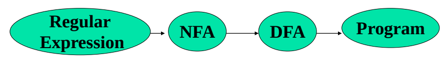
From Regular Expression To DFAs

1

Main Purpose

- Study an algorithm to:
  - Translate a regular expression into a DFA via NFA.



2

Contents

- From a Regular Expression to an NFA
  （中文课本3.6.3节）
- From an NFA to a DFA
  （中文课本3.6.1节）
- Minimizing the Number of States in a DFA
  （中文课本3.6.2节）

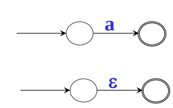
3

Regular Expression ⇒ NFA

(1)From regular expression to NFA

McMaughton-Yamada-Thompson algorithm

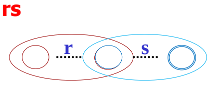
i. Construct the NFA of basic REs



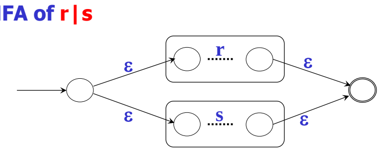
4

Regular Expression ⇒ NFA

ii. The NFA of rs



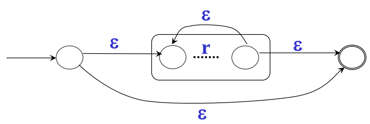
iii. The NFA of r|s



5

Regular Expression ⇒ NFA

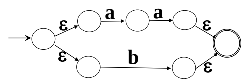
iv. The NFA of r*



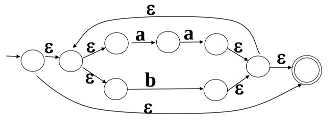
v. Connect together the NFAs of all subexpressions

6

Example： **(aa|b)*(a|bb)***

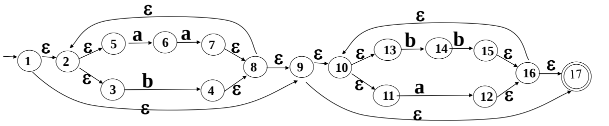


aa|b



(aa|b)*

7



(aa|b)*(a|bb)*

8

NFA ⇒ DFA

(2)NFA N ⇒ DFA D

DFA is a special **NFA**.

For any NFA N, we can find a DFA D with L(N)=L(D).

Get DFA by using subset construction

用子集构造法，构造DFA D的状态转换表Dtran

9

NFA ⇒ DFA

需要用到的几个定义：

- ε-closure(s)

The set of states reachable from state **s** on zero or more ε-transitions

- ε-closure(T)

状态集T的ε-闭包，可由0次或多次ε-转换从T中的状态到达的所有状态的集合

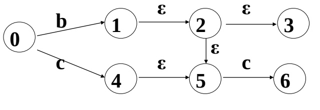
10

NFA ⇒ DFA

ε-closure(T)的计算：

- **If s∈T, then s∈ ε-closure(T)**；
- **If s∈T,** 则从**s**出发经任意条ε边能达到的所有状态都属于**ε-closure(T)**

Example：T={0,1,4}



ε-closure(T) ={0, 1, 4, 2, 3, 5}

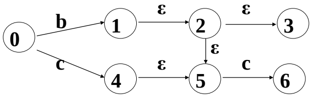
11

NFA ⇒ DFA

- **move[T, a]**的定义**(a∈Σ)**

是一个状态集，它是所有可以从**T**中的某一状态出发，经过一条**a**边（跳过**a**边前的任意条ε边）所能到达的状态组成的集合。

e.g. T={0,1,4}
move[T, c]={4, 6}



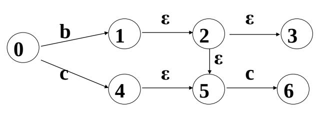
12

NFA ⇒ DFA

- **Dtran[T,a]的定义(a∈Σ)**

Dtran[T,a] = ε-closure(move[T, a])

e.g. T={0,1,4}
Dtran[T, c]
=ε-closure{4, 6}
={4, 6, 5}

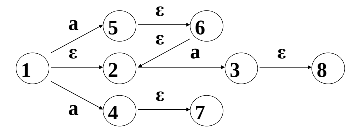


实际上： **$T \xrightarrow{a}$ Dtran[T,a]**

13

NFA ⇒ DFA

Example:



若T={1},
Dtran[T, a] = ε-closure(move[T, a])
= ε-closure({5,3,4})
={5,3,4,2,6,8,7}

14

NFA ⇒ DFA

Construct the DFA D

Step 1: Construct the transition table

i. 确定表头：表列数=Σ中字符数+1；

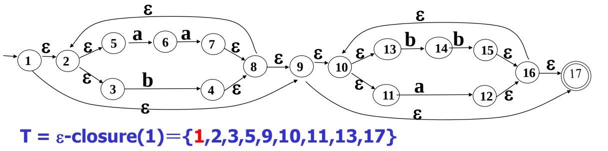
例：设**Σ={a,b}**，则表有**3**列，第**1**列为**T**（状态集），第**2**、**3**列分别为**Dtran[T,a]**、 **Dtran[T,b]**。

T	Dtran[T,a]	Dtran[T,b]
……	……	……

15

NFA M ⇒ DFA M

ⅱ.填表：第1行第1列元素为T=ε-closure(s_0)，然后求Dtran[T,a]和Dtran[T,b]；



T = ε-closure(1)={1,2,3,5,9,10,11,13,17}
Dtran[T,a]={6,12,16,17,10,11,13},
Dtran[T,b]= {4,14,8,9,10,11,13,17,2,3,5}

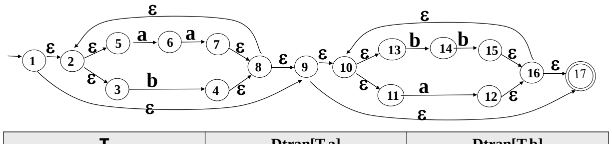
16

NFA M ⇒ DFA M

ⅲ. 填表：检查Dtran[T,a]和Dtran[T,b]是否在第1列出现过，把凡未在第1列出现过的Dtran[T,a]和Dtran[T,b]分别填在后面空行的第1列上，分别又求出相应的Dtran[T,a]和Dtran[T,b]。

直到所有Dtran[T,a]和Dtran[T,b]都在第1列上出现过为止。

17



T	Dtran[T,a]	Dtran[T,b]
{1,2,3,5,9,10,11,13,17}	{6,12,16,17,10,11,13}	{4,14,8,9,10,11,13,17,2,3,5}
{6,12,16,17,10,11,13}	{7,12,8,2,3,5,9,10,11,13,16,17}	{14}
{4,14,8,9,10,11,13,17,2,3,5}	{6,12,16,17,10,11,13}	{4,14,15,8,2,3,5,9,10,11,13,16,17}
{7,12,8,2,3,5,9,10,11,13,16,17}	{6,12,16,17,10,11,13}	{4,14,8,9,10,11,13,17,2,3,5}
{14}	Φ	{15,16,17,10,11,13}
{4,14,15,8,2,3,5,9,10,11,13,16,17}	{6,12,16,17,10,11,13}	{4,14,15,8,2,3,5,9,10,11,13,16,17}
{15,16,17,10,11,13}	{12,16,17,10,11,13}	{14}
{12,16,17,10,11,13}	{12,16,17,10,11,13}	{14}

18

NFA ⇒ DFA

Step 2: 把上述的转换表看做**DFA D**的状态转换表，并把每行每列的子集都作为状态重新命名（编号）：把**第1行第1列**的子集作为初态，以**1**代表，凡含有原终态的状态子集都作为终态，其余为非初非终态。

19

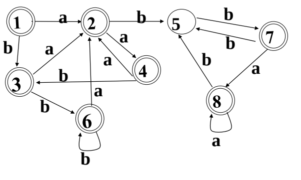
NFA ⇒ DFA

T	Dtran[T,a]	Dtran[T,b]
{1,2,3,5,9,10,11,13,17}1	{6,12,16,17,10,11,13}	{4,14,8,9,10,11,13,17,2,3,5}
{6,12,16,17,10,11,13} 2	{7,12,8,2,3,5,9,10,11,13,16,17}	{14}
{4,14,8,9,10,11,13,17,2,3,5}3	{6,12,16,17,10,11,13}	{4,14,15,8,2,3,5,9,10,11,13,16,17}
{7,12,8,2,3,5,9,10,11,13,16,17}4	{6,12,16,17,10,11,13}	{4,14,8,9,10,11,13,17,2,3,5}
{14}5	Φ	{15,16,17,10,11,13}
{4,14,15,8,2,3,5,9,10,11,13,16,17}6	{6,12,16,17,10,11,13}	{4,14,15,8,2,3,5,9,10,11,13,16,17}
{15,16,17,10,11,13}7	{12,16,17,10,11,13}	{14}
{12,16,17,10,11,13} 8	{12,16,17,10,11,13}	{14}

20

NFA ⇒ DFA

Step 3: 根据转换表画出状态转换图



21

Minimizing a DFA

(1)Concepts

- **Minimizing: Given any** DFA M, there is an **equivalent** DFA M' containing a minimum number of states, and this minimum-state DFA M' is unique.
- **等价状态**，对于一个DFA M，如果从状态s出发能够读出某字α，从状态t出发也能读出同样的字α，反之亦然，则称状态s和状态t等价。
- **可区别状态**，对于一个DFA M的两个状态t和s，如果不等价，则是可区别状态。

显然终态和非终态是可区别状态。

22

Minimizing a DFA

(2)化简过程，将状态集分割成一些不相交的子集，要求不同子集间的状态是**可区别的**，而子集内的状态是**等价的**。最后，每个子集用一个状态代表，得到最简状态。

23

Minimizing a DFA

对于那些在某个字符a上没有转换的状态，引入一个状态d，表示一个在a上的错误转换。

① 首先把**S**分成终态组**I**和非终态组**J**，此时**Π={I, J}**。然后拆分状态集。

② 若有**a**∈**Σ**，使得**move(I, a)**中的状态分布在**k**个不同的子集中，则将**I**分为**k**个组$I_1,I_2,\ldots,I_k$，划分的原则是：**move(I_j, a)**(j=1,…,k)中的状态分布在相同的子集中。

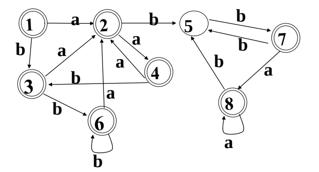
24

Minimizing a DFA

③对各个状态子集重复执行②，直到各个子集不可再分。

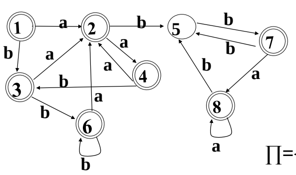
④ 各子集分别用一个状态$\mathbf{S}_i$代表，若子集中有原来的初态或终态，则$\mathbf{S}_i$为新的初态或终态。

25



首先把 **DFA M** 分成终态组 **I={1,2,3,4,6,7,8}** 和非终态组 **J={5}** 。然后拆分状态集。

26

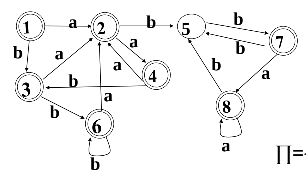


∏={I, J}={{1,2,3,4,6,7,8}, {5}}

由于move(I,a)={2,4,8}，move(I,b)={3,5,6,7}，

将I分为2组：{2,7,8}和{1,3,4,6} 。

27



∏={{1,3,4,6},{2,7,8},{5}}

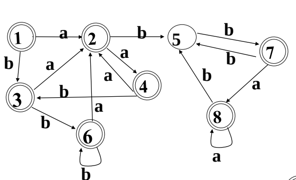
由于move(**{1,3,4,6}**,a)={2}，move(**{1,3,4,6}**,b)={3,6}，

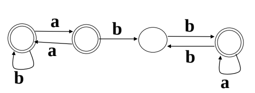
{1,3,4,6}不可分。

由于move(**{2,7,8}**,a)={4,8}，

将{2,7,8}分为{2}、{7,8}。

28





∏={{1,3,4,6},{2},{7,8},{5}}

29