编译原理

北方工业大学信息学院
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

3.3.1 正规式与正规集
3.3.2 确定有限自动机（DFA）
3.3.3 非确定有限自动机（NFA）
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

温故

3.2.4 状态转换图的实现

- 状态转换图的实现主要思想
  - 让每个状态结点对应一小段程序。
  - 对不含回路的分支结点，可以对应一个switch或一组if语句。
  - 对含回路的状态结点，可以对应一个while语句或if语句。
  - 终态结点对应一个return(code,value)语句。

8

3.2.4 状态转换图的实现

- **词法分析程序的设计与实现步骤**

词法规则 ➡ 状态转换图 ➡ 词法分析程序

1. 给出描述该语言各种单词符号的词法规则，以及输出形式；
2. 画出状态转换图；
3. 设计全局变量和过程，根据状态转换图构造词法分析器。

3.3.4 正规文法与有限自动机的等价性

3.3.1 正规式与正规集

- SQL语句

  select

  Select

  sElecT

  SELECT

  selECt

{S, E, L, C, T, s, e, l, c, t}

用一定的规则进行组合

3.3.1 正规式与正规集

- 正规式与正规集(Regular definition and regular expressions)

  字母表∑ 包含T、 **ε**和**Φ**

  定义在∑上的正规式和正规集的递归定义有三个步骤：

  ① 如果**ε**和**Φ**都是∑上的正规式，它们所表示的正规集分别为{ε}和Φ；

  ② 任何a∈∑，a是∑上的一个正规式，它所表示的正规集为{a}；

  ③ 假定**U**和**V**都是∑上的正规式，它们所表示的正规集分别记为**L(U)**和**L(V)**，
  那么，**(U|V)**、**(U•V)**和**(U)*** 也都是正规式

3.3.2确定有限自动机

- 确定有限自动机Deterministic finite automata (DFA)

① 含有m个状态和n个输入字符

② 图含有m个状态结点，从1个结点出发，**顶多有n条边**和别的结点相连接

有限自动机的所有边都是有向边

③ 每条边用∑中的1个**不同**输入字符作标记

④ 整张图含有**唯一的1个初态结点**

⑤ 有若干个(可以是0个)终态结点。

13

3.3.3 非确定有限自动机

- 非确定有限自动机Nondeterministic finite automata (NFA)
  - 该图含有m个状态结点
  - 每个结点有**若干条边**与别的结点相连接
  - 每条边用（∑ ∪ ε）中的一个字（**可以是相同的字**，而且**可以是空字ε**）作标记(称为输入字符)
  - 整张图**至少含有一个初态结点**
  - 有若干个(可以是0个)终态结点

14

DFA vs NFA

DFA M=（S, ∑, δ, s_0, F），其中δ是单值映射函数，s_0 是唯一初态。

同一个结点出来的箭弧上，不能有重复的字符，也不能有**ε**

NFA M=(S, ∑, δ, S_0, F)，其中δ 是多值映射函数， S_0 为非空初态集。

同一个结点出来的箭弧上，可以重复出现同样的字符，可以有**ε**

有限自动机通常表示为状态转换图，它是有限自动机的非形式化描述。

15

知新

3.3.3 非确定有限自动机

- **定义**：对于任何两个有限自动机M和M’，如果L(M)=L(M’)，则称M与M’等价。
- 自动机理论中一个重要的结论：判定两个自动机等价性的算法是存在的。
- 对于每个NFA M存在一个DFA M’，使得 L(M)=L(M’)。亦即DFA与NFA描述能力相同。

可以把NFA M转换为DFA M’

3.3.3 非确定有限自动机

- 证明对于任何两个有限自动机M和M’，如果L(M)=L(M’)，则称M与M’等价：

  假定NFA M=<S, Σ, δ, S_0, F>，我们对M的状态转换图进行以下改造：

  (1) 引进新的初态结点X和终态结点Y，X,Y∉S，

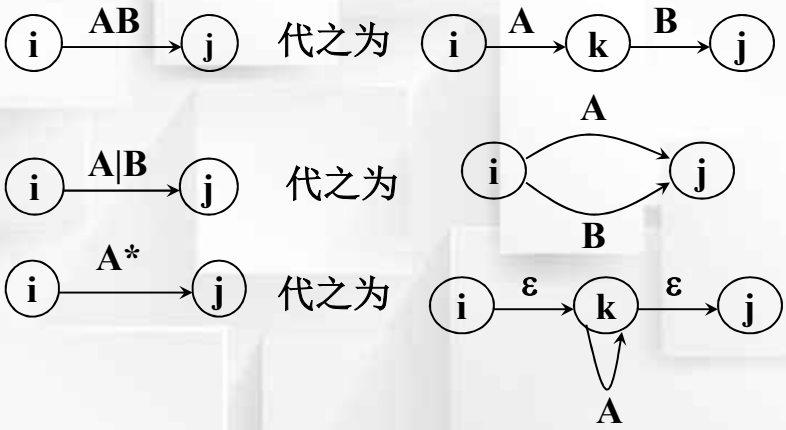
  从X到S_0中任意状态结点连一条ε边， 从F中任意状态结点连一条ε边到Y。

  (2) 对M的状态转换图进一步施行替换，其中k是新引入的状态。

  (3) 逐步把这个图转变为每条弧只标记为Σ上的一个字符或ε，最后得到一个NFA M’，显然L(M’)=L(M)

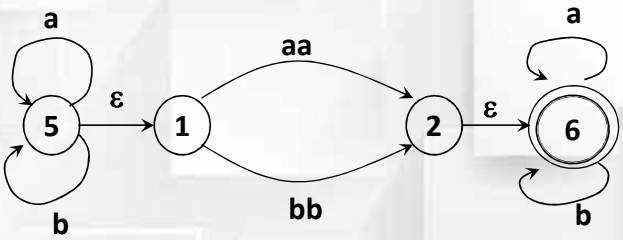
3.3.3 非确定有限自动机

- 改造使用的替换规则



3.3.3 非确定有限自动机

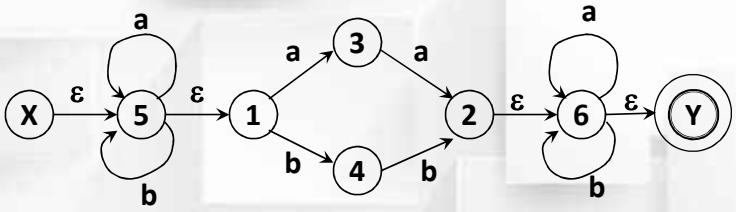
- 例题：识别所有含相继两个a或相继两个b的字符串，则NFA转换图如下



3.3.3 非确定有限自动机

替换后

NFA的状态转换图



21

3.3.3 非确定有限自动机

- 将NFA M确定化的方法，即变换成DFA M’，采用子集法：

  设I是NFA M’的状态集的一个子集，定义I的ε-闭包ε-closure(I)为:

  (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)。

22

3.3.3 非确定有限自动机

b) 设I是NFA M'的状态集的一个子集，a是Σ中的一个字符，定义

$$I_a = \varepsilon\text{-closure}(J)$$

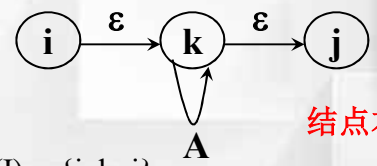
其中，J是那些可从I中的某一状态结点出发经过一条a弧而到达的状态结点的全体。

3.3.3 非确定有限自动机

I的ε-闭包ε-closure(I)：

ε是直通符

a) 找到一组NFA的状态结点，这些结点都能从NFA的某一个结点s ∈ I，或者多个类似结点s，通过ε边单独连接到。



结点本身要写进集合内

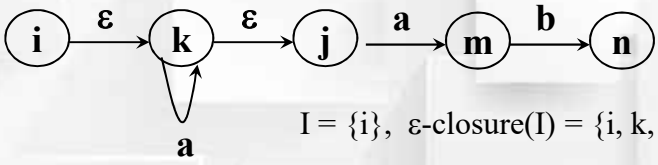
I = {i}
ε-closure(I) = {i, k, j}

I = {i, k}
ε-closure(I) = {i, k, j}

I = {k}
ε-closure(I) = {k, j}

3.3.3 非确定有限自动机

要找到I_a= ε-closure(J)，先得到ε-closure(I)，然后分两步：
第一步，符合条件**a** 可以达到的结点。找到所有可以通过a边连接到的结点集合J, 记为move(I, a)；**结点本身不写进集合内**
第二步，ε 是直通符。找到结点s' ∈ K通过ε边单独连接到的结点集合J'。



I = {i}，ε-closure(I) = {i, k, j} = I'

两步的结果：J= move(I', a) = {k, m}, J' = {j}

I_a= ε-closure(J) = J U J' = {k, j, m}

3.3.3 非确定有限自动机

c) Σ={a, b}，构造表(2+1)列，首行首列为ε-closure(x)

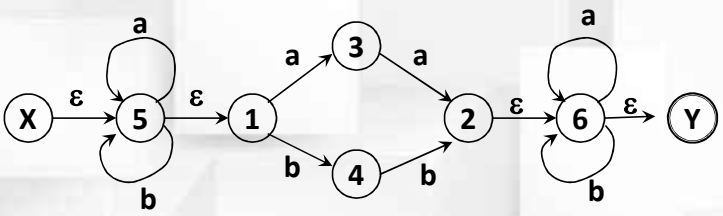
初始结点+直通到达结点 →

I	I_a	I_b
ε-closure(X)	符合条件a可达到的所有结点	符合条件b可达到的所有结点
I_a or I_b		

如果某一行的第一列的状态子集确定，则把该行的第二列置为I_a或I_b ...

然后，检查这两个I_a，I_b，看它们是否已在表中的第一列中出现，把未曾出现的填入后面的空行的第1列上，求出每行第2，3列上的集合...

3.3.3 非确定有限自动机

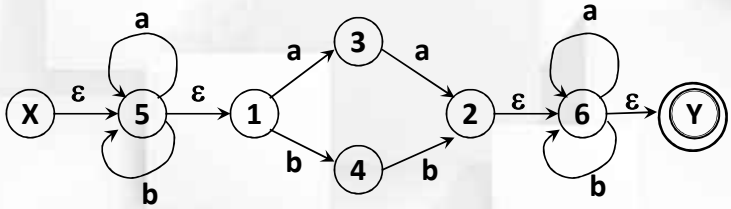


I	I_a	I_b
{X,5,1}	{5,3,1}	{5,4,1}
{5,3,1}	{?}	{?}

move(I, a)={5, 3}　I_a= ε-closure({5, 3})= {5, 1, 3}

move(I, b)={5, 4}　I_b= ε-closure({5, 4})= {5, 1, 4}

27



I	I_a	I_b
{X,5,1}	{5,3,1}	{5,4,1}
{5,3,1}	{5, 3,1,2,6,Y}	{5,4,1}
{5,4,1}	{5,3,1}	{5,4,1,2,6,Y}
{5,3,1,2,6,Y}	{5,3,1,2,6,Y}	{5,4,1,6,Y}
{5,4,1,6,Y}	{5,3,1,6,Y}	{5,4,1,2,6,Y}
{5,4,1,2,6,Y}	{5,3,1,6,Y}	{5,4,1,2,6,Y}
{5,3,1,6,Y}	{5,3,1,2,6,Y}	{5,4,1,6,Y}

28

3.3.3 非确定有限自动机

- 编号

I		I_a		I_b	
{X,5,1}	0	{5,3,1}	1	{5,4,1}	2
{5,3,1}	1	{5, 3,1,2,6,Y}	3	{5,4,1}	2
{5,4,1}	2	{5,3,1}	1	{5,4,1,2,6,Y}	5
{5,3,1,2,6,Y}	3	{5,3,1,2,6,Y}	3	{5,4,1,6,Y}	4
{5,4,1,6,Y}	4	{5,3,1,6,Y}	6	{5,4,1,2,6,Y}	5
{5,4,1,2,6,Y}	5	{5,3,1,6,Y}	6	{5,4,1,2,6,Y}	5
{5,3,1,6,Y}	6	{5,3,1,2,6,Y}	3	{5,4,1,6,Y}	4

29

3.3.3 非确定有限自动机

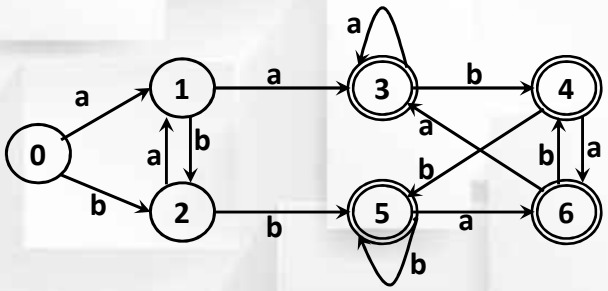
- 实现从NFA到DFA的转换
  - 现在把这张表看成一个状态转换矩阵，把其中的每个子集看成一个状态。
  - 这张表唯一刻划了一个确定的有限自动机M，它的初态是ε-closure({X})，它的终态是含有原终态Y的子集。
  - 这个DFA M与NFA M’等价。

30

3.3.3 非确定有限自动机

- 实现从NFA到DFA的转换

I	I_a	I_b
0	1	2
1	3	2
2	1	5
3	3	4
4	6	5
5	6	5
6	3	4



转换后的NFA

3.3.4 正规文法与有限自动机的等价性

3.3.4 正规文法与有限自动机的等价性

- 对于正规文法G和有限自动机M，如果L(G)＝L(M)，则称G和M是等价的。关于正规文法和有限自动机的等价性，有以下结论：
  - G的任何产生式为：
    - (1) A→αB 或A→α　　右线性正规文法
    - (2) A→Bα 或A→α　　左线性正规文法
    - 其中α∈V_T^*，A、B∈V_N。

3.3.4 正规文法与有限自动机的等价性

1.对每一个右线性正规文法G或左线性正规文法G，都存在一个有限自动机(FA) M，使得L(M)＝L(G)。

2.对每一个FA M，都存在一个右线性正规文法G_R和左线性正规文法G_L，使得L(M)＝L(G_R)＝L(G_L)。

3.3.4 正规文法与有限自动机的等价性

- 程序设计语言的单词符号可用乔母斯基3型文法——正规文法来描述。
- 对于正规文法所描述的语言可用一种有限自动机来识别。
- 下面给出从右线性正规文法构造相应有限自动机的方法。

35

3.3.4 正规文法与有限自动机的等价性

- 右线性正规文法构造相应有限自动机的方法

1.字母表与G的终结符号相同；

2.为G中的每个非终结符生成M的一个状态，G的开始符号S是有限自动机的开始状态S;

3.增加一个新状态Z，作为NFA的终态；

36

3.3.4 正规文法与有限自动机的等价性

- 右线性正规文法构造相应有限自动机的方法

4.对G中的形如A→tB，其中t为终结符或ε，A和B为非终结符的产生式，构造M的一个转换函数f(A,t)=B；

5.对G中的形如A→t的产生式，构造M的一个转换函数f(A,t)=Z。

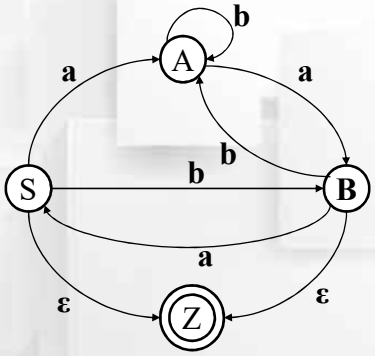
3.3.4 正规文法与有限自动机的等价性

- 右线性正规文法构造相应有限自动机的方法

例:求与文法G[S]等价的NFA

G[S]: S→aA|bB|ε
A→aB|bA
B→aS|bA|ε

求得：



第三章 小结

第三章 小结

3.3 正规式与有限自动机

3.3.3 非确定有限自动机（NFA）

3.3.4 正规文法与有限自动机的等价性

Coursework

3.2 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。