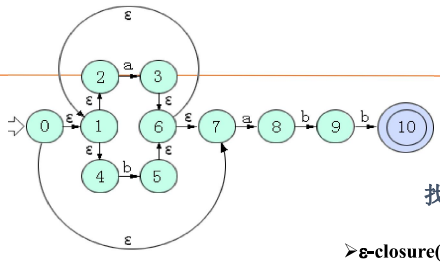
南京中医药大学 Nanjing University of Chinese Medicine | 智信学院



等价的DFA



南京中医药大学 Nanjing University of Chinese Medicine | 智信学院



找出等价的DFA的初态

- ε-closure(0)={0,1,2,4,7}
- 令A={0,1,2,4,7}, move(A,a)={3,8}
- ε-closure({3,8})={1,2,3,4,6,7,8}

南京中医药大学 Nanjing University of Chinese Medicine | 智信学院

DFA的化简

- 如果不同的DFA能识别相同的语言，则称它们是等价的DFA。
- 在等价的DFA中，如果某一个DFA的状态数是最少的，则这个DFA是最简的。
- 对于任一个DFA，存在一个唯一的状态最少的等价的DFA

最简的DFA ⇔ 它没有多余状态和等价状态

南京中医药大学 Nanjing University of Chinese Medicine | 智信学院

定义1 **多余状态**：从开始状态出发,任何输入串也不能到达的状态

例:

	0	1
S_0	S_1	S_5
S_1	S_2	S_7
S_2	S_2	S_5
S_3	S_5	S_7
S_4	S_5	S_6
S_5	S_3	S_1
S_6	S_8	S_0
S_7	S_0	S_1
S_8	S_3	S_6

画状态图可以看出S_4,S_6,S_8为不可达状态应该消除

	0	1
S_0	S_1	S_5
S_1	S_2	S_1
S_2	S_2	S_5
S_3	S_5	S_1
S_5	S_3	S_1
S_7	S_0	S_1

南京中医药大学 Nanjing University of Chinese Medicine | 智信学院

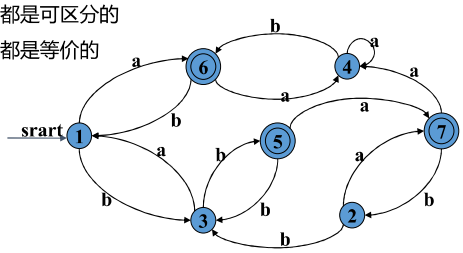
定义2 等价状态 ⟺ 状态s和t的等价条件

① 状态S和T必须同时为终态或非终态

② 对于所有输入符号，S和T必须转换到等价的状态里

DFA最小化算法的基本思想

- 把DFA的状态划分成一些不相交的子集
- 任何不同的两个子集的状态都是可区分的
- 同一子集中的任何两个状态都是等价的



南京中医药大学 Nanjing University of Chinese Medicine | 智信学院

（1）将所有状态分成两个子集：终态集和非终态集

（2）对每个集合中的符号分别用输出字母去查看它们到达状态的集合是否在同一个集合中

（3）若不在同一个集合，将它们划分在不同的集合中

（4）直到不能再划分为止

分割法

解：(一)区分终态与非终态

	a	b	区号
1	6	3	1
2	7	3	
3	1	5	2
4	4	6	
5	7	3	3
6	4	1	
7	4	2	

南京中医药大学 Nanjing University of Chinese Medicine | 智信学院

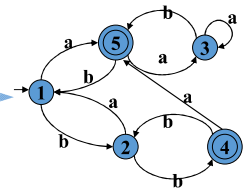
	a	b	区号
1	6	3	1
2	7	3	
3	1	5	2
4	4	6	3
5	7	3	4
6	4	1	
7	4	2	

	a	b	区号
1	6	3	1
2	7	3	
3	1	5	2
4	4	6	3
5	7	3	4
6	4	1	
7	4	2	5

南京中医药大学 Nanjing University of Chinese Medicine | 智信学院

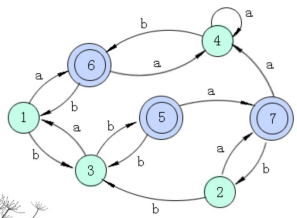
将区号代替状态号

	a	b
1	5	2
2	1	4
3	3	5
4	5	2
5	3	1

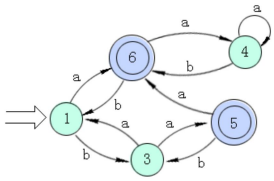


化简后的有穷自动机具有较少的状态，实现起来更加简洁。

南京中医药大学 Nanjing University of Chinese Medicine | 智信学院



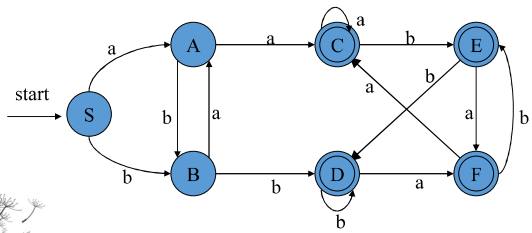
化简前



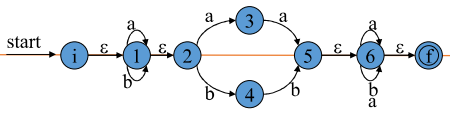
化简后

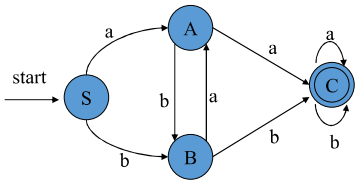
南京中医药大学 Nanjing University of Chinese Medicine | 智信学院

DFA最小化

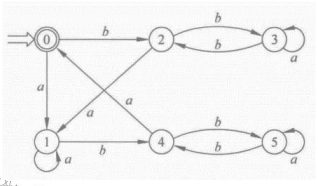


CDEF等价

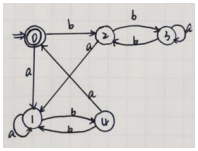




课内习题



1、这个自动机是DFA还是NFA？
2、确定化
3、最小化



正规式和有穷自动机的等价性

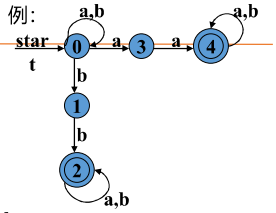
（1）对于字母表Σ上的NFA M，可以构造一个Σ上的正规式R，使得L(R)=L(M)；

（2）对于字母表Σ上的每个正规式R，可以构造一个Σ上的NFA M，使得L(M)=L(R)。

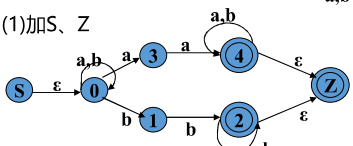
1. NFA M ⇒ 正规式 R

(1)在M上加两个结点S,Z，从S结点用ε弧到M的所有初态，从M的所有终态用ε到Z结成与M等价的M'，M'只有一个初态S和一个终态Z。

例：



解: (1)加S、Z



1. NFA M ⇒ 正规式 R

(2)逐步消去M' 中的所有结点，直至只剩下S和Z结点。

在消结过程中,逐步用正规式来标记弧, 规则如下:

1.对于 

代之为

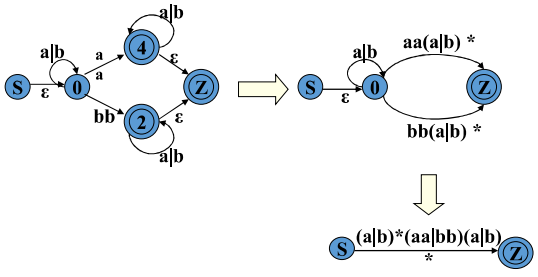
2.对于 

代之为

3.对于 

代之为

(2)消除M中的所有结点



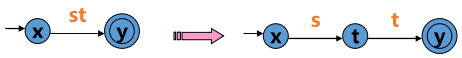
2.正规式 R ⇒ NFA M

（1）对NFA M构造一个广义的状态图，其中只有一个初态S和终态Z，连接S和Z的有向弧标记为正规式。

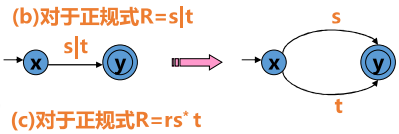
（2）对正规式依次进行分解，分解的过程是一个不断加入结点和弧的过程，直到转换图上的所有弧标记上都是字母表Σ上的元素或ε为止。

若s,t为Σ上的正规式

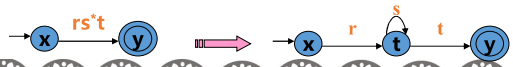
(a)对于正规式R=st



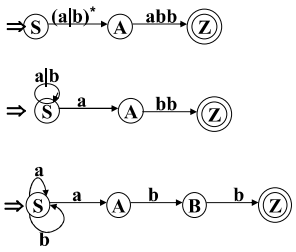
(b)对于正规式R=s|t



(c)对于正规式R=rs* t



例: 为r=(a|b)*abb 构造NFA，使得L(N)=L(r)

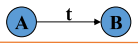


正规文法和有穷自动机的等价性

（1）对于NFA M，存在一个右线性文法（左线性文法）G，使得L(G)=L(M)；

（2）对于右线性文法（左线性文法）G，可以构造一个NFA M，使得L(M)=L(G)。

1. NFA ⇒ 正规文法



（1）NFA的字母表为文法的终结符号集；

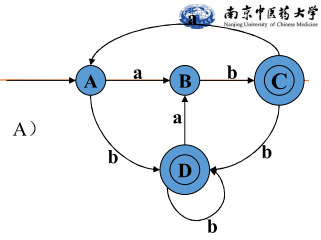
（2）NFA的状态集为文法的非终结符号集；

（3）NFA的初态对应于文法的开始符号；

（4）NFA的转换函数 f(A,t)=B，写成一个产生式 A→tB；

（5）对NFA的终态 Z，增加一个产生式 Z→ε。

例：给出与该NFA等价的正规文法G



G=（{A，B，C，D}，{a，b}，P，A）

其中P: A→ aB
A→ bD
B→ bC
C→ aA
C→ bD
C→ ε
D→ aB
D→ bD
D→ ε

2. 正规文法 ⇒ NFA

（1）文法的终结符号集为NFA的字母表；

（2）文法的非终结符号集为NFA的状态集；

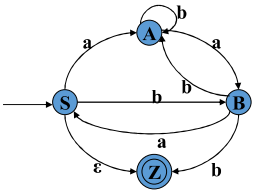
（3）文法的开始符号作为NFA的初态；

（4）对文法中形如A→tB的产生式，其中t为终结符或ε，A和B为非终结符，构造NFA的一个转换函数f(A,t)=B；

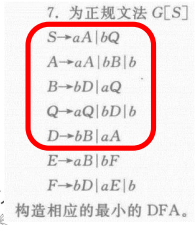
（5）对文法中形如A→t的产生式，构造NFA的一个转换函数f(A,t)=Z。

例：求与文法G[S]等价的NFA

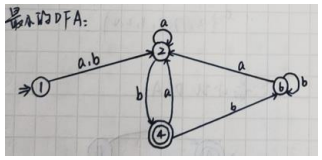
G[S]: S→aA|bB|ε
A→aB|bA
B→aS|bA|b



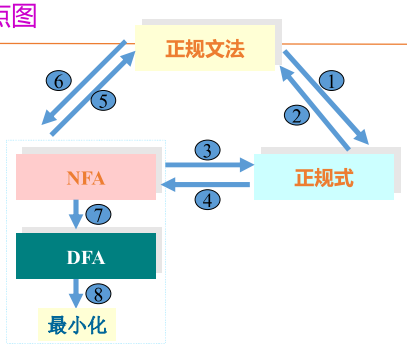
课内习题


7. 为正规文法 G[S]
S→aA|bQ
A→aA|bB|b
B→bD|aQ
Q→aQ|bD|b
D→bB|aA
E→aB|bF
F→bD|aE|b
构造相应的最小的 DFA。

1、构造出对应的NFA
2、确定化
3、最小化



本章核心知识点图



词法分析程序的自动构造工具

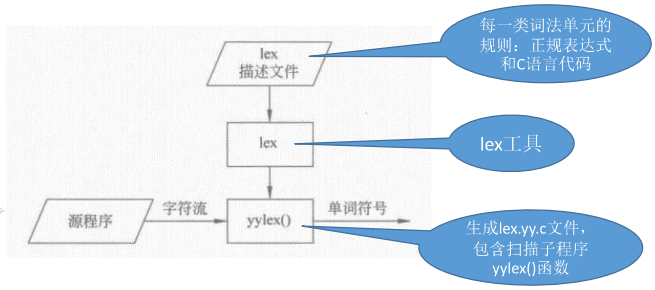
正规式 → NFA → DFA → 识别

每一类词法单元都对应一个正规式，所有正规式以文本方式作为自动构造工具的输入。

自动构造工具将每一个正规式转换成NFA，必要时将NFA确定化为DFA，再进行最小化。

生成扫描程序的代码，选择对每一类词法单元所对应的有限自动机依次模拟运行，并从当前输入符号序列中识别下一个单词，然后返回相应的单词类别及值。

词法分析自动构造工具——LEX



实例1

- 编写程序，实现下述LEX源程序的功能
  - 辅助定义：

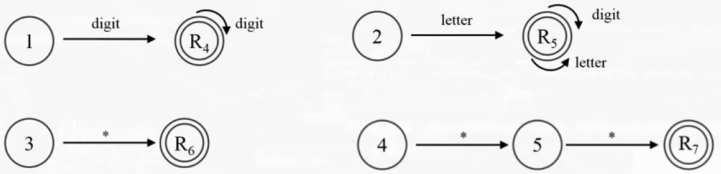
(1)digit→0|1|...|9　　(2)letter→A|B|...|Z

  - 识别规则:

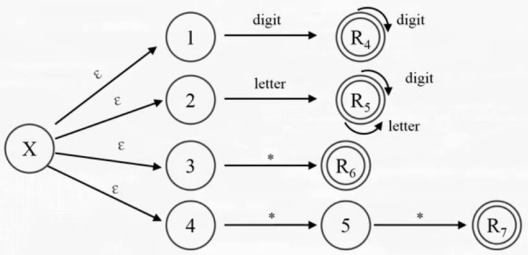
(1)digit(digit)* {Return(4, val)}　　(2)letter(letter|digit)* {Return(5，Token)}

(3)　*　{Return(6, _)}　　(4)　**　{Return(7, _)}

1、各识别规则的NFA为：

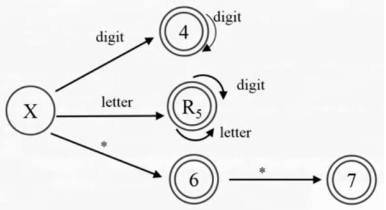


2、加入新初态X，构成NFA M整体



3、确定化为DFA M′

I	I_{digit}	I_{letter}	I_*
{x,1,2,3,4}	{R_4}	{R_5}	{R_6，5}
{R_4}	{R_4}		
{R_5}	{R_5}	{R_5}	
{R_6，5}			{R_7}
{R_7}			



4、写出实现其功能的程序

```
PROGRAM LEX(input,output)
BEGIN   TOKEN:=' ' ; getchar;
    CASE char OF
      '0' .. '9' : [while char IN[ '0' .. '9' ]DO
                      BEGIN TOKEN:=TOKEN+char; getchar
                      END; retract;
                      IF VAL(token,value) THEN return(4,value)];
      'A' .. 'Z' :[while char IN[ 'A' .. 'Z' ,' 0' .. '9' ]DO
                      BEGIN TOKEN:=TOKEN+char; getchar
                      END; retract;
                      Return(5,token)];
      '*' :         [getchar;
                      IF char=' *'   THEN Return(7,-)
                      ELSE [retract;Return(6,-)];
    ELSE:           ERROR;
    END{of case};
END;
```

南京中医药大学 | 智信学院

参考资料

中南大学 陈志刚教授 《编译原理》

https://www.icourse163.org/course/CSU-1206894807?from=searchPage

06 词法分析_4

课时

3.4.1 词法分析器的自动生成

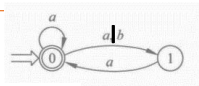
3.4.2 词法分析程序实现实例

3.4.3 小结

小结

南京中医药大学 | 智信学院

CH3习题



1、将右图的NFA确定化并最小化后，写出其识别的正规式。

2、构造一个最简DFA，使它接受这样的0,1组成的字符串：每个0都有1个1直接跟在其右边。