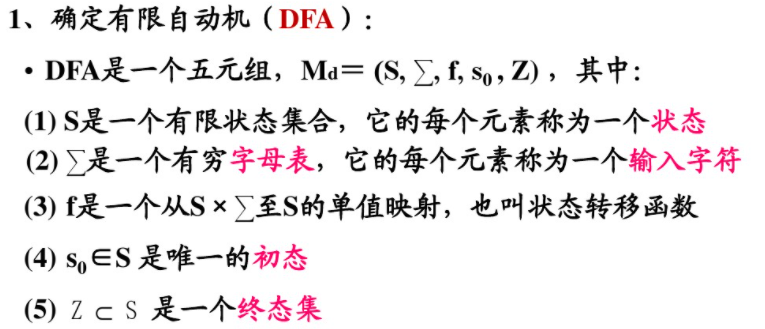
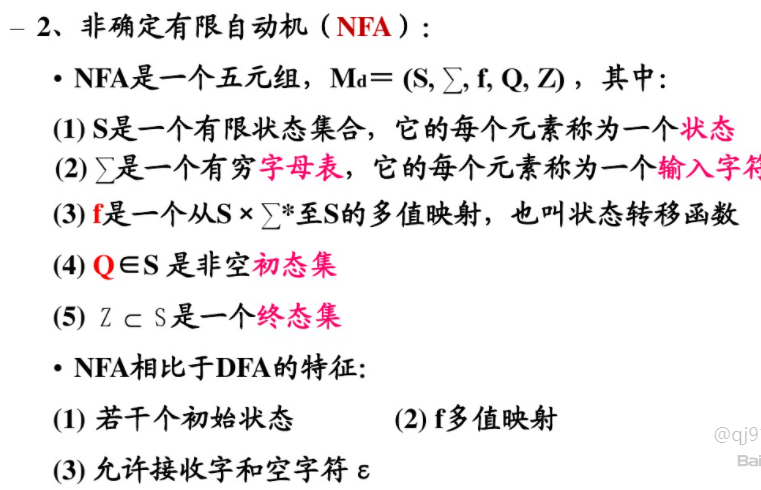
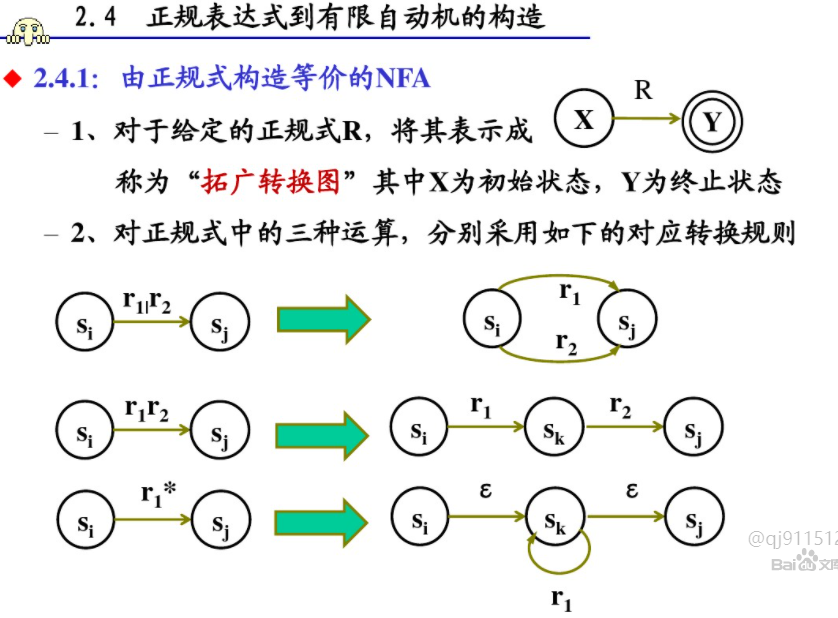
# 定义

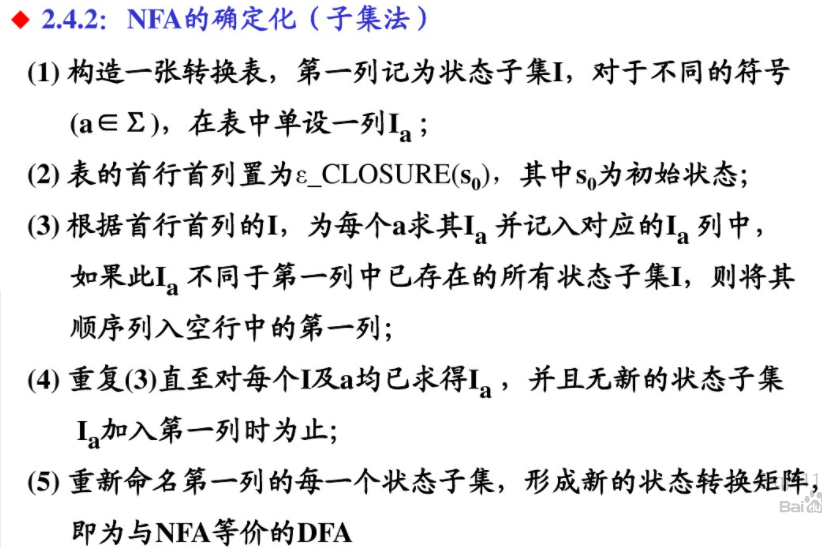




# CreateNFA



# Nfa2dfa



Subset Construction

初始Dstates，它仅仅含有状态(D的状态)ε-closure(s0)，并且状态未被标记，s0表示开始状态，注意，Dstates放的是D的状态  
while ( Dstates 有未标记的状态 T ) {

// T是D中的一个状态，也是N中一个状态集  
标记 T;  
for ( input symbol a ){

// 遍历所有的input symbol

// move为NFA的move函数  
  U = ε-closure(move(T, a));          
  if ( U 不在 Dstates 中 )  
      把U作为尚未标记的状态加入Dstates;  
       Dtran[T, a] = U  
    }  
}  
  
注意，状态s，ε-closure(s)一定包含s

把T的所有状态压入stack(栈);  
ε-closure(T)的初始值为 T 中的所有元素 ;  // 也就是一定包含他们本身  
while( 栈非空 ) {  
    弹出栈顶元素 t ;  
    for( 每个属于 move(t, ε) 的状态 u ){  
       if( u 不在 ε-closure(T) 中 ){  
          u 加入 ε-closure(T);  
          把 u 入栈;  
       }  
    }  
}

# Dfa化简

4.DFA最小化

## 4.1 最小状态DFA的含义

1.没有多余状态(死状态)

2. 没有两个状态是互相等价（不可区别）

## 什么是多余状态？

从这个状态没有通路到达终态；S1

从开始状态出发，任何输入串也不能到达的那个状态。S2

如何消除多余状态？

删除

--------------------------------------------------------

作者：zhezhelin是我

链接：https://wenku.baidu.com/view/ba58db6003d8ce2f01662336.html

来源：百度文库

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

## 4.2 两个状态s和t等价的条件

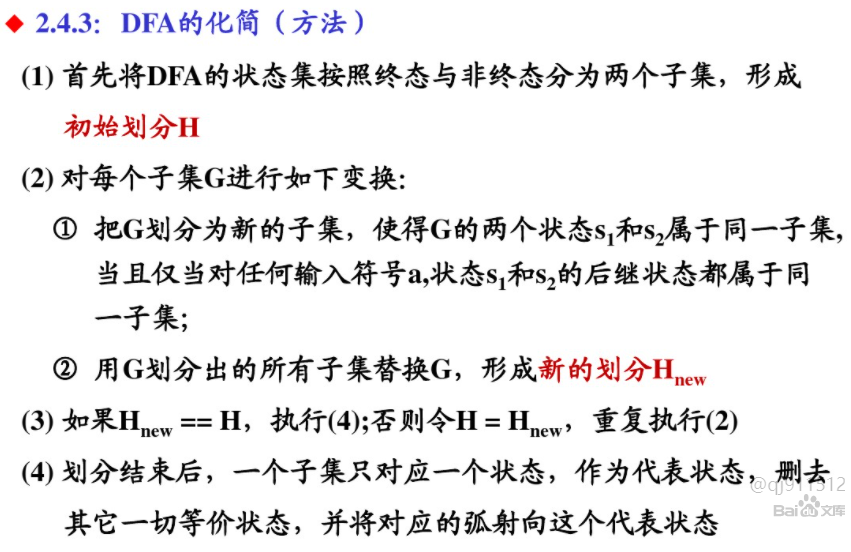
兼容性（一致性）条件——同是终态或同是非终态

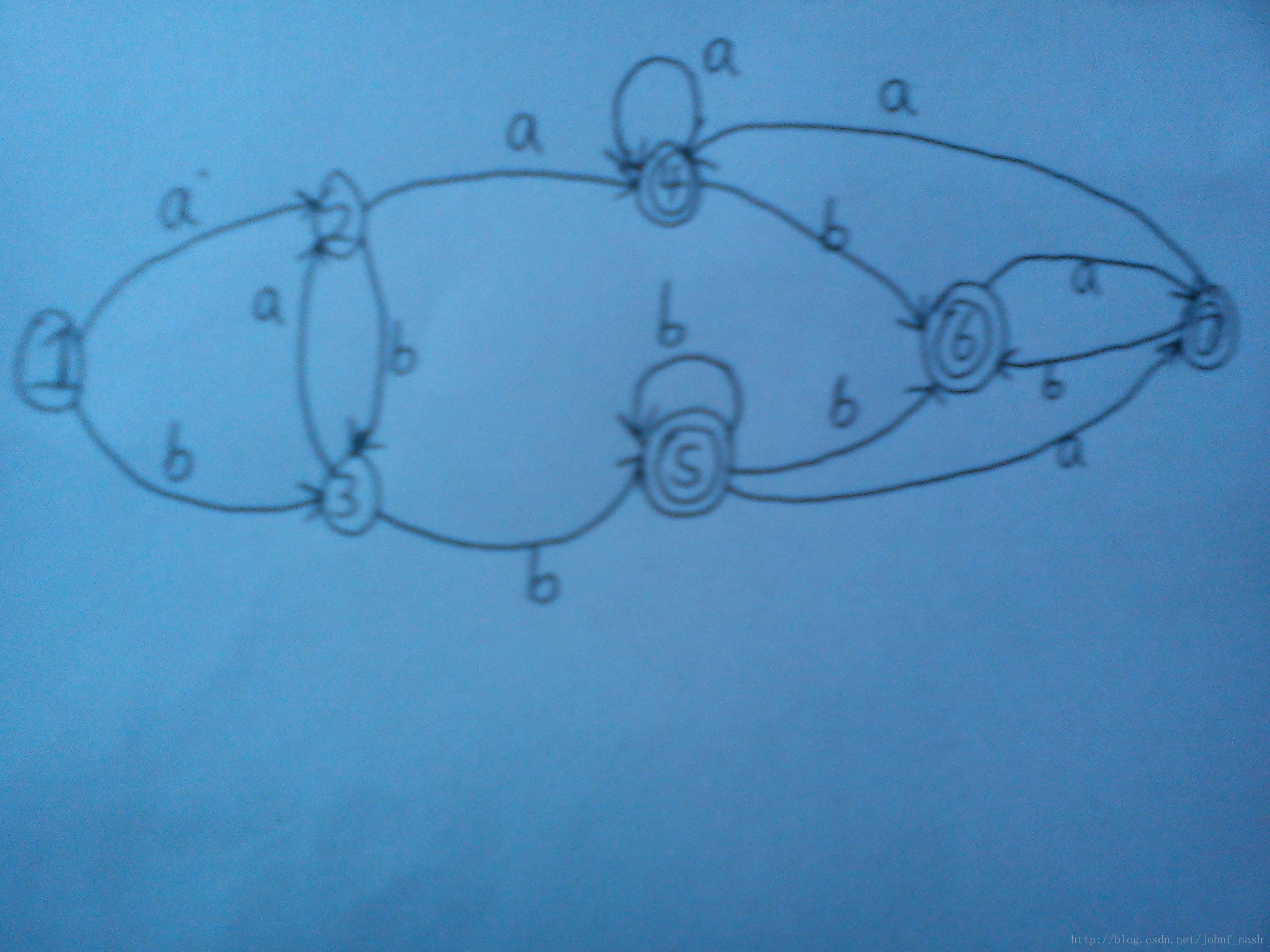
传播性（蔓延性）条件——从s出发读入某个a和从t出发经过某个a并且经过某个b到达的状态等价。

————————————————

版权声明：本文为CSDN博主「xxc1605629895」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/johnf_nash/article/details/21988565>





DFA的最小化

1. 将状态列表S中的状态分为两个子集:终结状态s1(也叫可接受状态)和非终结状态s2

对于上图中的DFA, s1 = {4, 5, 6, 7}, s2 = {1, 2, 3}

2. 查看s2 = {1, 2, 3}是否可以继续划分

状态2遇到a转向s1中的状态4,不在s2中，于是将s2继续划分，分为{1, 3}和{2}这两个集合。

继续查看{1, 3}是否可以继续划分。由于3遇到b转向s1中的状态5,于是将{1, 3}继续划分，分为{1}和{3}这两个集合。

3. 查看 s1= {4, 5, 6, 7}是否可以继续划分

因为4, 5, 6, 7经过a和b跳转到的状态都在s1中，不可再分。因而这四个节点可以用一个节点来表示。

需要注意的是，当最终所以的子集不可再分时，如果一个子集内的状态不止一个，则由于同一子集内的状态等价，

同一子集内的节点之间的状态跳转可以不用考虑。但是如果这个子集内的某一状态遇到一个符号转向本身时，

这种向自身的转移要体现在新的替换的状态上。例如4,5,6,7最终属于同一子集，最终用节点4来替换这4个节点时，

需要在状态转移矩阵中加上遇到a跳转到节点4，与遇到b跳转到节点4的情况。

————————————————

版权声明：本文为CSDN博主「xxc1605629895」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/johnf\_nash/article/details/21988565