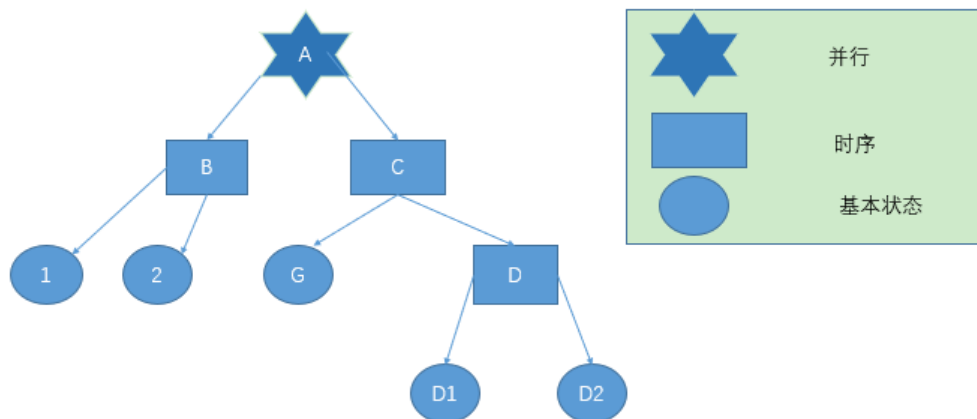


年级	15 级	专业（方向）	移动信息工程
学号	15352007	姓名	蔡林航
电话	13727021970	Email	1368963367@qq.com
开始日期	2017.10.29	完成日期	2017.10.30

1 状态图有层次地、同步地建模。状态图的状态间的转移可以被条件使能控制。同时，状态转移也可以和活动(action)联系到一起。通过活动 (action)，可以触发新的事件，也可以执行计算任务。

2 尽管在规模扩大时，状态图的性能比有限状态机要好，但是对于规模庞大的系统而言，状态图显得太过复杂，难以理解。这在模块重用 (re-use) 中会表现出劣势。状态图的与状态转移相关联的活动 (action) 提供了强有力的可扩展性，但是，这样的后果是，大规模使用活动会将系统的状态信息转移到变量上。被隐藏的状态使得系统分析变成棘手。

3



$$\begin{aligned}
 4 \quad Z_A &= Z_B \times Z_C = (Z_1 \cup Z_2) \times (Z_G \cup Z_D) = (Z_1 \cup Z_2) \times (Z_G \cup (Z_{D1} \cup Z_{D2})) \\
 &= (Z_1, Z_G) \cup (Z_1, Z_{D1}) \cup (Z_1, Z_{D2}) \cup (Z_2, Z_G) \cup (Z_2, Z_{D1}) \cup (Z_2, Z_{D2})
 \end{aligned}$$

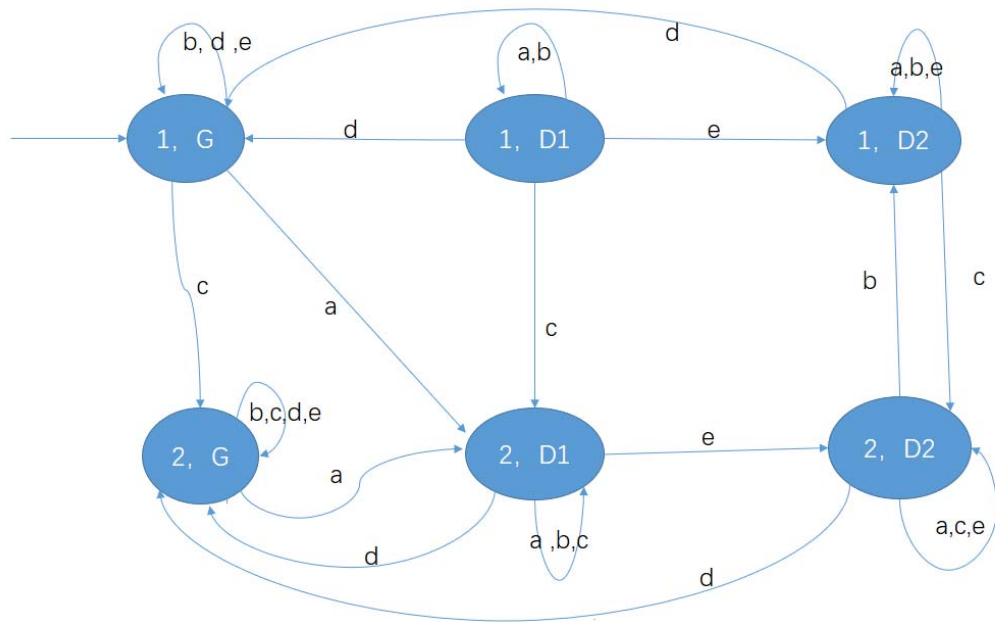
5

事件 Event	状态 B	状态 C	状态 A
初始	1	G	1,G

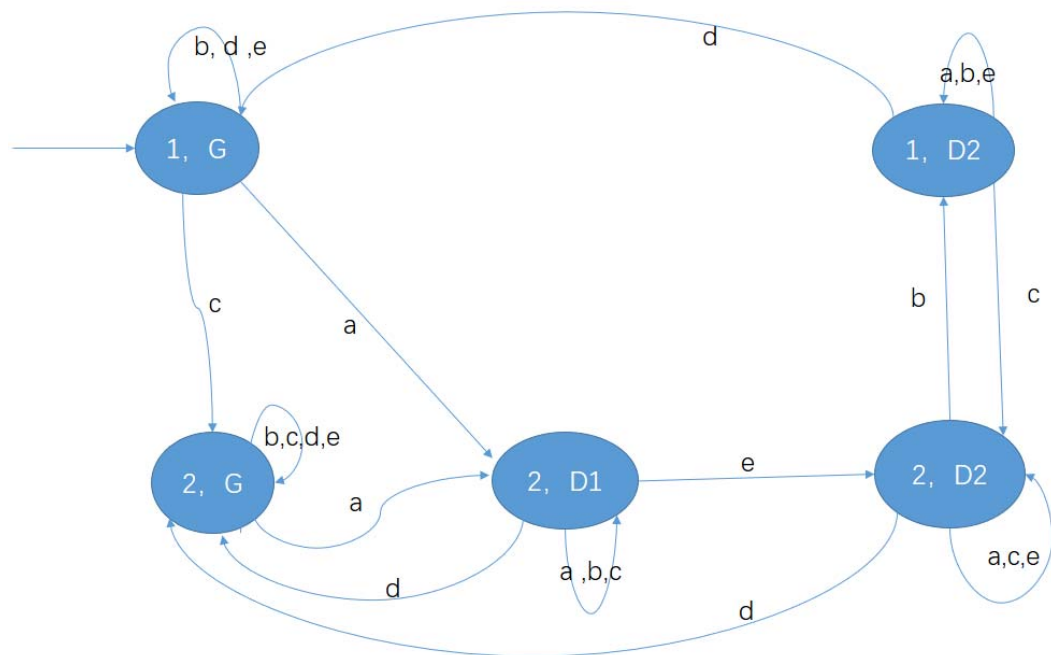
a	2	D1	2,D1
b	2	D1	2,D1
e	2	D2	2,D2
b	1	D2	1,D2
d	1	G	1,G
b	1	G	1,G

6

等效的没有最小化状态数量的有限状态机如下：



最小化状态数量的有限状态机如下：



7.

插入硬币，点一杯茶：

$$\begin{aligned}
 A_1.0 &\xrightarrow{\text{coin_in} / \text{ok}} A_1.1 \\
 A_2.A &\xrightarrow{\text{ok} /} A_2.B \\
 A_2.B &\xrightarrow{\text{req_tea} / \text{start_tea}} A_2.D \\
 A_2.D &\xrightarrow{\text{drink_ready} / \text{done}} A_2.A \\
 A_1.1 &\xrightarrow{\text{done} /} A_1.0
 \end{aligned}$$

利用 bug，不花钱就能买茶：

$$\begin{aligned}
 A_1.0 &\xrightarrow{\text{coin_in} / \text{ok}} A_1.1 \\
 A_2.A &\xrightarrow{\text{ok} /} A_2.B \\
 A_2.B &\xrightarrow{\text{req_tea} / \text{start_tea}} A_2.D \\
 A_1.1 &\xrightarrow{\text{cancel} / \text{coin_out} \quad , \text{resrt}} A_1.0 \\
 A_2.D &\xrightarrow{\text{drink_ready} / \text{done}} A_2.A
 \end{aligned}$$

修复这个 bug 的状态图：

Input events
from the environment



coin_in



cancel



drink_ready

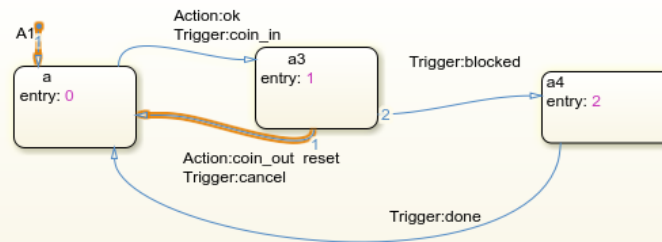


req_coffee



req_tea

?



Output events
to the environment



start_coffee



start_tea



coin_out

A2

