编译原理

北方工业大学计算机学院
School of Information Science and Technology,
North China University of Technology
束劼
shujie@ncut.edu.cn
瀚学楼1122，88801615

第三章 词法分析

第三章 词法分析

- 本章目录

第三章 词法分析

- 大纲要求

1. 掌握：词法分析器的设计与实现方法，基于状态转换图的词法分析器的构造算法。
2. 理解：状态转化图的作用与画法。
3. 了解：对于词法分析器的要求；正规文法与有限自动机的等价性，正规式与有限自动机的等价性；词法分析器的自动产生工具LEX的基本作用。

3.3 正规式和有限自动机

3.3 正规式和有限自动机

3.3.5 正规式与有限自动机的等价性

3.3.5 正规式与有限自动机的等价性

- 正规式与有限自动机的等价性

① 1. 对任何FA M，都存在一个正规式r，使得L(r)=L(M)。

② 2. 对任何正规式r，都存在一个FA M，使得L(M)=L(r)。

对状态转换图概念加以拓广，令每条弧可用一个正规式作标记。

3.3.5 正规式与有限自动机的等价性

- 正规式构造相应有限自动机的方法

  假设r和s是正规文法的描述，则N(r)和N(s)是NFA对于r和s的构造图。

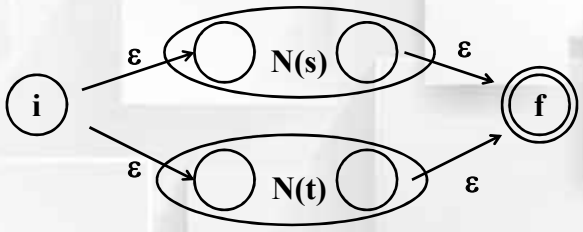
  我们分4步来描述构造方法。

9

3.3.5 正规式与有限自动机的等价性

- 正规式构造相应有限自动机的方法

  ① 假如r=s | t, 则N(r)是NFA对于r的构造图，N(s)和 N(t)分别是s和t的NFA图。则 N(s)和N(t)分别有一个开始结点和一个终态结点，并且有一条ε边指向这个开始结点，以及一条从终态结点指出的ε边。

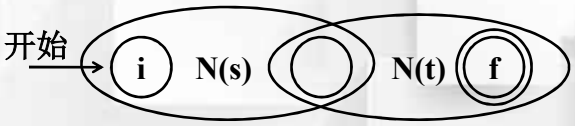


10

3.3.5 正规式与有限自动机的等价性

- 正规式构造相应有限自动机的方法

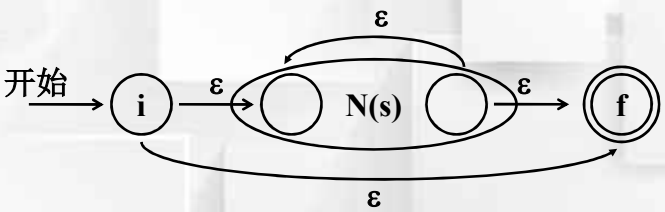
② 假如r=s t (连接积), N(s)的开始结点跟N(r)的开始结点是同一个，N(t)的终态结点则是唯一的终态结点。我们把N(s)的终态结点和N(t) 开始结点合并。



3.3.5 正规式与有限自动机的等价性

- 正规式构造相应有限自动机的方法

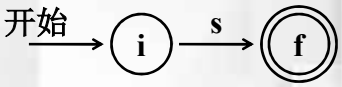
③ 假如r=s *，N(r)有个开始结点和仅有一个终态结点，从i到f可以直接由一条ε边（s*包含ε，即$\mathbf{L(s)^0}$），或者从i出发一条ε边到N(s)的开始结点。然后，N(s)的终态结点出发一条ε边指回N(s) 的开始结点。



3.3.5 正规式与有限自动机的等价性

- 正规式构造相应有限自动机的方法

  ④ 假如r=(s)，L(r) = L(s)，N(r)和N(s)一样。



13

3.3.5 正规式与有限自动机的等价性

- 正规式构造相应有限自动机的方法

  例题：构造正规文法r=(a | b)* abb的NFA图。

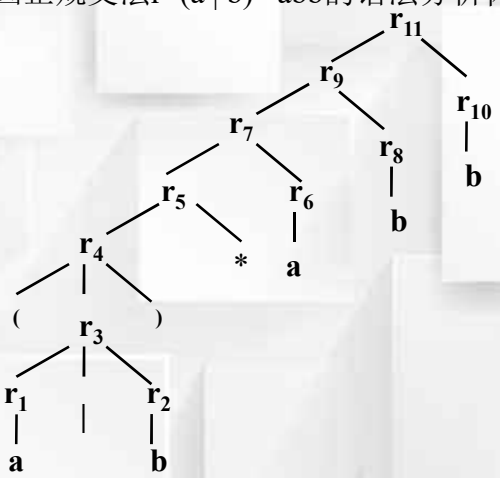
  ① 画语法分析树

  ② 按语法分析树从自底向上，分别使用前面的方法画单个的NFA图，并按顺序衔接开始结点和终态结点。
  (注意：符号不用画)

14

3.3.5 正规式与有限自动机的等价性

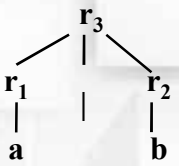
① 画正规文法r=(a | b)* abb的语法分析树。



15

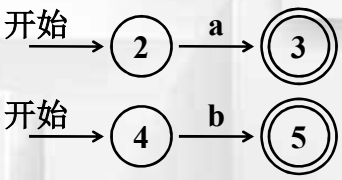
3.3.5 正规式与有限自动机的等价性

② 画$\mathbf{r_1}$和$\mathbf{r_2}$的NFA图。 r=(a | b)* abb



正规文法的最初起点和最后的终态结点。

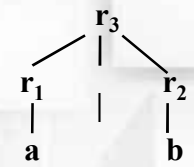
单独画N($\mathbf{r_1}$)和N($\mathbf{r_2}$)。



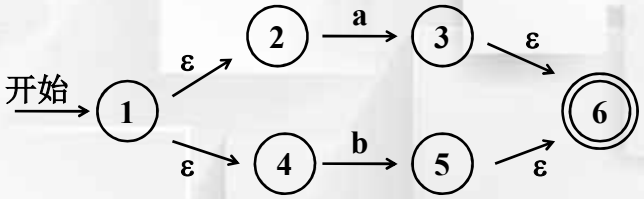
16

3.3.5 正规式与有限自动机的等价性

③ 画$\mathbf{r_3}$的NFA图。 r=(a | b)* abb

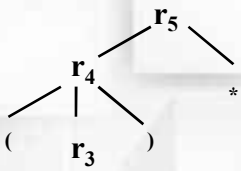


衔接N($\mathbf{r_1}$)和N($\mathbf{r_2}$)和正规文法的最初起点和最后的终态结点形成N($\mathbf{r_3}$)。

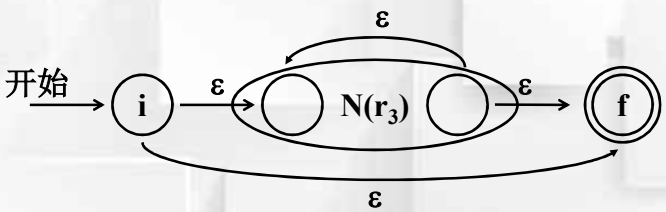


3.3.5 正规式与有限自动机的等价性

④ 画$\mathbf{r_5}$的NFA图。 r=(a | b)* abb

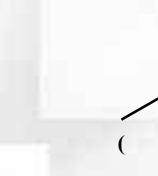


单独画N($\mathbf{r_5}$)，($\mathbf{r_3}$)*的NFA图。

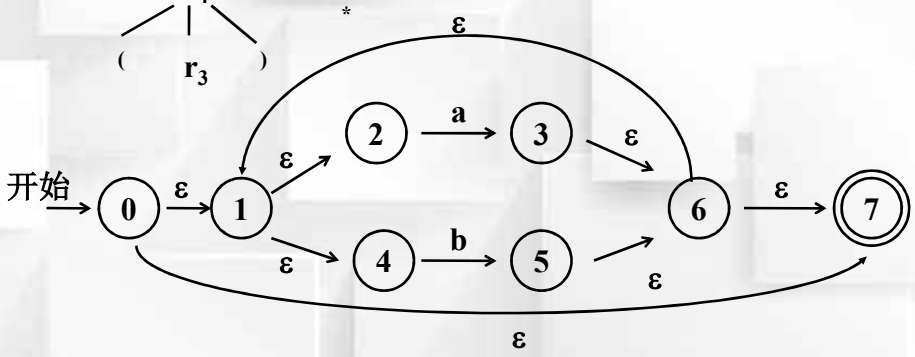


3.3.5 正规式与有限自动机的等价性

⑤ 画$\mathbf{r}_5$的NFA图。 r=(a | b)* abb

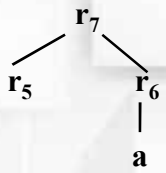


把N($\mathbf{r}_3$)与N($\mathbf{r}_5$) 衔接起来。

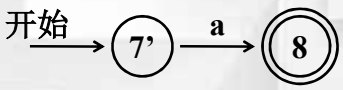


3.3.5 正规式与有限自动机的等价性

⑥ 画$\mathbf{r}_6$的NFA图。 r=(a | b)* abb

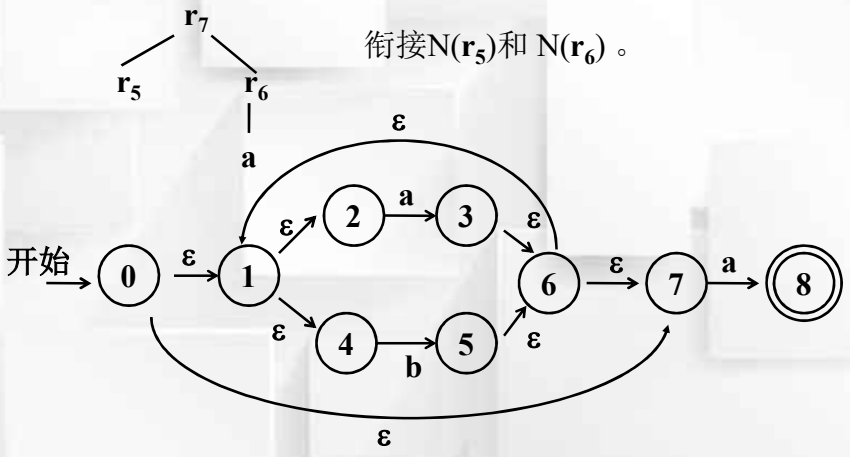


单独画N($\mathbf{r}_6$)的NFA图。



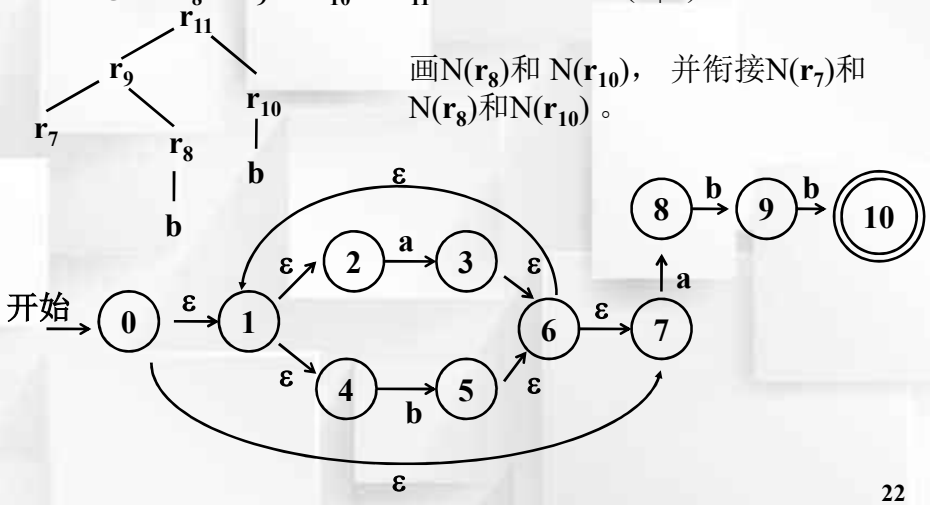
3.3.5 正规式与有限自动机的等价性

⑦ 画$\mathbf{r}_7$的NFA图。 r=(a | b)* abb



3.3.5 正规式与有限自动机的等价性

⑧ 画$\mathbf{r}_8$、 $\mathbf{r}_9$ 、 $\mathbf{r}_{10}$ 、 $\mathbf{r}_{11}$的NFA图。 r=(a | b)* abb



3.3.6 确定有限自动机的化简

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  DFA M寻找一个状态数比M少的DFA M’，使得L(M)=L(M’) 。

  假设s和t为M的两个状态，称**s和t等价**:如果从状态s出发能读出某个字α而停止于终态，那么同样，从t出发也能读出α而停止于终态；反之亦然。

  如果M的两个状态不等价，则称这两个状态是**可区别的**。

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  假设s和t为M的两个状态，某个字符串可区别s和t，从s和t这两个结点出发形成以这个字符串为边的两条路径，并且最后有且仅有一条路径的终态是终态结点。

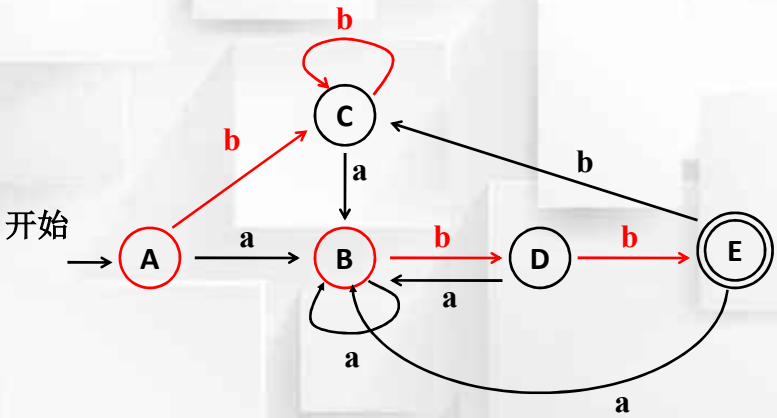
  则称这个**字符串可区别s和t**两个状态。

25

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  例：字符串bb可区别结点A和结点B



26

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  对一个DFA M最少化的基本思想:

  把M的状态集划分为一些不相交的子集，使得任何两个**不同子集**的状态都是**可区别**的，而**同一子集**的任何两个状态都是**等价**的。最后，在每个子集中选出一个代表，同时消去其他等价状态。

27

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  输入: DFA M有状态集合S，输入字母Σ，开始结点s_0，终态结点集合F；

  输出： DFA M’有状态集合S’，L(M’) = L(M)，S’ 状态数量少于S；

  化简步骤：4步

28

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  方法步骤：

  ① 首先，把S划分为终态结点和非终态结点两个子集，形成基本分划Π。例如子集F(终态结点)和S-F(非终态结点)，子集通用G表示。

  ② 形成新的划分Π_{new}。 循环

  例如划分后的Π_{new}有4个子集，分别是F_1(终态结点), F_2(终态结点), S_1 (非终态结点), S_2 (非终态结点) 。

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  ② 划分子集方法：

  假如每个子集用G表示， Π_{new} 为初始划分后的子集G

  For (Π中的每个子集G){

  取出状态s和t或更多；

  if(从s和t出发的a边连接的结点，都在同一个子集G中)

  s和t仍然在同一个子集G；

  else G被划分为两个子集，形成新的划分；

  更换Π_{new} 中的子集为新的子集划分；

  }

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简
  - ③ 如果$\Pi_{new} = \Pi$，则$\Pi_{final} = \Pi$， 继续第4步；否则继续第2步；
  - ④ 从Π_{final} 的每个子集中，挑选一个状态作为当前子集的代表，这些代表状态属于DFA M'状态集合S'。化简后的状态结点挑选遵循一定规则。

31

3.3.6 确定有限自动机的化简

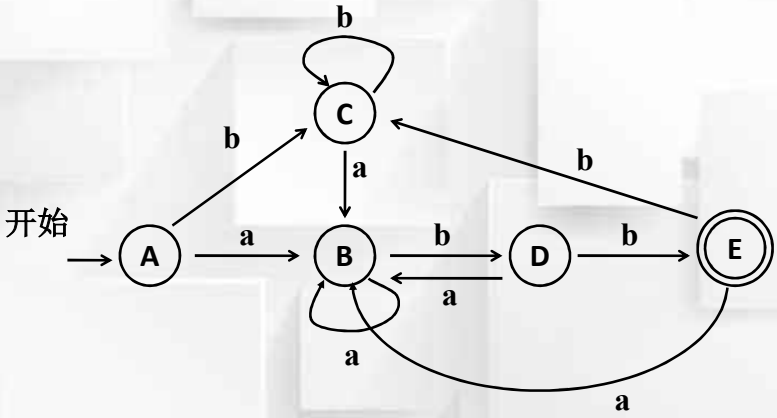
- 确定有限自动机的化简
  - ④ 挑选化简后状态结点遵循的规则：
  - a. 划分后的子集中，如果某个子集含有原始DFA M的开始结点s_0，则该s_0 自动成为当前子集的代表，也是DFA M'的开始结点；
  - b. 划分后的子集中，如果某个子集含有原始DFA M的终态结点，则该终态结点自动成为当前子集的代表，也是DFA M'的终态结点；
  - c. 对于每个子集的代表，删去其它一切等价的状态，并把射向其它状态的箭弧改为射向这个作为代表的状态。

32

3.3.6 确定有限自动机的化简

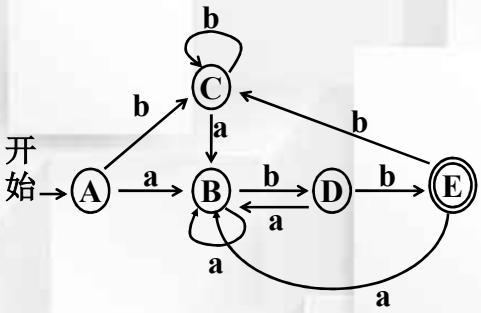
- 确定有限自动机的化简

  例：对当前的DFA进行化简



33

3.3.6 确定有限自动机的化简

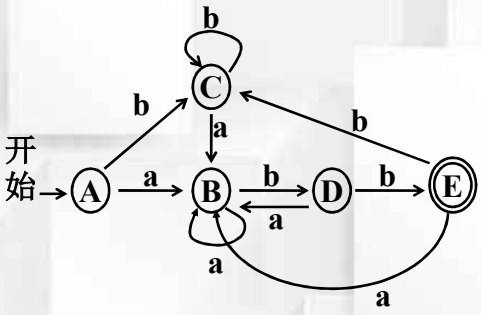


- 第一步，把S划分为终态结点和非终态结点两个子集，形成基本分划Π。

  终态结点子集{E}，非终态结点子集{A, B, C, D}

34

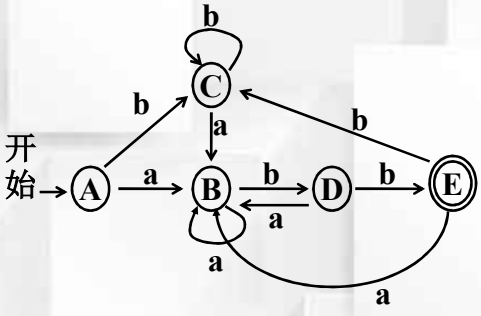
3.3.6 确定有限自动机的化简



- 第二步，形成新的划分Π_{new}。对两个子集{E}、 {A, B, C, D}**循环第一次**，以**a**边进行划分

  终态结点子集{E}，非终态结点子集{A, B, C, D}

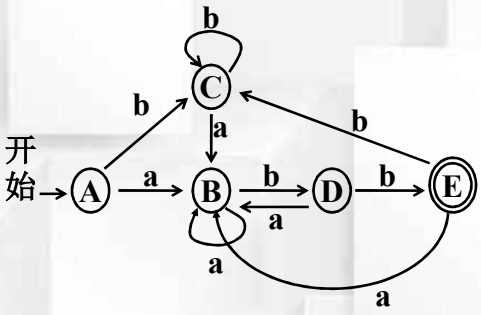
3.3.6 确定有限自动机的化简



- 第二步，形成新的划分Π_{new}。对两个子集{E}、 {A, B, C, D}**循环第一次**，以**b**边进行划分

  终态结点子集{E}，非终态结点子集{A, B, C}, {D}

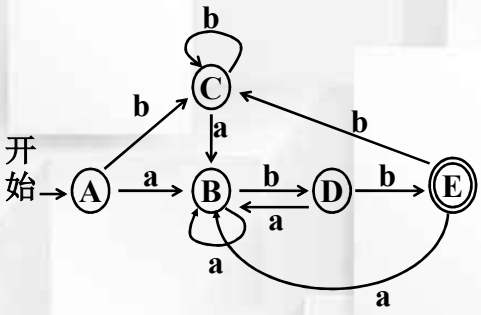
3.3.6 确定有限自动机的化简



- 第二步，形成新的划分Π_{new}。对两个子集{E}、{A, B, C, D}**循环第二次**，以**a**边进行划分

终态结点子集{E}，非终态结点子集{A, B, C}, {D}

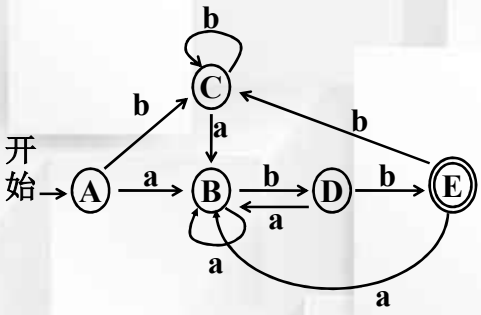
3.3.6 确定有限自动机的化简



- 第二步，形成新的划分Π_{new}。对两个子集{E}、{A, B, C, D}**循环第二次**，以**b**边进行划分

终态结点子集{E}，非终态结点子集{A, C}, {B}, {D}

3.3.6 确定有限自动机的化简



- 第二步，形成新的划分Π_{new}。对两个子集{E}、 {A, B, C, D}**循环第三次**，以**a**、**b**边进行划分

  终态结点子集{E}，非终态结点子集{A, C}, {B}, {D}

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  第三步，Π_{final} = 终态结点子集{E}，非终态结点子集{A, C}, {B}, {D}

  第四步，子集的代表结点，A是DFA M的开始结点，是{A, C}的代表，也是DFA M'的开始结点。

  终态结点子集{E}，非终态结点子集{A}, {B}, {D}。

3.3.6 确定有限自动机的化简

- 确定有限自动机的化简

  第四步， 形成化简后的状态转换矩阵

  终态结点子集{E}，非终态结点子集{A}, {B}, {D}

State	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

化简

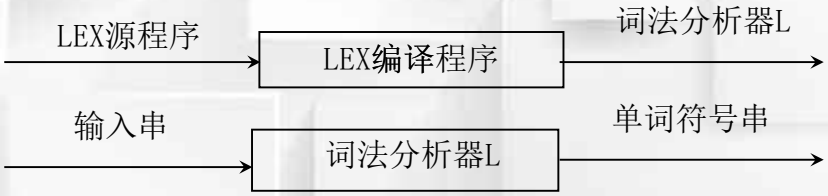
State	a	b
A	B	A
B	B	D
D	B	E
E	B	A

41

3.4 词法分析器的自动产生

3.4 词法分析器的自动产生

- 词法分析器的自动产生
- 第一种方法是用手工方式，即根据识别语言单词的状态转换图，使用某种高级语言，例如C语言直接编写词法分析程序。
- 第二种方法是利用词法分析程序的自动生成工具LEX自动生成词法分析程序。



43

3.4 词法分析器的自动产生

- 词法分析器的自动产生

高级语言的词法分析器的自动生成器——如LEX。

LEX是一个广泛使用的工具，UNIX系统中使用lex命令调用。它用于构造各种各样语言的词法分析程序。

44

3.4 词法分析器的自动产生

- 词法分析器的自动产生

  一个LEX源程序主要包括两部分。一部分是正规式，另一部分是识别规则。

  1、正规式

  letter→A|B|C|…|Z|a|b|c|…|z

  digit→0|1|2|…|9

  identifier→letter_(letter_|digit)*

  integer→digit(digit)*

3.4 词法分析器的自动产生

- 词法分析器的自动产生

  2、识别规则

正规式	动作描述
P_1	$\{A_1\}$
P_2	$\{A_2\}$
……	
P_n	$\{A_n\}$

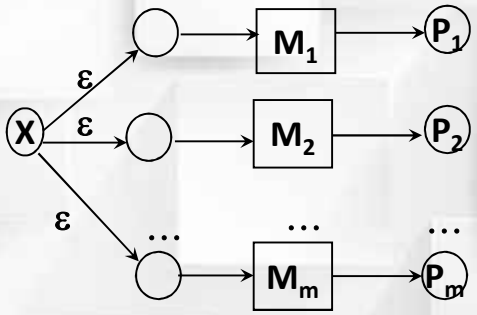
  P_i含有所有字母表中的字符，以及正规式左部所定义的任何标识符

3.4 词法分析器的自动产生

- LEX的工作过程：
  - 首先，对每条识别规则P_i构造一个相应的非确定有限自动机M_i；
  - 然后，引进一个新初态X，通过ε弧，将这些自动机连接成一个新的NFA；
  - 最后，把M确定化、最小化，生成该DFA的状态转换表和控制执行程序。

3.4 词法分析器的自动产生

- LEX的工作过程：



在等价的DFA M中，到达一个DFA的终态时并不停止，会继续工作下去，以便寻找更长的匹配，直到无法继续前进为止。

3.4 词法分析器的自动产生

- LEX的工作过程：
  正规式用于描述单词的结构十分简洁方便。而把一个正规式转换为一个NFA，进而转换为相应的DFA，这个NFA或DFA正是识别该正规式所表示的语言的句子的识别器。基于这种方法来构造词法分析程序。

3.4 词法分析器的自动产生

- 词法分析程序的设计技术可应用于其它领域，例如查询语言以及信息检索系统等，这种应用领域的程序设计特点是，通过字符串模式的匹配来引发动作，LEX可以看成是一个模式动作语言。

- 词法分析程序的自动构造工具也广泛应用于许多方面，例如用以生成一个程序，可识别印刷电路板中的缺陷，又如开关线路设计和文本编辑的自动生成等。

第三章 小结

第三章 小结

3.3 正规式与有限自动机
3.3.4 正规文法与有限自动机的等价性
3.3.5 正规式与有限自动机的等价性
3.3.6 确定有限自动机的化简
3.4 词法分析器的自动产生

Coursework

3.1 给定右线性文法G：

S→0S｜1S｜1A｜0B

A→1C｜1

B→0C｜0

C→0C｜1C｜0｜1

请给出文法G对应的NFA。

Coursework

3.2 构造一个DFA，它接受∑={0，1}上所有满足如下条件的字符串：每个1都有0直接跟在右边。

Coursework

3.3 一个人带着狼、山羊和白菜在一条河的左岸。有一条船，大小正好能装下这个人和其他三件东西中的一件。人和他的随行物都要过到河的右岸。人每次只能将一件东西摆渡过河。但若人将狼和羊留在同一岸而无人照顾的话，狼将把羊吃掉。类似地，若羊和白菜留下来无人照看，羊将会吃掉白菜。请问是否有可能渡过河去，使得羊和白菜都不被吃掉？如果可能，请用有限自动机写出渡河的方法。

提示：每个角色都可以看成一个状态，每个状态都用一个不同的非终结符表示。

Coursework

3.4 给定文法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

请构造相应最小化的DFA。

Coursework

3.5 构造下列正规式相应的DFA

1(0｜1)*101

1(1010*｜1(010)*1)*0