14353120 焦子鸣 14M1

ex2:

1. 状态图的优点:

I. 状态图形式简单,是对具体问题的抽象,便于直观体会不同状态的关系

II. 层次清晰,如何到达,离开一个状态一目了然.

III. 容易实现,方便人们画出并使用,许多工具都可画出

2. 状态图的缺点:

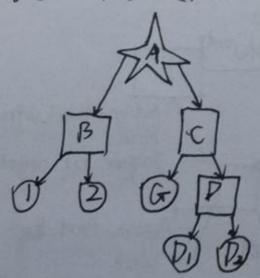
I. 不适合用来表示分布式的应用,因为难以表示

II. 一些非功能性的行为不能描述

III. 一般用于描述面向过程的情况,需要描述面向对象的思想不可用.

3. 将所给的状态图转换为树状表达:

基本状态用圆形表示;或类型超状态用矩形表示;与类型超状态用星形表示



4. 将所给的状态图转换为状态集表达:

首先明确:

基本状态的状态集就是它本身

或类型超状态的状态集是它所有子集的并集

与类型超状态的状态集是它所有子集的笛卡尔积

$$Z_A = Z_B \times Z_C$$

$$= (Z_1 \cup Z_2) \times (Z_G \cup Z_D)$$

$$= (Z_1 \cup Z_2) \times [Z_G \cup (Z_{D_1} \cup Z_{D_2})]$$

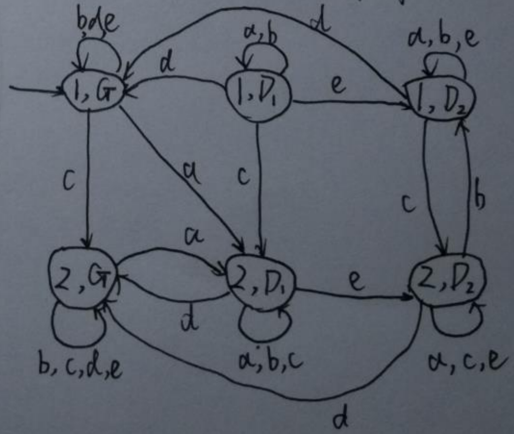
$$= (Z_1, Z_G) \cup (Z_1, Z_{D_1}) \cup (Z_1, Z_{D_2}) \cup (Z_2, Z_G) \cup (Z_2, Z_{D_1}) \cup (Z_2, Z_{D_2})$$

5. 分析:

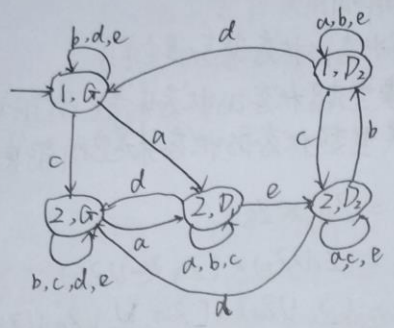
事件	状态B	状态C	状态A
初始状态	1	G	1, G
a	2	D_1	2, D_1
b	2	D_1	2, D_1
e	2	D_2	2, D_2
b	1	D_2	1, D_2
d	1	G	1, G
b	1	G	1, G

6. 将所给的状态图转换为有限状态机:

作为基本状态及其外部触发事件:



去除无法到达的基本状态



7. I. $A_1.D \xrightarrow{coin\_in/ok} A_1.I$

$A_2.A \xrightarrow{ok/} A_2.B$

$A_2.B \xrightarrow{req\_tea/start\_tea} A_2.D$

$A_2.D \xrightarrow{drink\_ready/done} A_2.A$

$A_1.I \xrightarrow{done/} A_1.D$

II. $A_1.D \xrightarrow{coin\_in/ok} A_1.I$

$A_2.A \xrightarrow{ok/} A_2.B$

$A_2.B \xrightarrow{req\_tea/start\_tea} A_2.D$

$A_1.I \xrightarrow{cancel/coin\_out,reset} A_1.D$

$A_2.D \xrightarrow{drink\_ready/done} A_2.A$

III. 作为一个修复bug的状态图：