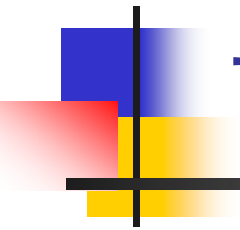


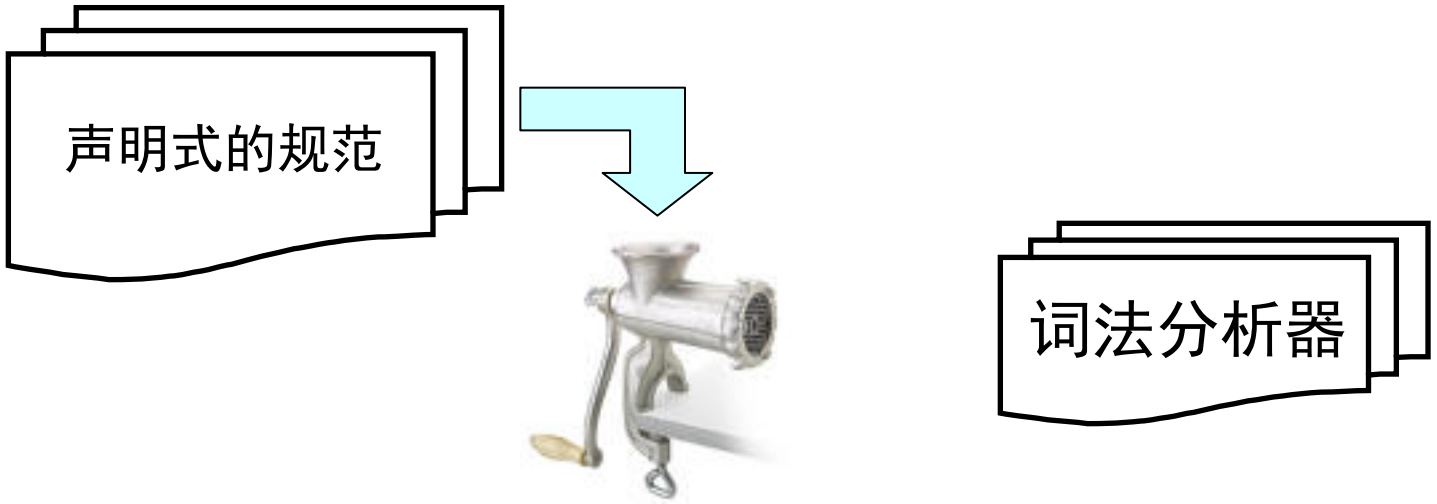
词法分析---DFA的代码表示

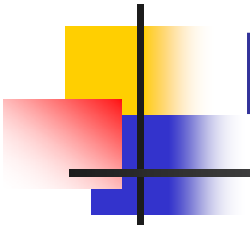
编译原理

华保健

bjhua@ustc.edu.cn

回顾：自动生成

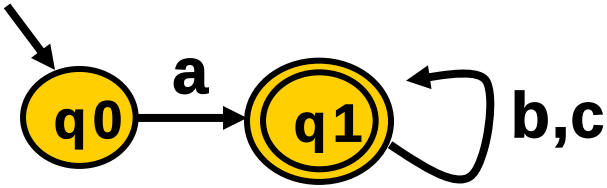

声明式的规范
词法分析器
Thompson算法
子集构造算法
Hopcroft
最小化算法
RE
NFA
DFA
词法分析器代码

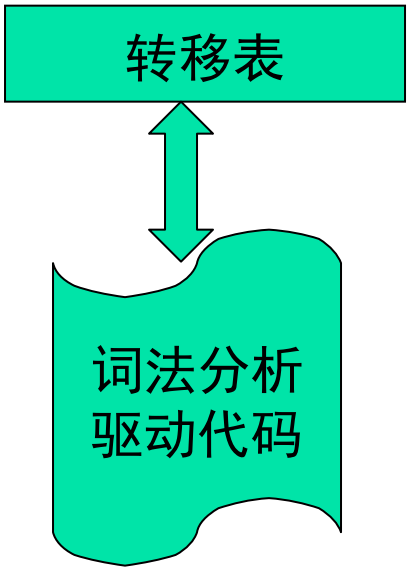


DFA的代码表示

- 概念上讲, DFA是一个有向图
- 实际上，有不同的DFA的代码表示
  - 转移表 (类似于邻接矩阵)
  - 哈希表
  - 跳转表
  - 。 。 。
- 取决于在实际实现中，对时间空间的权衡

转移表

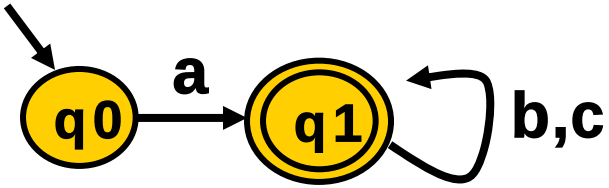




状态\字符	a	b	c
0	1		
1		1	1

```
char table[M][N];

table[0]['a']=1;
table[1]['b']=1;
table[1]['c']=1;
// other table entries
// are ERROR
```

驱动代码

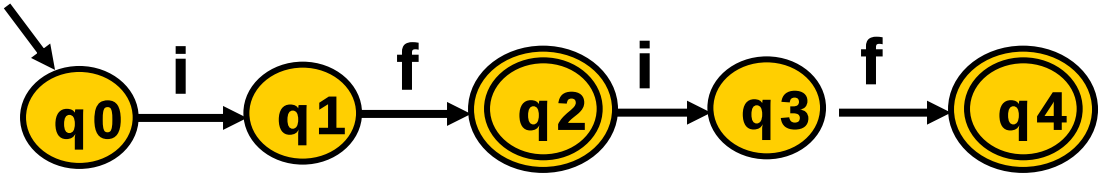



```
nextToken()
  state = 0
  stack = []
  while (state!=ERROR)
    c = getChar()
    if (state is ACCEPT)
      clear(stack)
    push(state)
    state = table[state][c]

  while(state is not ACCEPT)
    state = pop();
    rollback();
```

状态\字符	a	b	c
0	1		
1		1	1

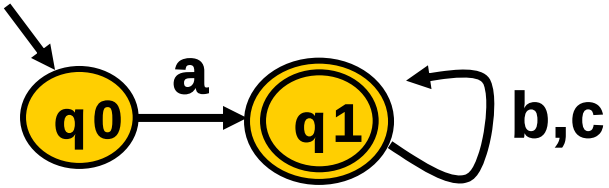
```
char table[M][N];

table[0]['a']=1;
table[1]['b']=1;
table[1]['c']=1;
// other table entries
// are ERROR
```

最长匹配




```
nextToken()
  state = 0
  stack = []
  while (state!=ERROR)
    c = getChar()
    if (state is ACCEPT)
      clear(stack)
    push(state)
    state = table[state][c]

  while(state is not ACCEPT)
    state = pop();
    rollback();
```

跳转表



状态\字符	a	b	c
0	1		
1		1	1

```
nextToken()
  state = 0
  stack = []
  goto q0

q0:
  c = getChar()
  if (state is ACCEPT)
    clear (stack)
  push (state)
  if (c==‘a’)
    goto q1:
```

```
q1:
  c = getChar()
  if (state is ACCEPT)
    clear (stack)
  push (state)
  if (c==‘b’||c==‘c’)
    goto q1
```