

# 嵌入式系统理论作业

曹广杰

15352015 数据科学与计算机

授课教师：黄凯

2017/10/27

## 状态图之于状态机的优势在哪

“状态图（Statechart Diagram）是描述一个对象基于事件驱动的动态行为，描述了该对象如何根据当前状态对不同的事件或特定的时间点做出的反应。状态图是状态机的图形化表示，状态图显示了一个状态机，包含了状态机的所有特性。”<sup>1</sup>

状态图和状态机可以描述该对象对于特定时间和特定事件的行为，但是相对于状态机而言：

1. “状态图是状态机的图形化表示”，这使得状态图更加直观，具有较高的可读性——由于图形化的表现形式，便于开发人员的沟通与理解；
2. 状态图可以有层次、同步地建模。  
“包含了状态机的所有特性”，同时根据PPT上的阐述，由于状态图可以实现base状态和super状态的组合，状态图可以包含更多的信息，甚至可以包含更多的状态机模型；
3. 状态之间的转移可以被条件和活动控制，活动触发新的事件，事件触发新的活动，如此可以执行计算任务；

## 状态图的劣势

1. 由于状态图的极度灵活性，故而从语法上实现状态图的构建可能会遇到很多困难，而状态机的Verilog语言体系已经非常成熟了；
2. 规模过大的系统，状态图过于复杂。状态信息过多地转移到变量上，隐藏状态的出现，状态分析更加难于入手；
3. 由于状态图的体系较之于状态机更为碎片化，则当需要表示的状态转移以及对应的处理情况更为复杂的时候，状态图的可读性就会下降；

## 状态图到状态树的转换

Given the StateChart in Figure 1. Draw the state space of the StateChart as a tree, which shows the hierarchy of states and denotes the state types (basic state, sequential states, and parallel states).

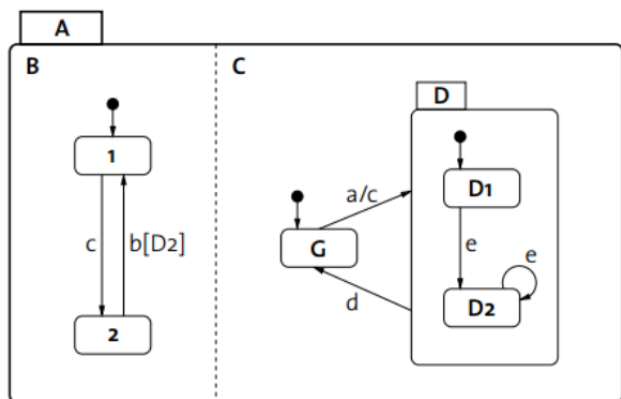
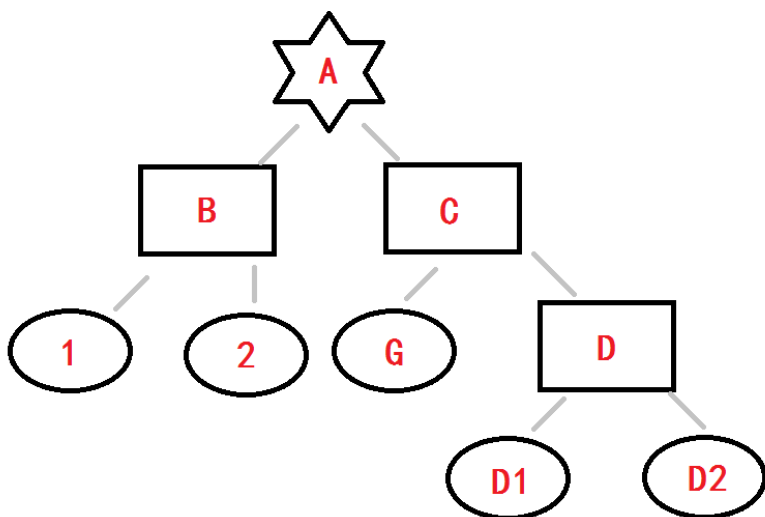


Figure 1: StateChart

对应的树状图——没有触发条件的树状图：



### 状态数目的计算

为了计算状态产生的数量，需要先整理当前最宏观的super状态：

$$\begin{aligned}
 A &= (a \cup b) \times (G \cup D1 \cup D2) \\
 &= (a \times (G \cup D1 \cup D2)) \cup (b \times (G \cup D1 \cup D2)) \\
 &= (a, G) \cup (a, D1) \cup (a, D2) \cup (b, G) \cup (b, D1) \cup (b, D2)
 \end{aligned}$$

综上，共有六种状态。

### 状态转移

已知有一系列触发事件发生：a,b,e,b,d,b——计算figure1的状态转移：

- 初始活跃状态：G, 1, A, B, C 【=> 表示转移】，列表中表示每一个阶段新增加的状态转移

Event	super state B	super state C	super state A
Initial	1	G	1, G
a	2	D1	2, D1
b	2	D1	2, D1
e	2	D2	2, D2
b	1	D2	2, D2
d	1	G	1, G
b	1	G	1, G

```

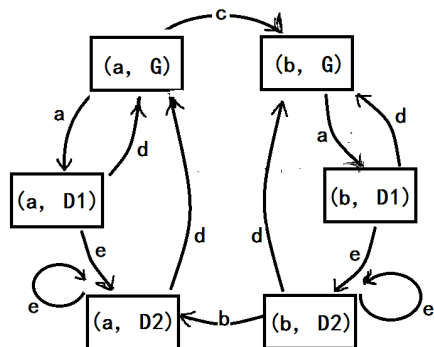
a: (1, G) => (1, D1); (2, G) => (2, D1)-- 此时/c表示触发了c的event效果
b: null -- 因为D2还没有任何状态
e: (1, D1) => (1, D2); (1, D2) => (1, D2);
    (2, D1) => (2, D2); (2, D2) => (2, D2)
b: (2, D2) => (1, D2)
d: (1, D2) => (1, G); (2, D2) => (2, G)
b: null -- D2的状态已经被转移, 当前D2处于基本状态

```

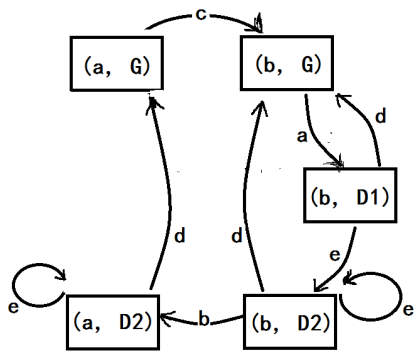
## 状态图转状态机

- 由之前状态数量的计算, 得知有6种基本状态
- 由以上状态转移的分析, 得知转移的条件

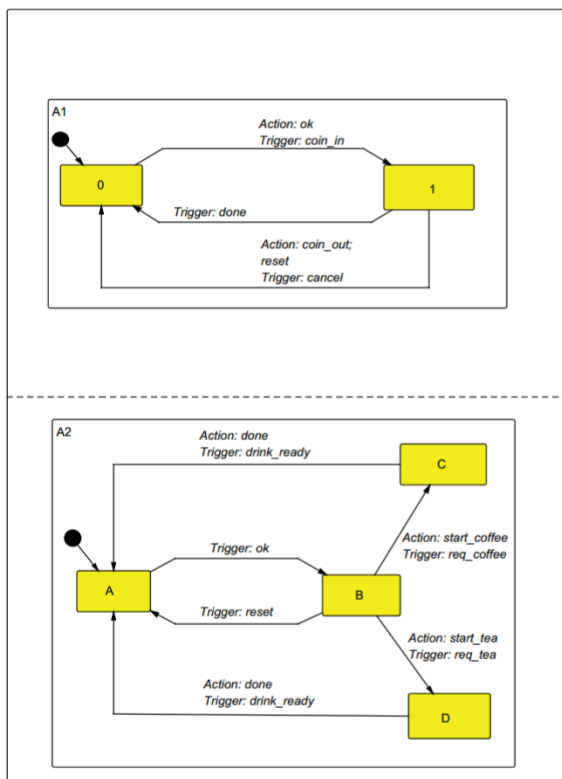
以下为实现6种基本状态之间的转移:



最小化状态转移——删掉入度为0的状态



## StateChart model of a vending machine



描述购买状态的发生

1. 在上面的状态机中，投入硬币则触发发生，发出饮品请求，结束触发，硬币收纳，重置触发

0 => 1

2. 下面的状态机中：

- 随着投入硬币A触发发生并进入B

A => B

- 由上面超级状态的输出作为触发，进入CD

B => C; B => D

- C、D状态将饮品投出，B状态重置A状态的触发

D => A, C => A, B => A

解释当前零售机的bug

Bug出现在上面的super state中，如果在获得饮品之后取消订单，就可以不用付款。

绘制修复bug之后的状态图

