



# 词法分析---NFA转换到DFA

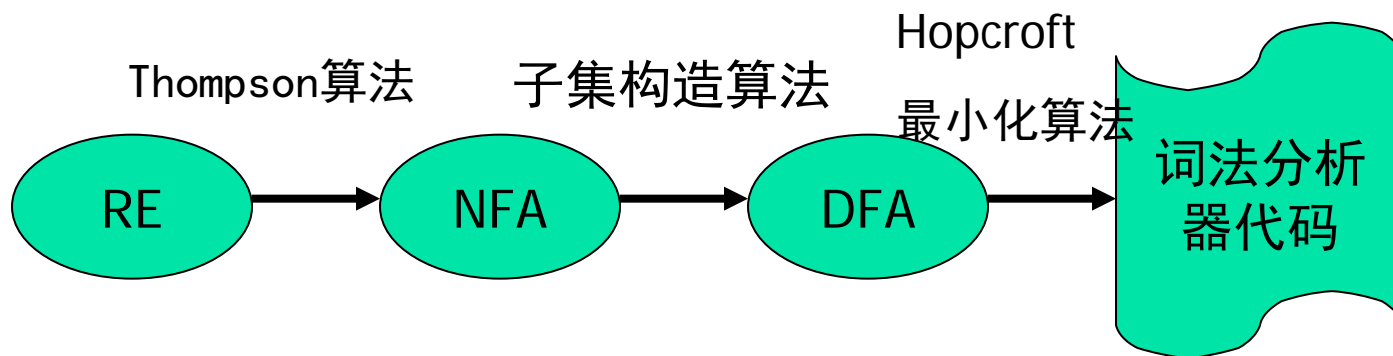
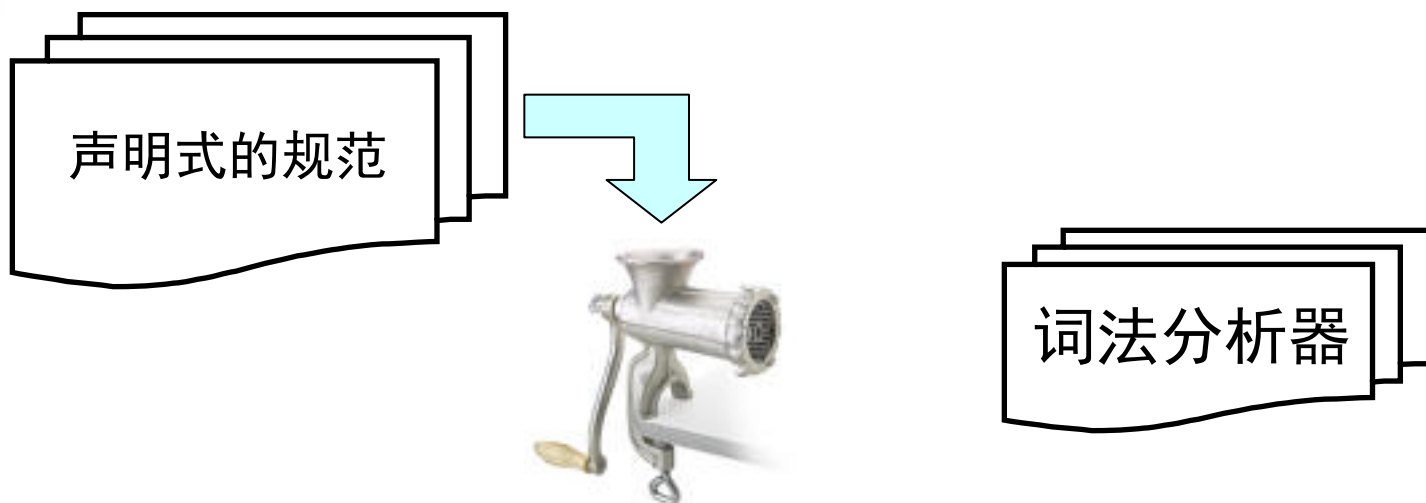
---

编译原理

华保健

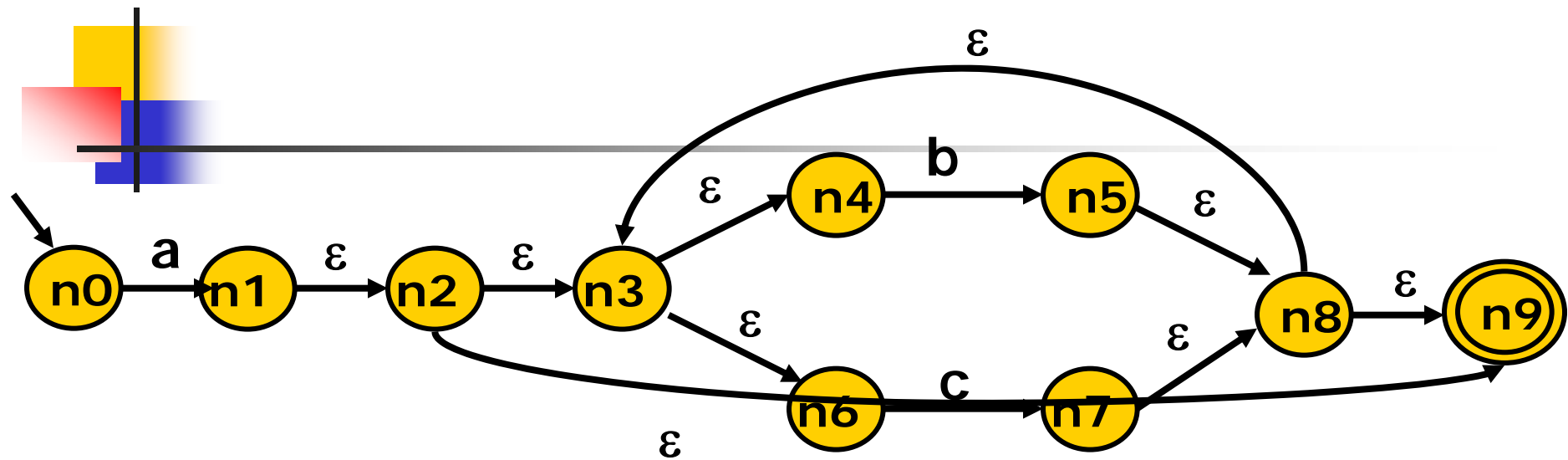
[bjhua@ustc.edu.cn](mailto:bjhua@ustc.edu.cn)

# 回顾：自动生成



# 算法思想

$a(b|c)^*$





# 子集构造算法

---

(\* 子集构造算法：工作表算法 \*)

```
q0 <- eps_closure (n0)
```

```
Q <- {q0}
```

```
workList <- q0
```

```
while (workList != [])
```

```
  remove q from workList
```

```
  foreach (character c)
```

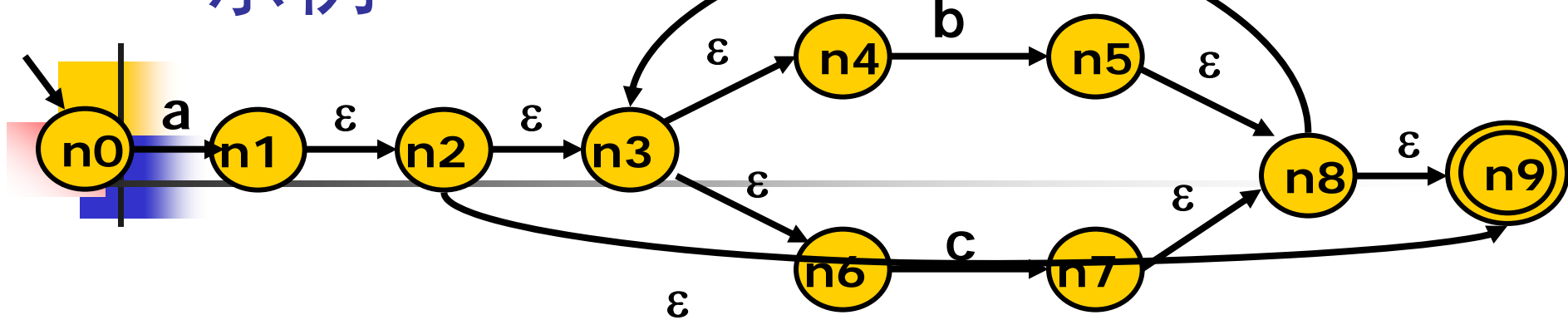
```
    t <- e-closure (delta (q, c))
```

```
    D[q, c] <- t
```

```
    if (t \not\in Q)
```

```
      add t to Q and workList
```

# 示例



(\* 子集构造算法：工作表算法 \*)

```
q0 <- eps_closure (n0)
```

```
Q <- {q0}
```

```
workList <- q0
```

```
while (workList != [])
```

```
  remove q from workList
```

```
  foreach (character c)
```

```
    t <- e-closure (delta (q, c))
```

```
    D[q, c] <- t
```

```
    if (t \not\in Q)
```

```
      add t to Q and workList
```



# 对算法的讨论

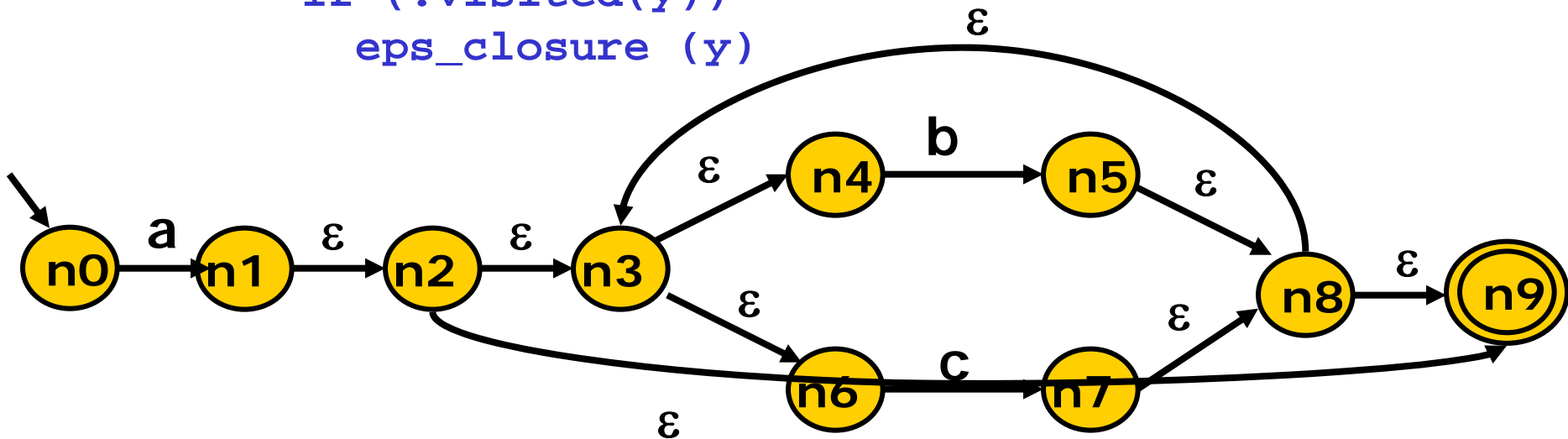
---

- 不动点算法
  - 算法为什么能够运行终止
- 时间复杂度
  - 最坏情况 $O(2^N)$
  - 但在实际中不常发生
    - 因为并不是每个子集都会出现

# $\epsilon$ -闭包的计算：深度优先

```
/*  $\epsilon$ -closure: 基于深度优先遍历的算法 */  
set closure = {};
```

```
void eps_closure (x)  
    closure += {x}  
    foreach (y: x-- $\epsilon$ --> y)  
        if (!visited(y))  
            eps_closure (y)
```





# $\varepsilon$ -闭包的计算：宽度优先

---

```
/*  $\varepsilon$ -closure: 基于宽度优先的算法 */  
set closure = {};  
Q = []; // queue  
void eps_closure (x) =  
    Q = [x];  
    while (Q not empty)  
        q <- deQueue (Q)  
        closure += q  
        foreach (y: q-- $\varepsilon$ --> y)  
            if (!visited(y))  
                enqueue (Q, y)
```