

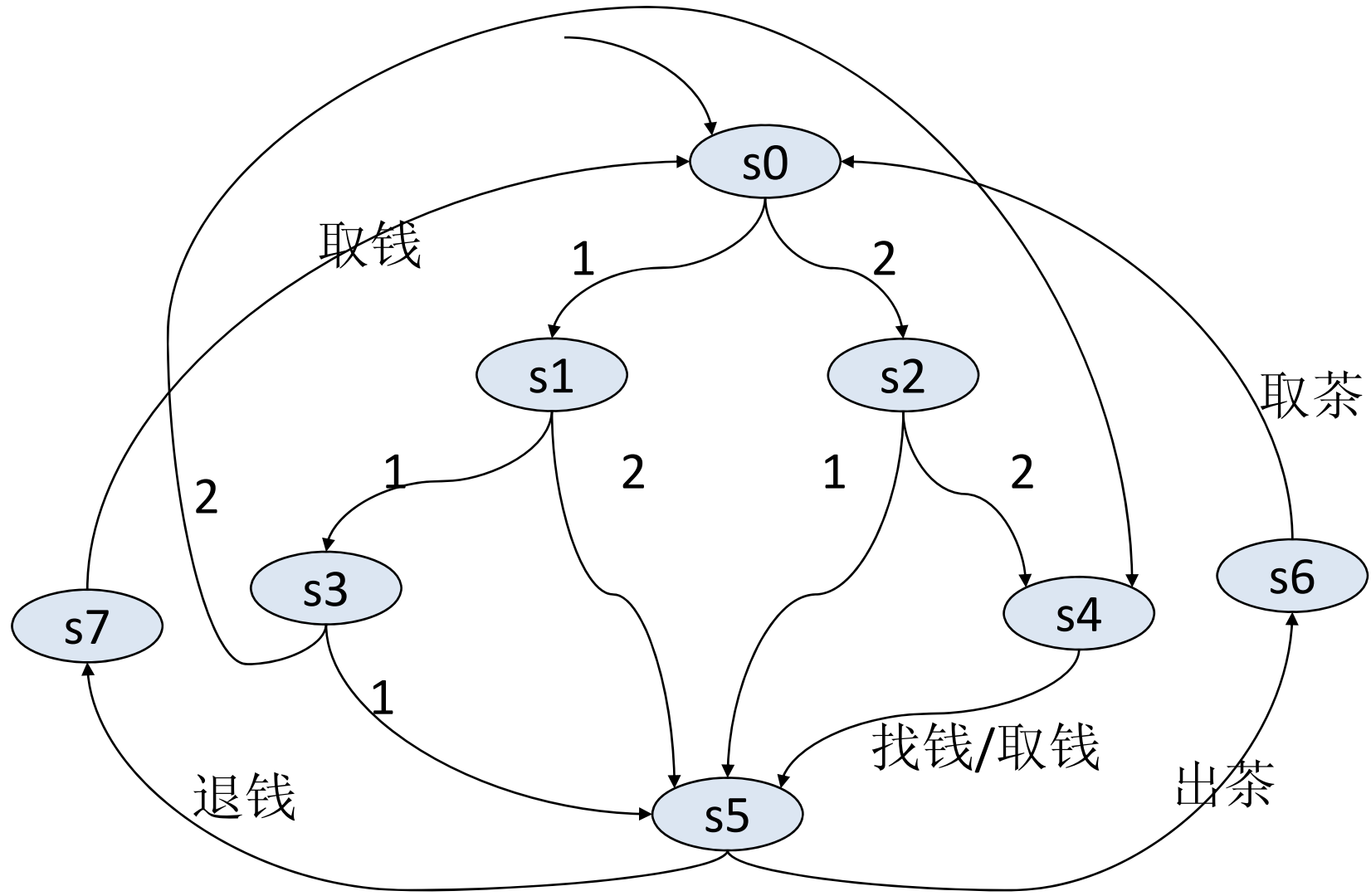
# 标号迁移系统

中国科学院软件研究所

张文辉

<http://lcs.ios.ac.cn/~zwh/pv>

# 自动售茶机



# 互斥协议：卫式迁移模型

- 迁移集合：

$$a = s_0 \quad \longrightarrow \quad (y, t, a) := (1, 1, s_1)$$

$$a = s_1 \wedge (x = 0 \vee t = 0) \quad \longrightarrow \quad (a) := (s_2)$$

$$a = s_2 \quad \longrightarrow \quad (y, a) := (0, s_3)$$

$$a = s_3 \quad \longrightarrow \quad (y, t, a) := (1, 1, s_1)$$

$$b = t_0 \quad \longrightarrow \quad (x, t, b) := (1, 0, t_1)$$

$$b = t_1 \wedge (y = 0 \vee t = 1) \quad \longrightarrow \quad (b) := (t_2)$$

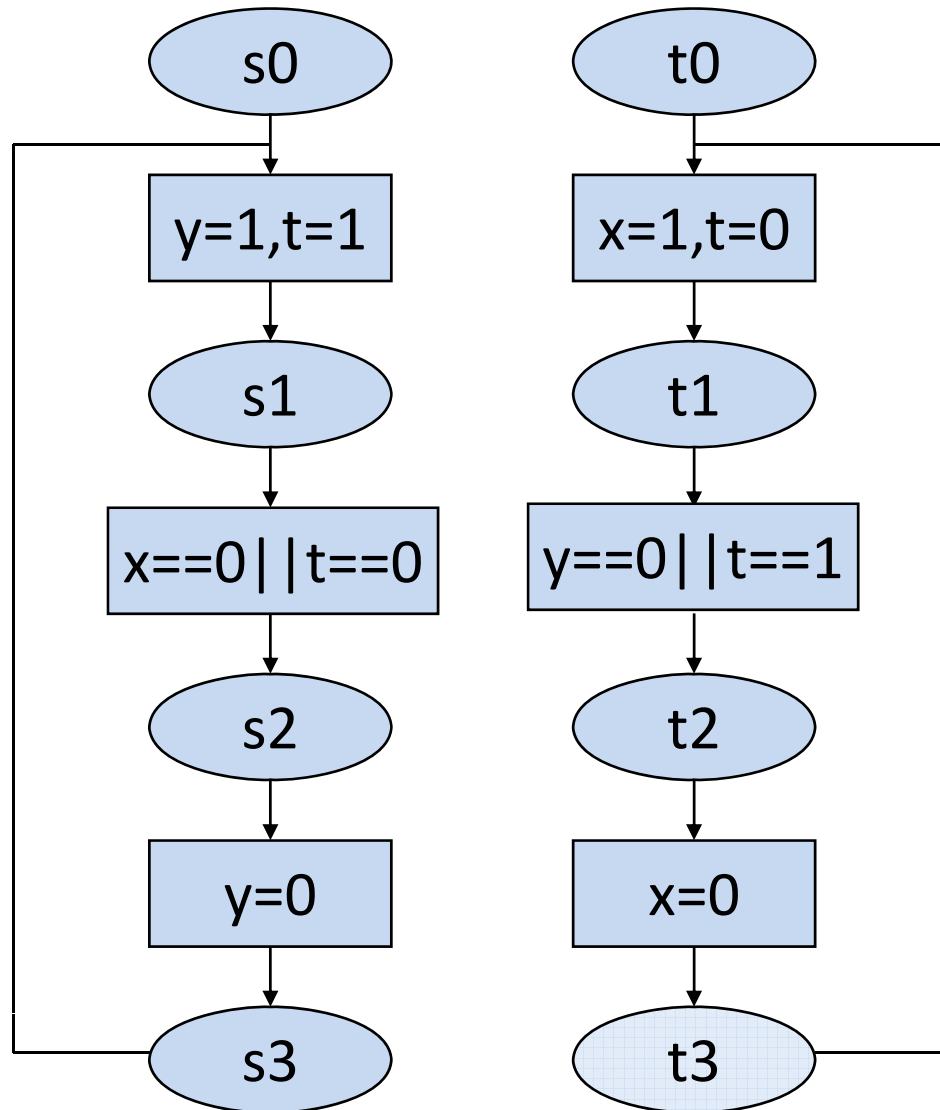
$$b = t_2 \quad \longrightarrow \quad (x, b) := (0, t_3)$$

$$b = t_3 \quad \longrightarrow \quad (x, t, b) := (1, 0, t_1)$$

- 初始状态：

$$(a = s_0 \wedge b = t_0 \wedge x = 0 \wedge y = 0 \wedge t = 0)$$

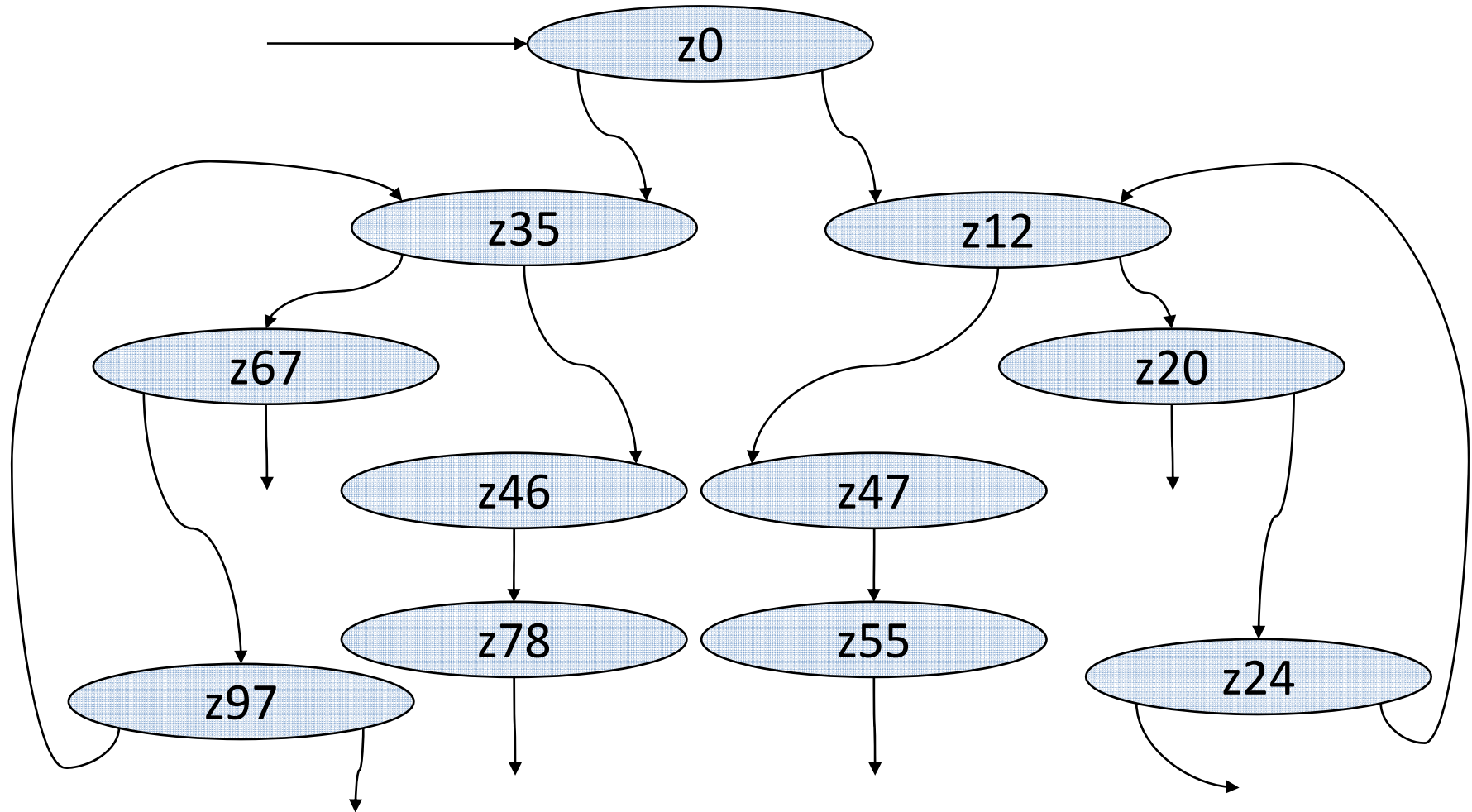
# 互斥协议：示意图



初始状态

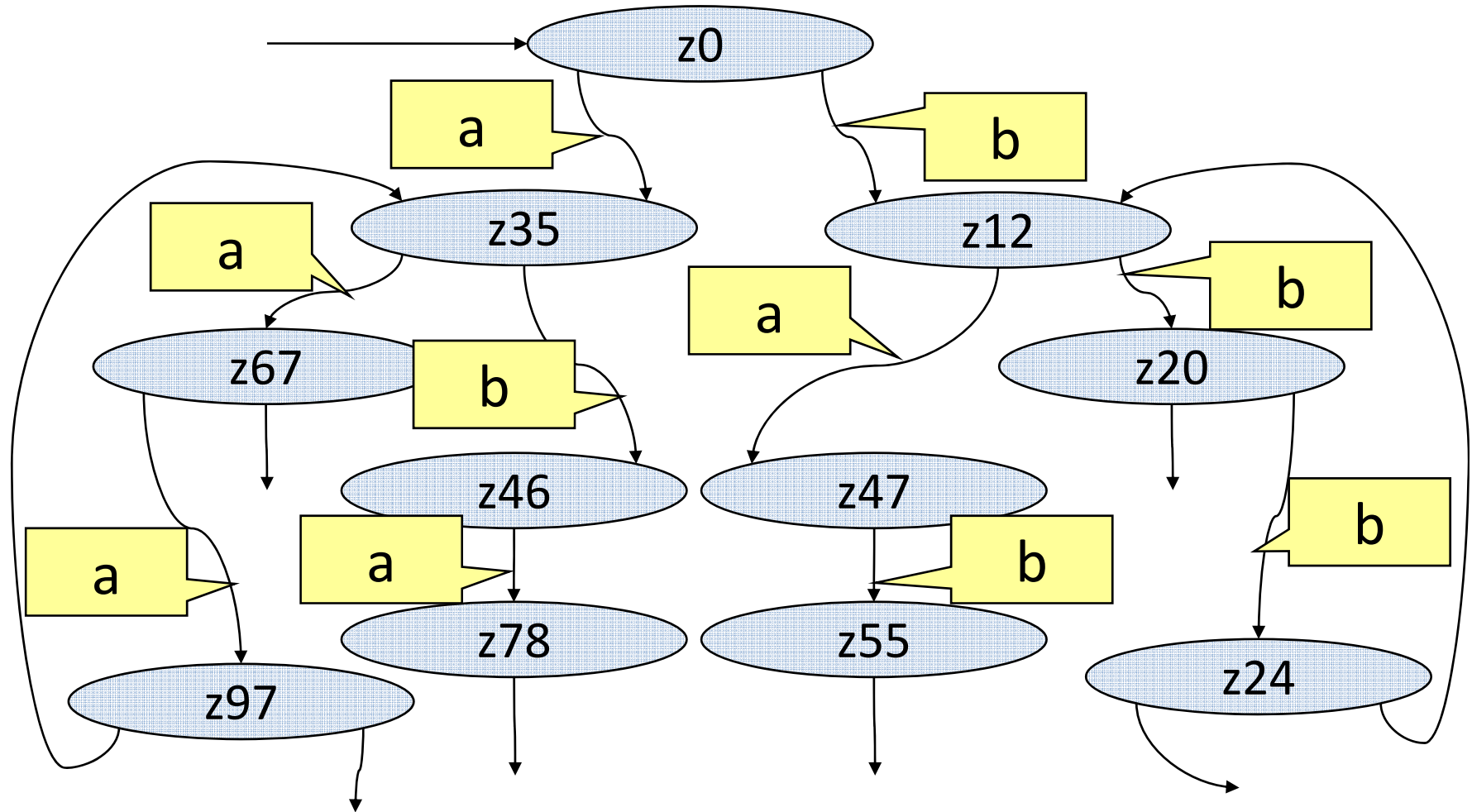
$s_0$   
 $t_0$   
 $x=0$   
 $y=0$   
 $t=0$

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

# 标号迁移系统

# 标号迁移系统

- 动作信息                      符号
- 系统状态                      抽象状态
- 状态变化                      三元组
- 初始状态                      状态集合



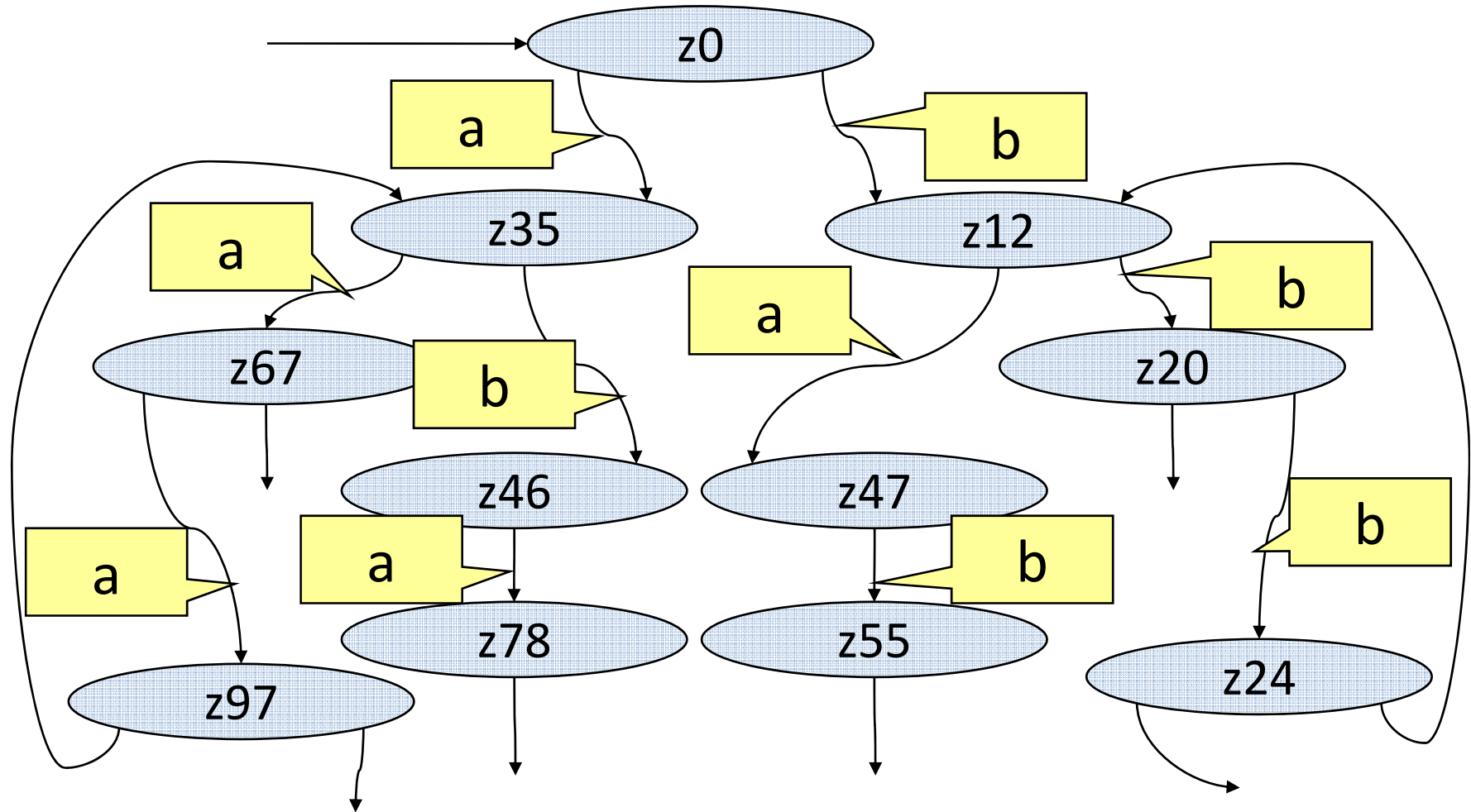
标号迁移系统

# 标号迁移系统：例子

- 标号集合：  $\{ a, b \}$
- 状态集合：  $\{ z_0, z_1, z_2, z_3, \dots \}$
- 迁移关系：  $\{ (z_0, a, z_{35}), (z_0, b, z_{12}), \dots \}$
- 初始状态集：  $\{ z_0 \}$

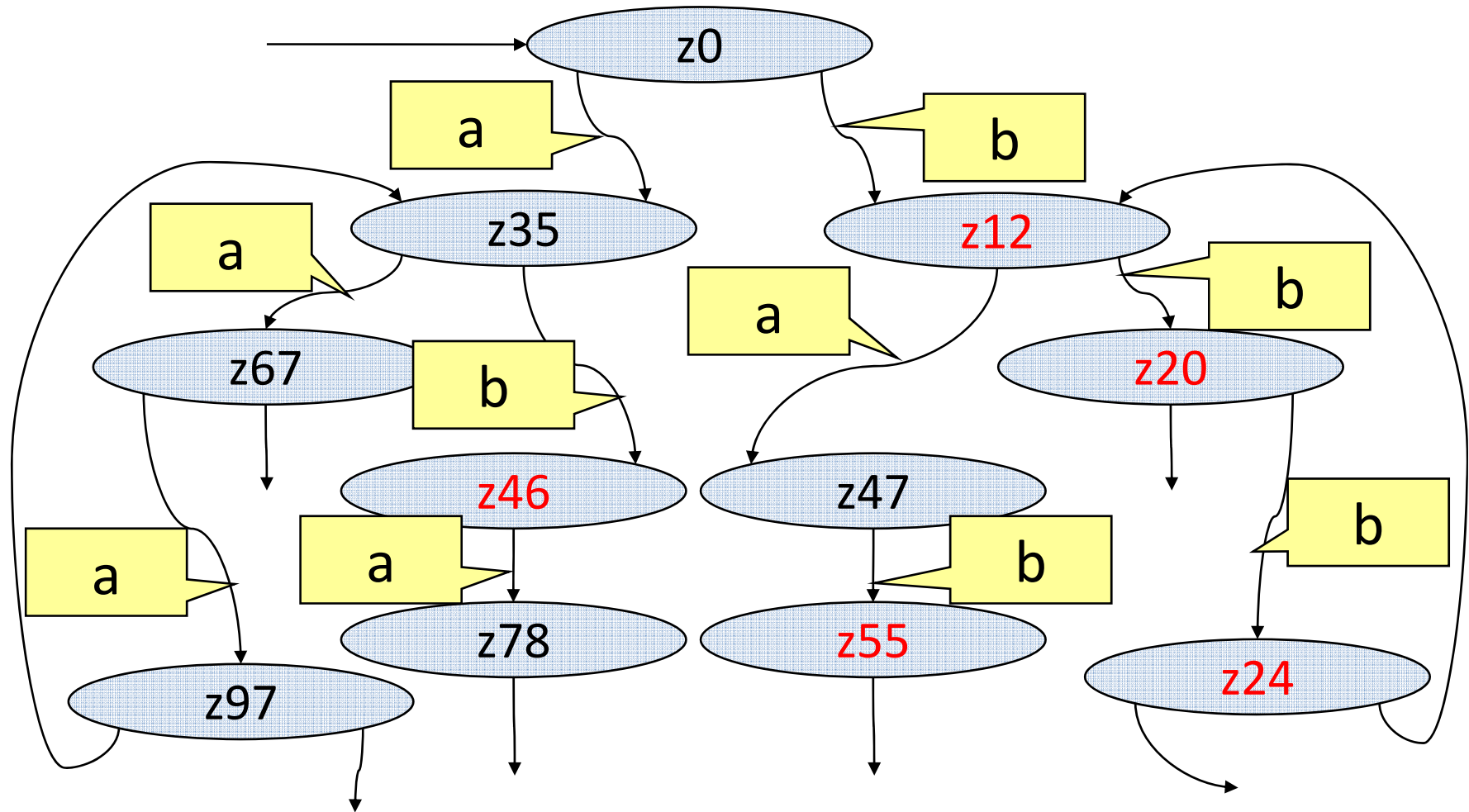
# Büchi 自动机

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

# Büchi 自动机

- 动作信息
- 系统状态
- 状态变化
- 初始状态
- 公平性约束

符号

抽象状态

三元组

状态集合

状态集合



Büchi 自动机

# Büchi 自动机：例子

- 标号集合：  $\{ a, b \}$
- 状态集合：  $\{ z_0, z_1, z_2, z_3, \dots \}$
- 迁移关系：  $\{ (z_0, a, z_{35}), (z_0, b, z_{12}), \dots \}$
- 初始状态集：  $\{ z_0 \}$
- 接受状态集：  $\{ z_{12}, z_{20}, z_{46}, \dots \}$

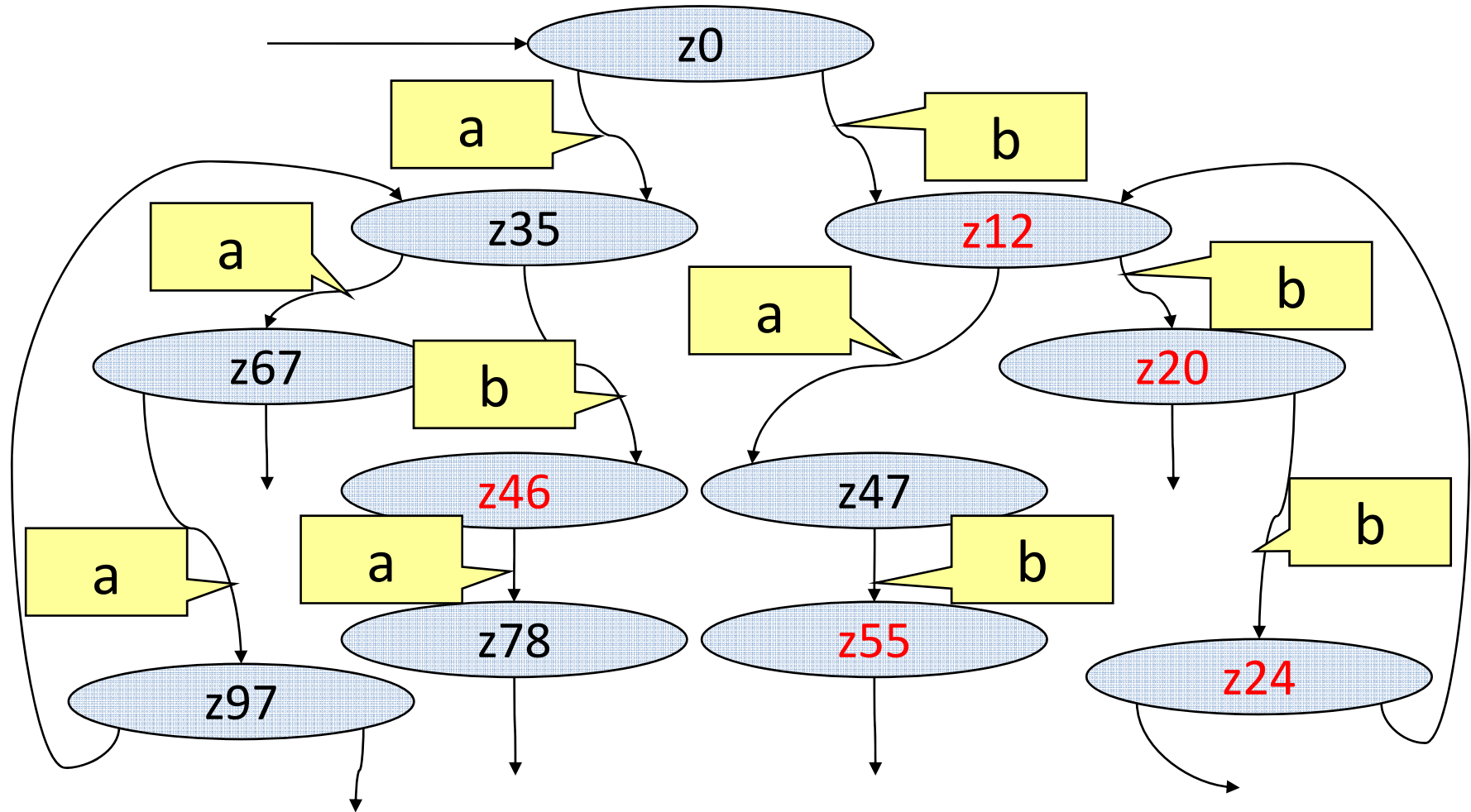
# Büchi自动机：运行/语言

- z0 z35 z67 z97 ...
- z0 z35 z46 z78 ...
- a a a ...
- a b b ...

语言：  $(a|b)^\omega$  的子集

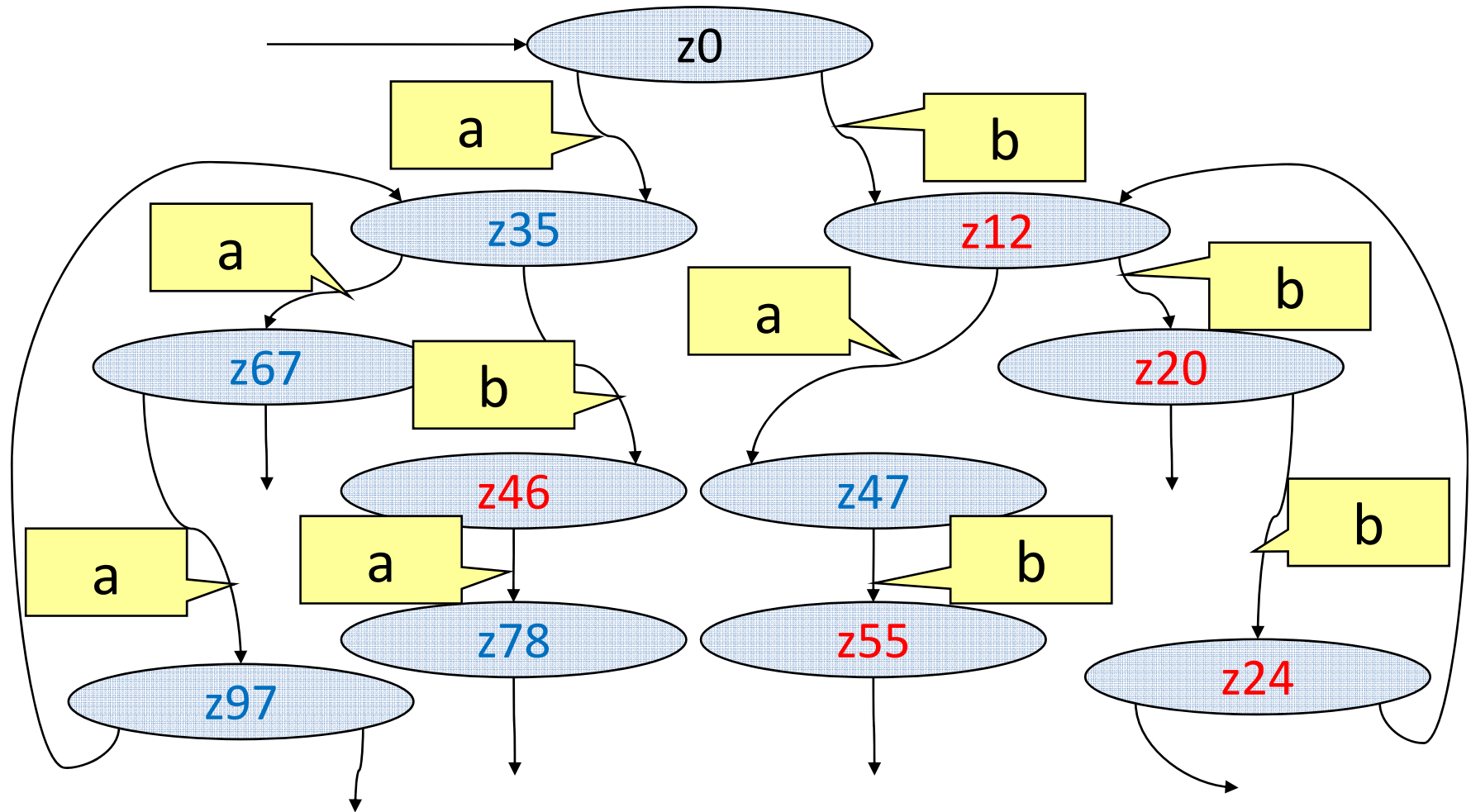
# 扩展Büchi自动机

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

# 扩展Büchi自动机

- 动作信息                      符号
- 系统状态                      抽象状态
- 状态变化                      三元组
- 初始状态                      状态集合
- 多元公平性                    状态集合的集合



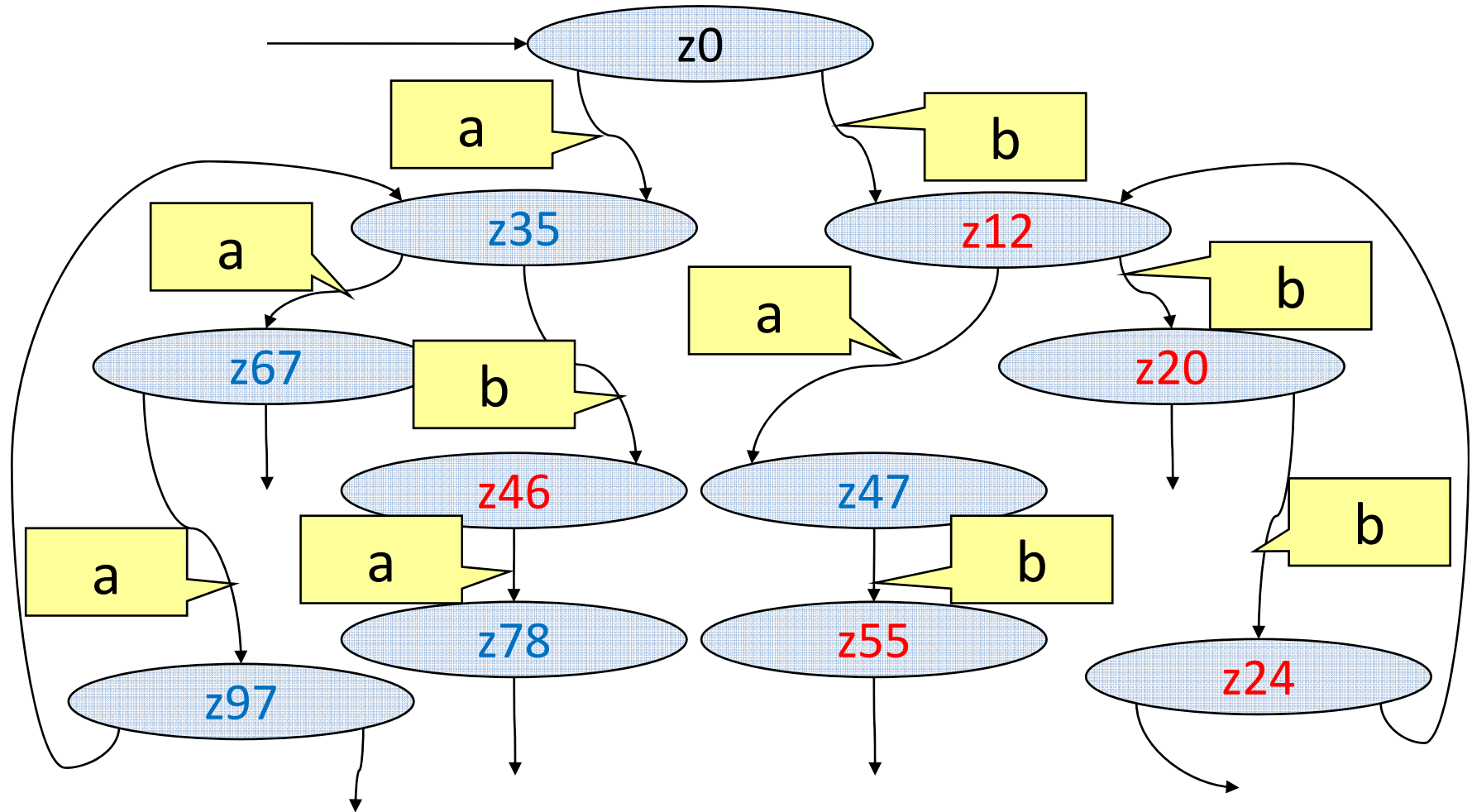
扩展Büchi自动机

# 扩展Büchi自动机：例子

- 标号集合： $\{ a, b \}$
- 状态集合： $\{ z_0, z_1, z_2, z_3, \dots \}$
- 迁移关系： $\{ (z_0, a, z_{35}), (z_0, b, z_{12}), \dots \}$
- 初始状态集： $\{ z_0 \}$
- 接受状态集集合： $\{ \{z_{12}, z_{20}, \dots\}, \{z_{35}, z_{67}, \dots\} \}$

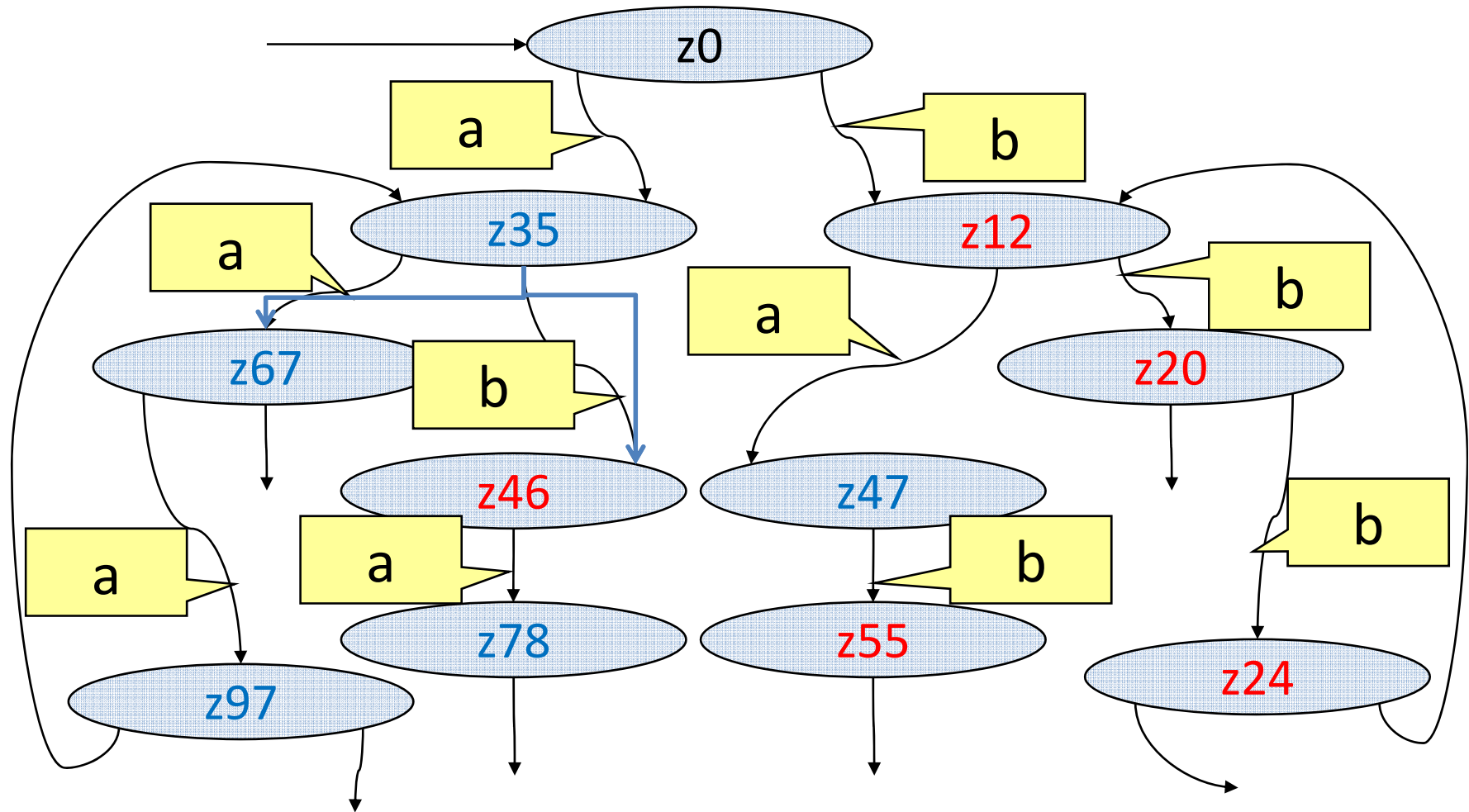
# Streett 自动机

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

# Streett 自动机

- 动作信息
  - 系统状态
  - 状态变化
  - 初始状态
  - 强公平性
- 符号  
抽象状态  
三元组  
状态集合  
状态集合对的集合



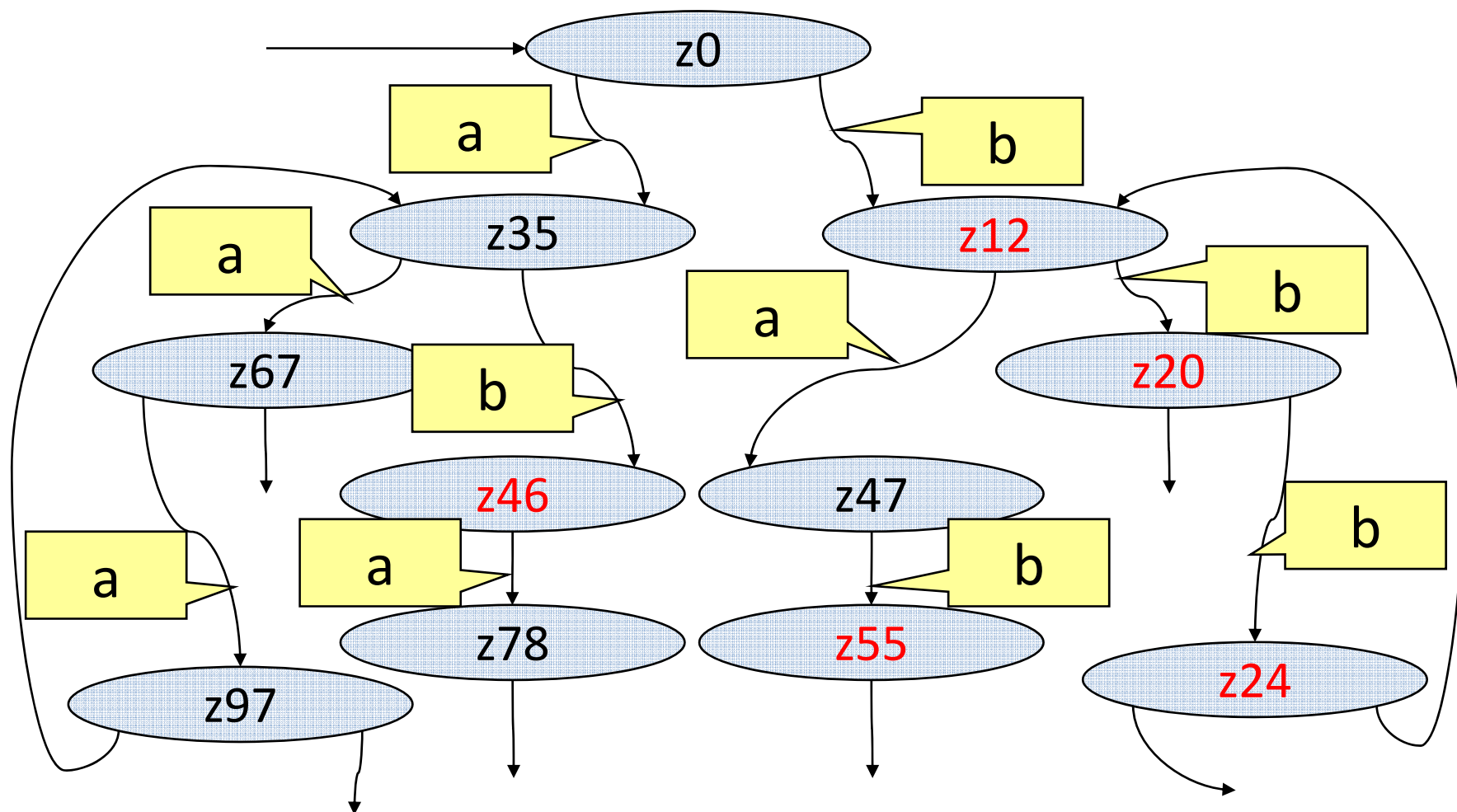
Streett 自动机

# Streett 自动机： 例子

- 标号集合： $\{ a, b \}$
- 状态集合： $\{ z_0, z_1, z_2, z_3, \dots \}$
- 迁移关系： $\{ (z_0, a, z_{35}), (z_0, b, z_{12}), \dots \}$
- 初始状态集： $\{ z_0 \}$
- 状态集合对的集合： $\{ (\{z_{35}\}, \{z_{67}\}), (\{z_{35}\}, \{z_{46}\}), (\{z_{35}, z_{12}\}, \{z_{97}, z_{24}\}), \dots \}$

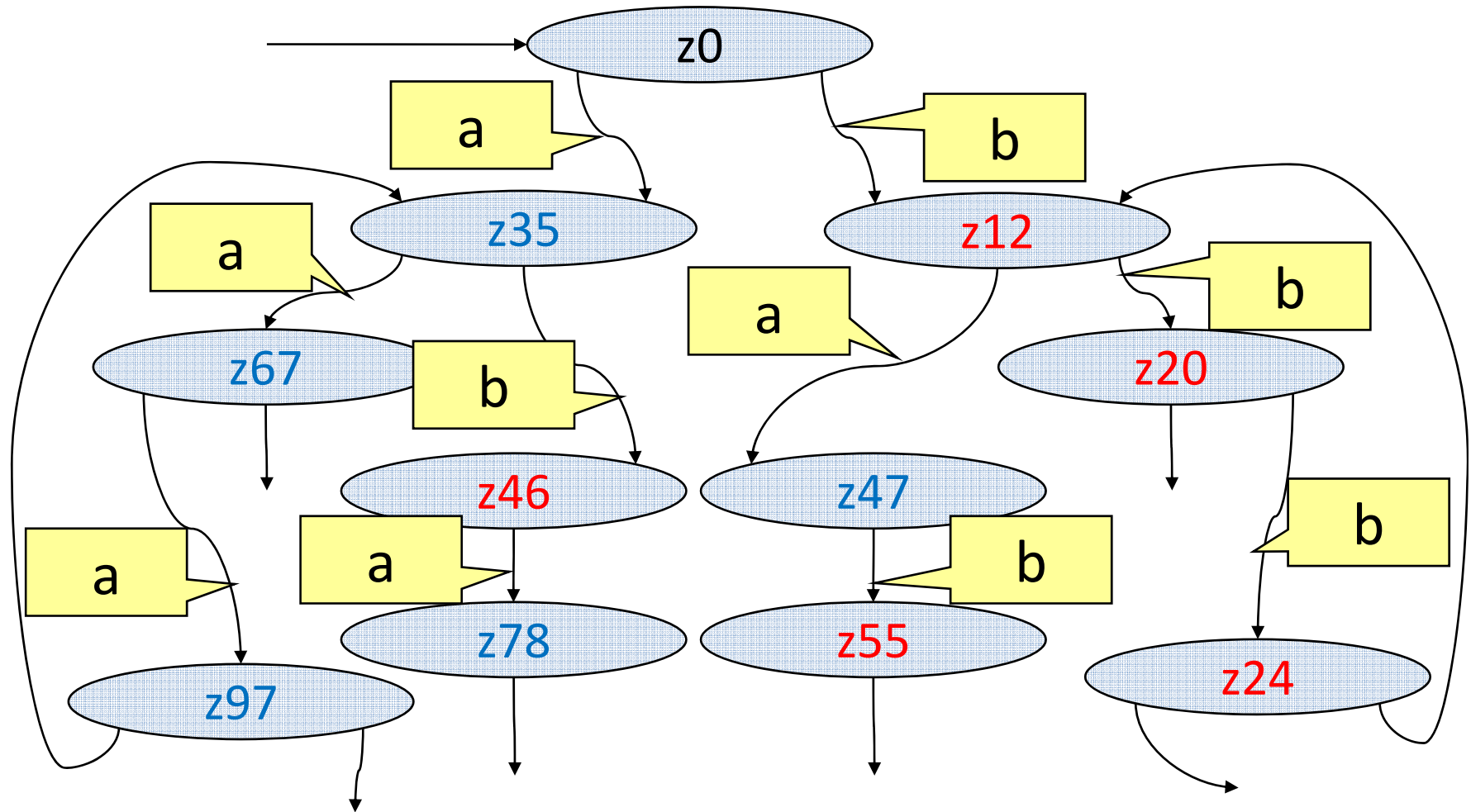
# 基于迁移的扩展Büchi自动机

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

- 抽象状态变化图:



a: 进程A的运行    b: 进程B的运行

# 基于迁移的扩展Büchi自动机

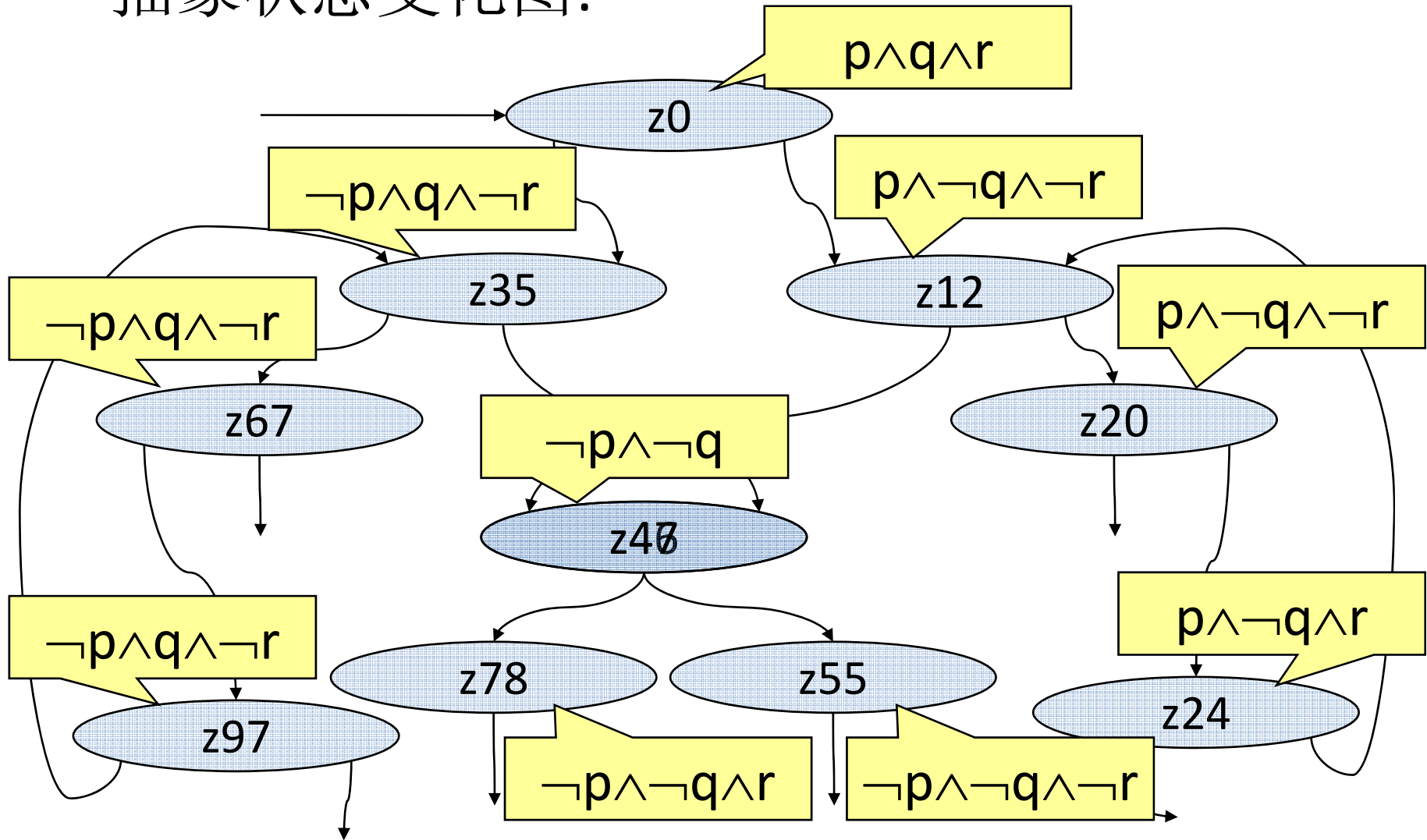
- 动作信息
  - 系统状态
  - 状态变化
  - 初始状态
  - 多元公平性
- 符号  
抽象状态  
三元组  
状态集合  
迁移集合的集合

# 基于迁移的扩展Büchi自动机

- 标号集合:  $\{ a, b \}$
- 状态集合:  $\{ z_0, z_1, z_2, z_3, \dots \}$
- 迁移关系(T):  $\{ (z_0, a, z_3), (z_0, b, z_1), \dots \}$
- 初始状态集:  $\{ z_0 \}$
- 迁移集合的集合:  $\{ \{ (x, a, y) \mid (x, a, y) \in T \}, \{ (x, b, y) \mid (x, b, y) \in T \} \}$

# 交错迁移系统

- 抽象状态变化图:



$p: a=s_0 \quad q: b=t_0 \quad r: t=0 \quad s: a=s_0 \wedge b=t_0$

# 迁移关系

$(z35, z67)$

$(z35, z46/z47)$

$(z35, \{z67\})$

$(z35, \{z46, z47\})$

$(z35, a, \{z67\})$

$(z35, b, \{z46, z47\})$

# 交错迁移系统

- 动作信息
- 系统状态
- 状态变化
- 初始状态

符号

抽象状态

三元组  $(S, \Sigma, 2^S)$

状态集合

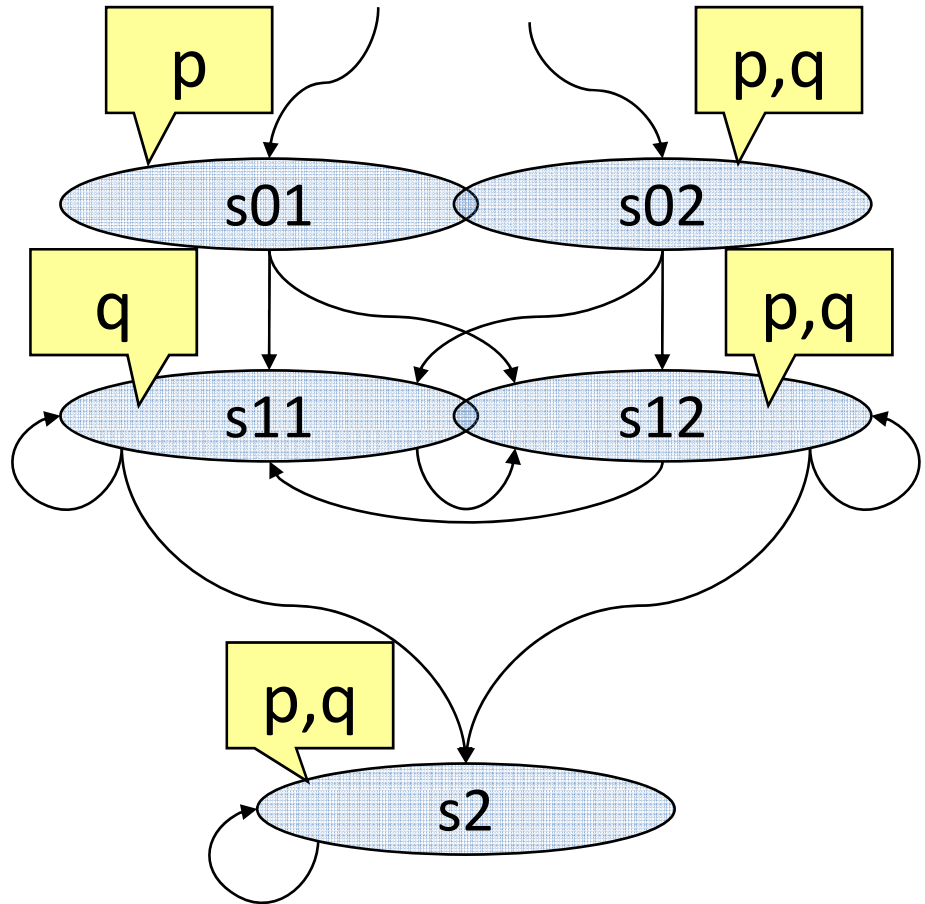
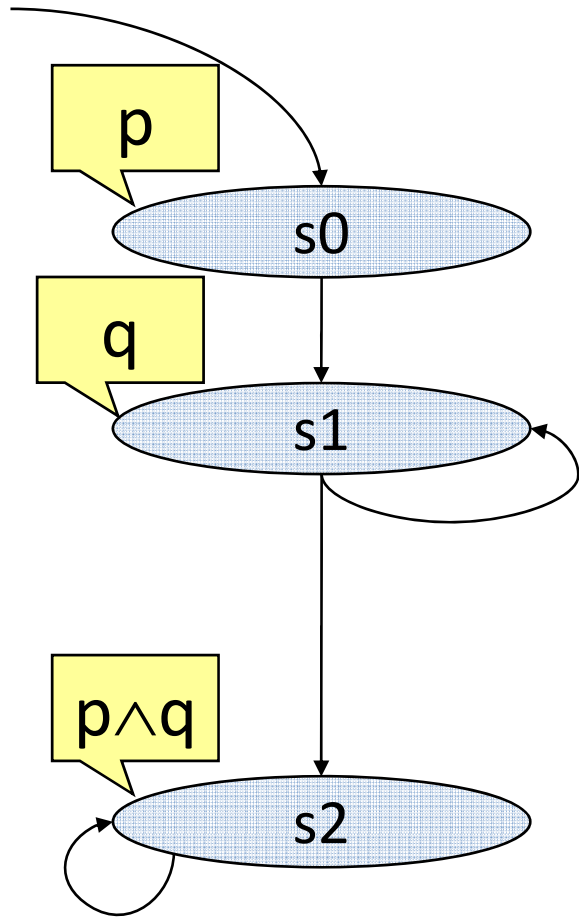


交错迁移系统

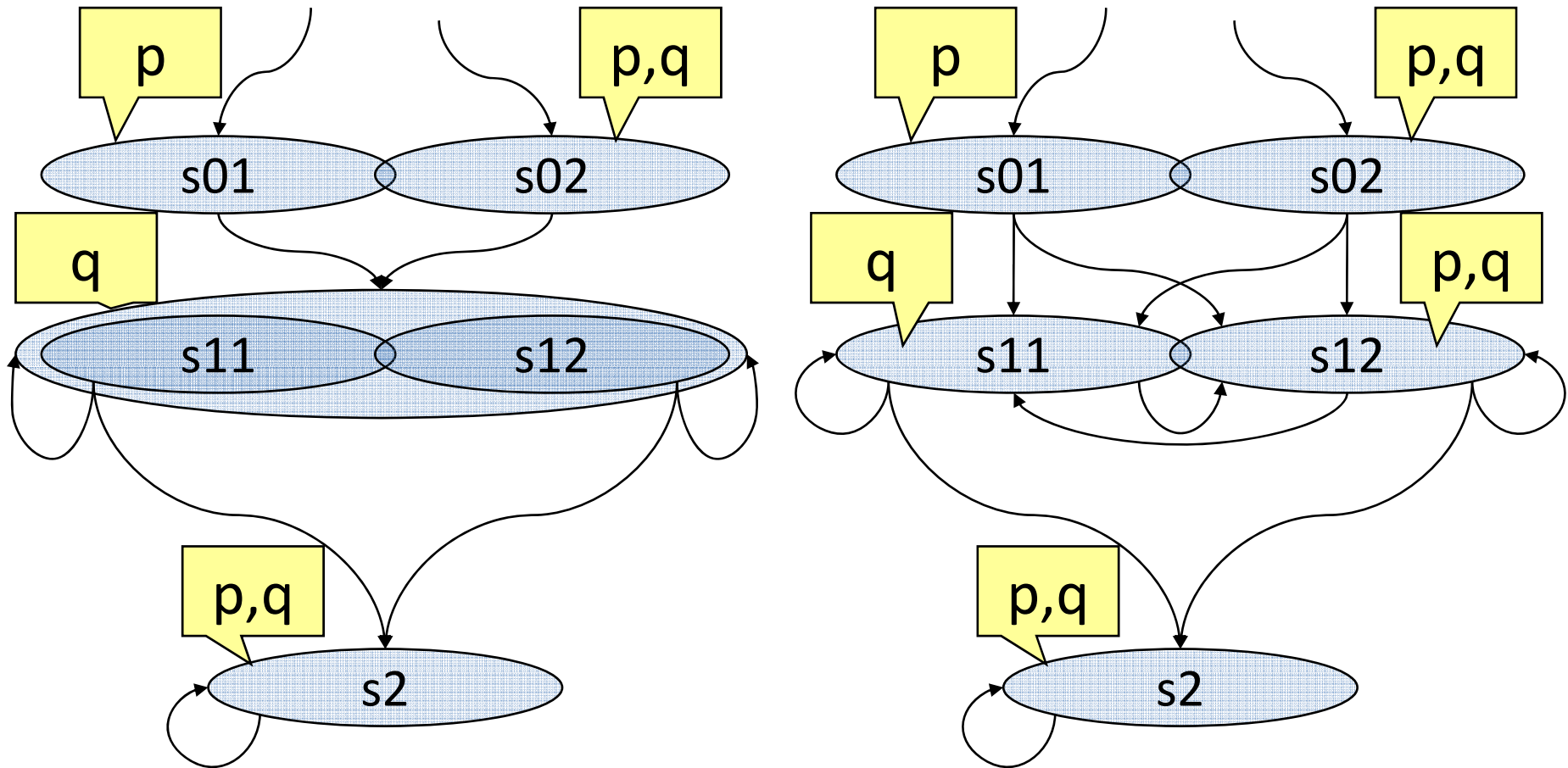
# 交错迁移系统：例子

- 标号集合：  $\{ a, b \}$
- 状态集合：  $\{ z_0, z_1, z_2, z_3, \dots \}$
- 迁移关系：  $\{ (z_0, a, \{z_{35}\}), (z_{35}, b, \{z_{46}, z_{47}\}), \dots \}$
- 初始状态集：  $\{ z_0 \}$

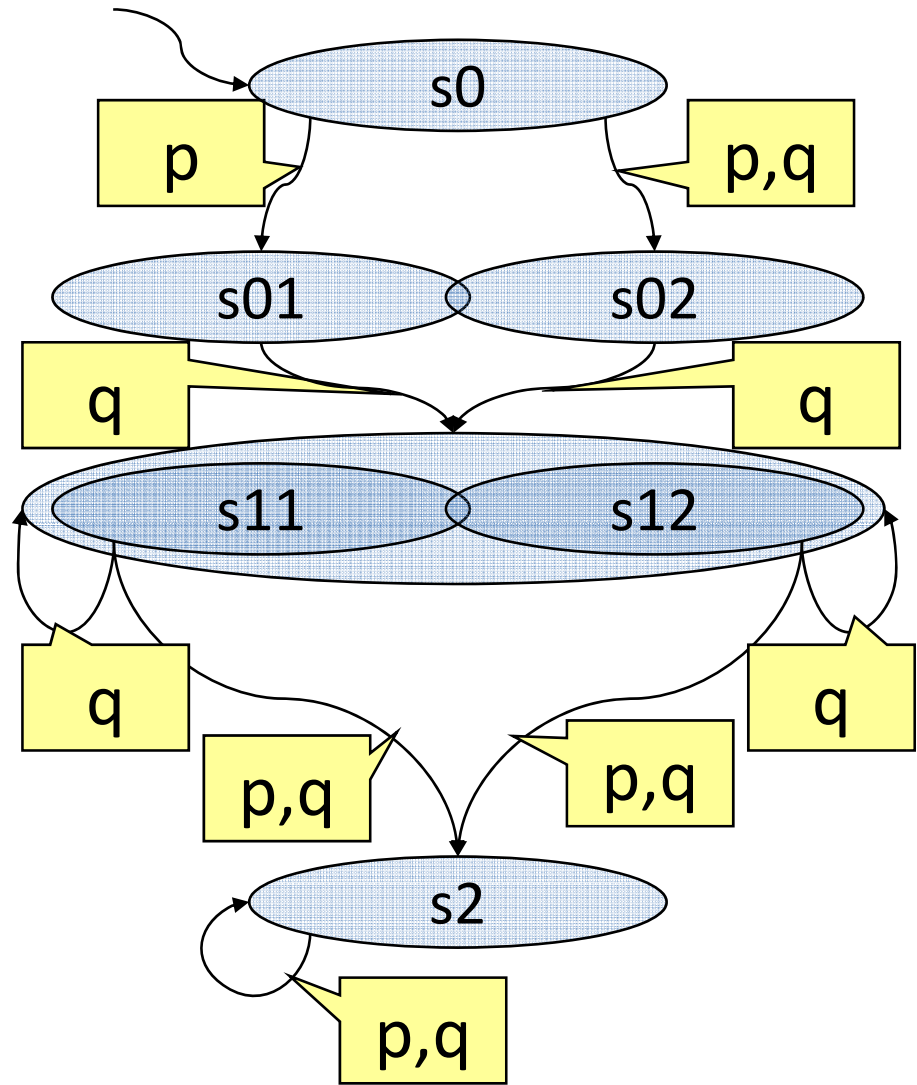
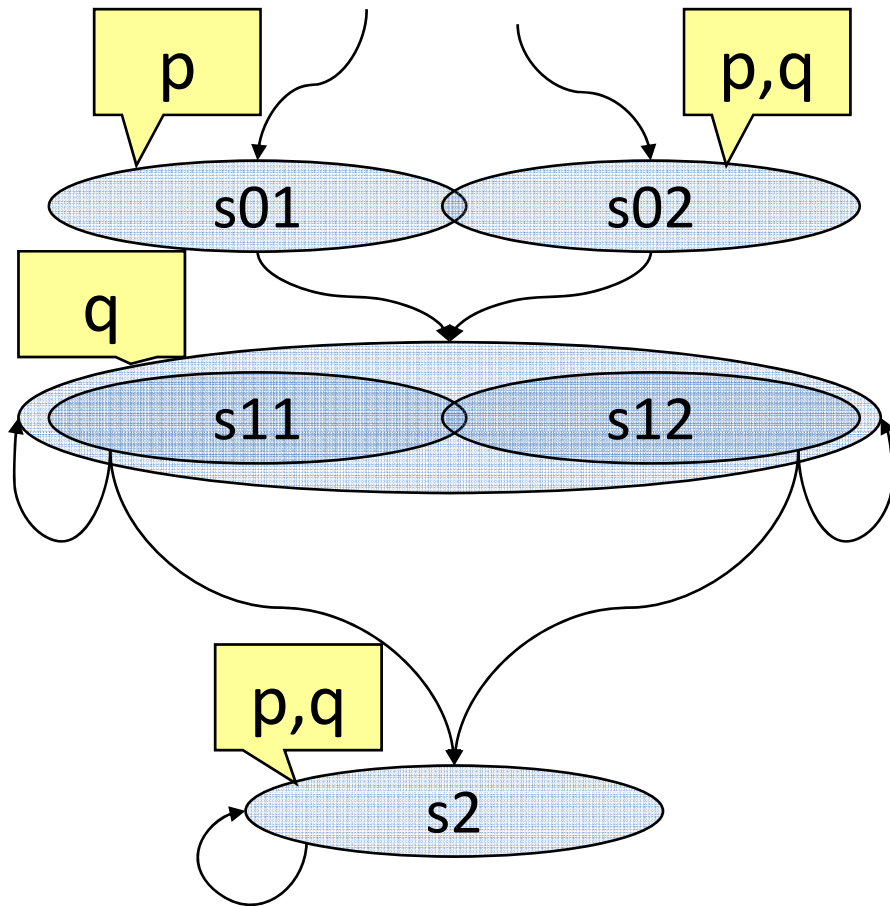
# 交错迁移系统：例2



# 交错迁移系统：例2



# 交错迁移系统：例2



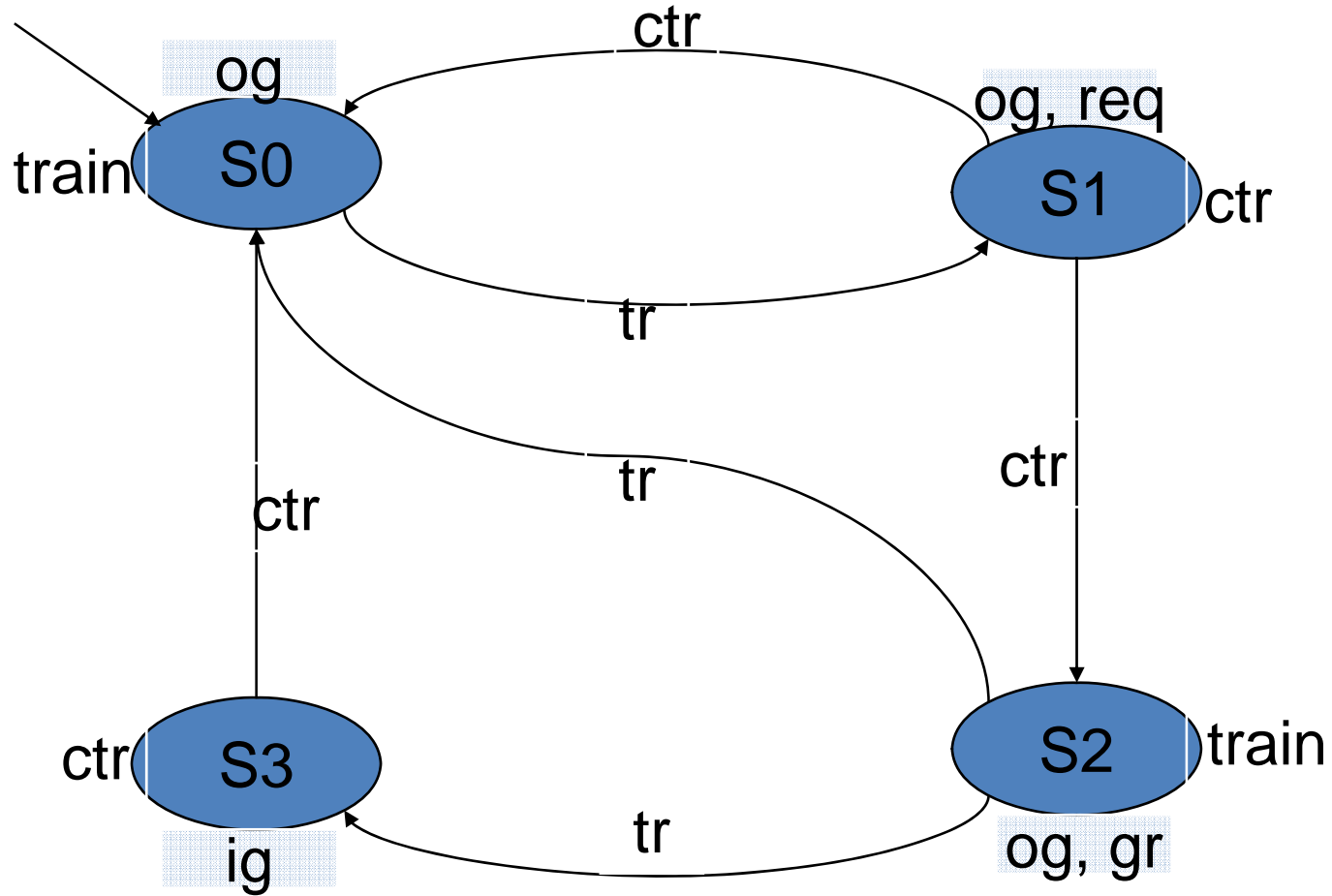
## 交错迁移系统：例2

$(s_0, p, \{s_{01}\}),$        $(s_0, pq, \{s_{02}\}),$   
 $(s_{01}, q, \{s_{11}, s_{12}\}),$   
 $(s_{02}, q, \{s_{11}, s_{12}\}),$   
 $(s_{11}, q, \{s_{11}, s_{12}\}),$        $(s_{11}, pq, \{s_2\}),$   
 $(s_{12}, q, \{s_{11}, s_{12}\}),$        $(s_{12}, pq, \{s_2\}),$   
 $(s_2, pq, \{s_2\})$

## 交错迁移系统：例2

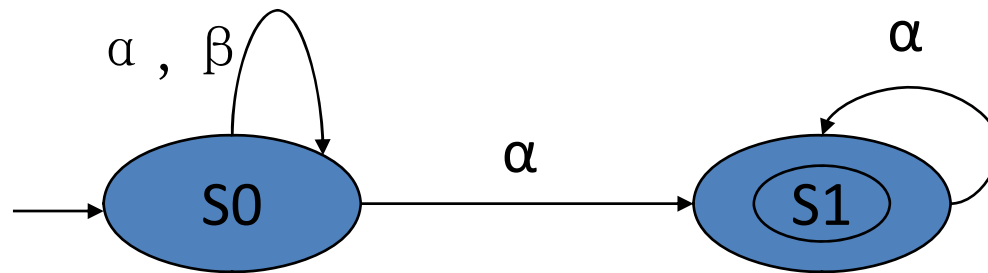
- 标号集合： $\{ p, q, pq, \varepsilon \}$
- 状态集合： $\{ s_0, s_{01}, s_{02}, s_{11}, s_{12}, s_2 \}$
- 迁移关系： $\{ (s_0, p, \{s_{01}\}), (s_0, pq, \{s_{02}\}), \dots \}$
- 初始状态集： $\{ s_0 \}$

# 火车进站控制



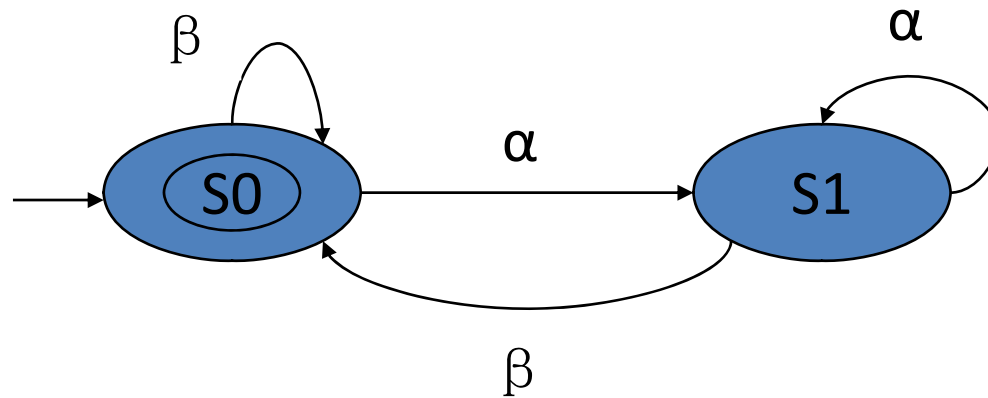
# 非确定型自动机/确定型自动机

# 非确定型Buchi自动机



$$L(A) = (\alpha + \beta)^* \alpha^\omega$$

# 确定型Buchi自动机



$$L(B) = L(\neg A) = (\alpha^* \beta)^\omega$$

# 确定型Buchi自动机的补

$$L(A) \neq (\alpha + \beta)^* \alpha^\omega$$

$\alpha^\omega$

$\alpha^{n1} \beta \alpha^\omega$

$\alpha^{n1} \beta \alpha^{n2} \beta \alpha^\omega$

...

$\alpha^{n1}$

$\alpha^{n1} \beta \alpha^{n2}$

$\alpha^{n1} \beta \alpha^{n2} \beta \alpha^{n3}$

...

无限多个 $\beta$ 、无限多次经过接受状态

# Buchi 自动机非空判定

start()

{

$W = A = B = \{\}$ ;

    for each initial state  $s \in I$ ,

        if ( $s$  is not in  $A$ ) { add  $s$  to  $W$ ; dfs1(); }

}

# Buchi 自动机非空判定

```
dfs1()
```

```
{
```

```
  q = last element from W;
```

```
  add q to A;
```

```
  for each successor state s of q,
```

```
    if (s is not in A) { add s to W; dfs1(); }
```

```
  if (accept(q)) { add q to B; dfs2(); }
```

```
  delete q from W;
```

```
}
```

# Buchi 自动机非空判定

```
dfs2()
```

```
{
```

```
  q = last element from B;
```

```
  for each successor state s of q,
```

```
    if (s is in W) report("nonempty");
```

```
    if (s is not in B) { add s to B; dfs2(); }
```

```
}
```

# 算法正确性和复杂度

- 算法报告 “nonempty” 当且仅当自动机的语言非空
- Buchi 自动机语言非空判定算法的复杂度是线性的