



状态图

PEREC

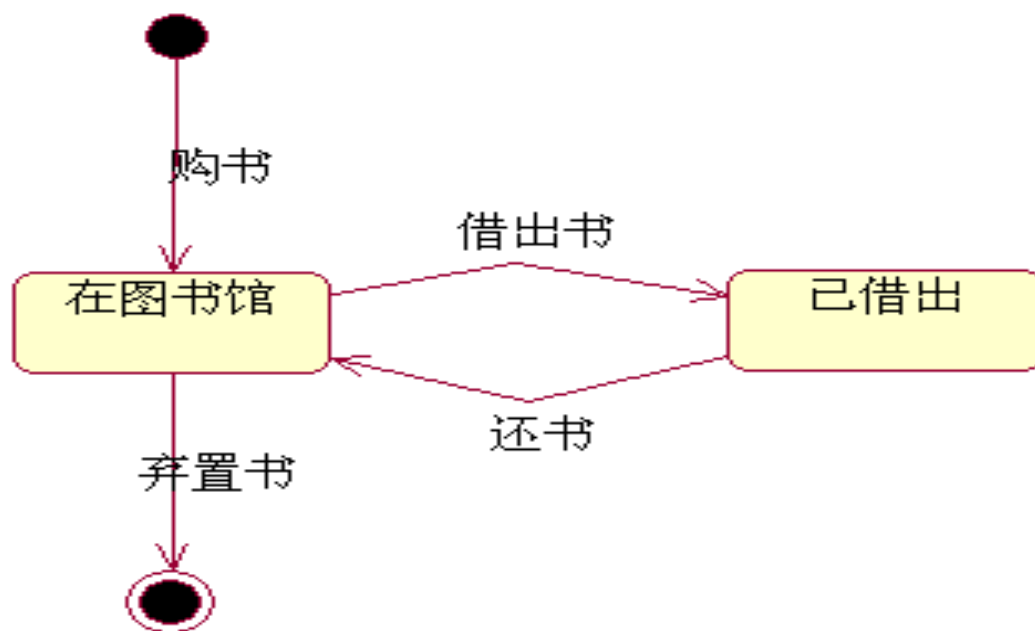
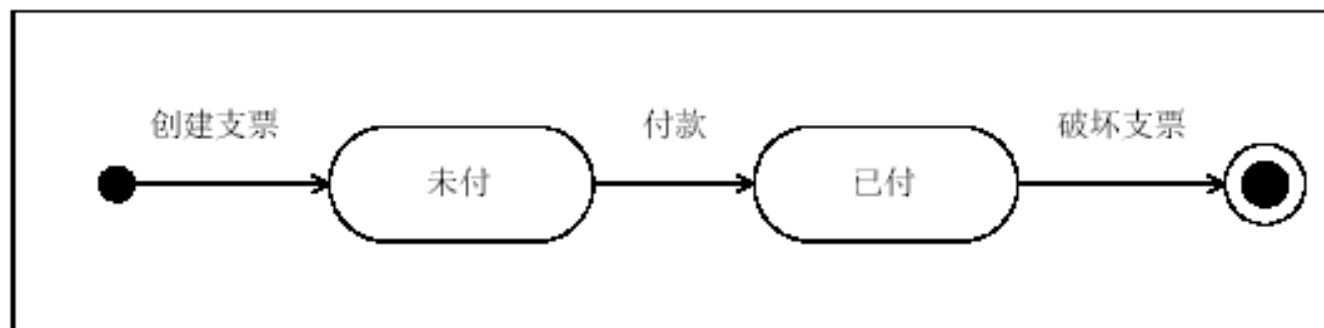


一 状态图的定义

№ 2

- ⊕ 状态定义为对象行为在某个时刻的快照或者转折点
- ⊕ 每个对象在其生命存在期间都有不同的存在状态
 - 支票 已付
 - 汽车 停
 - 发动机 运行
 - 小王 已婚
 - 图书馆中图书 在馆 或不在馆
- ⊕ 状态图是用来建模对象是如何改变其状态的

PEREC





二 状态图的标记符

№ 4

- ⊕ 状态
- ⊕ 转移
- ⊕ 决策点
- ⊕ 同步
- ⊕ 状态的动作
- ⊕ 状态的事件

PEREC



2.1 状态

№ 5

在状态图中,有三种类型的状态:

⊕ 开始状态

⊕ 结束状态

⊕ 标准状态



标准状态

PEREC



注意:

№ 6

- ⊕ 状态图不一定同时具有开始和结束状态
- ⊕ 状态图中可以有0到1个开始状态
- ⊕ 状态图中可以具有多个结束状态

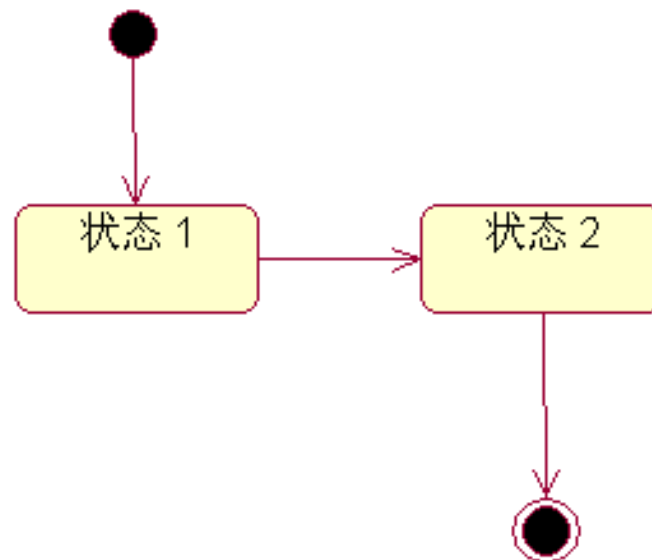
PEREC



2.2 转移

No 7

- ⊕ 转移用来显示从一个状态到另一个状态的处理流
- ⊕ 使用的标记符是一个开放箭头

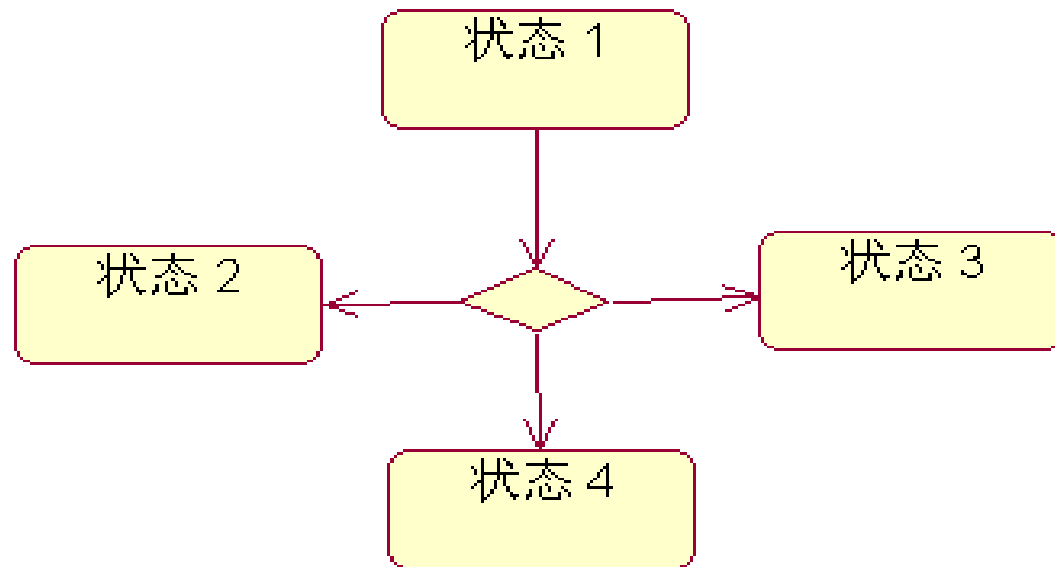


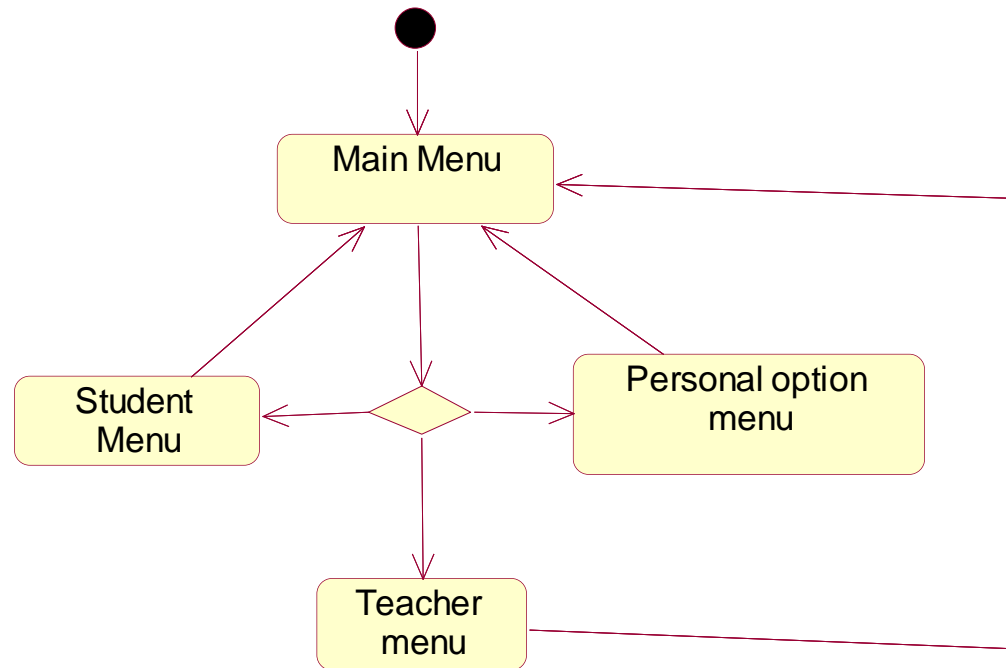


2.3 决策点

No 8

- ⊕ 决策点用来说明从同一个状态向不同的状态进行转移
- ⊕ 可以提高状态图的可读性
- ⊕ 是一个空心的菱形



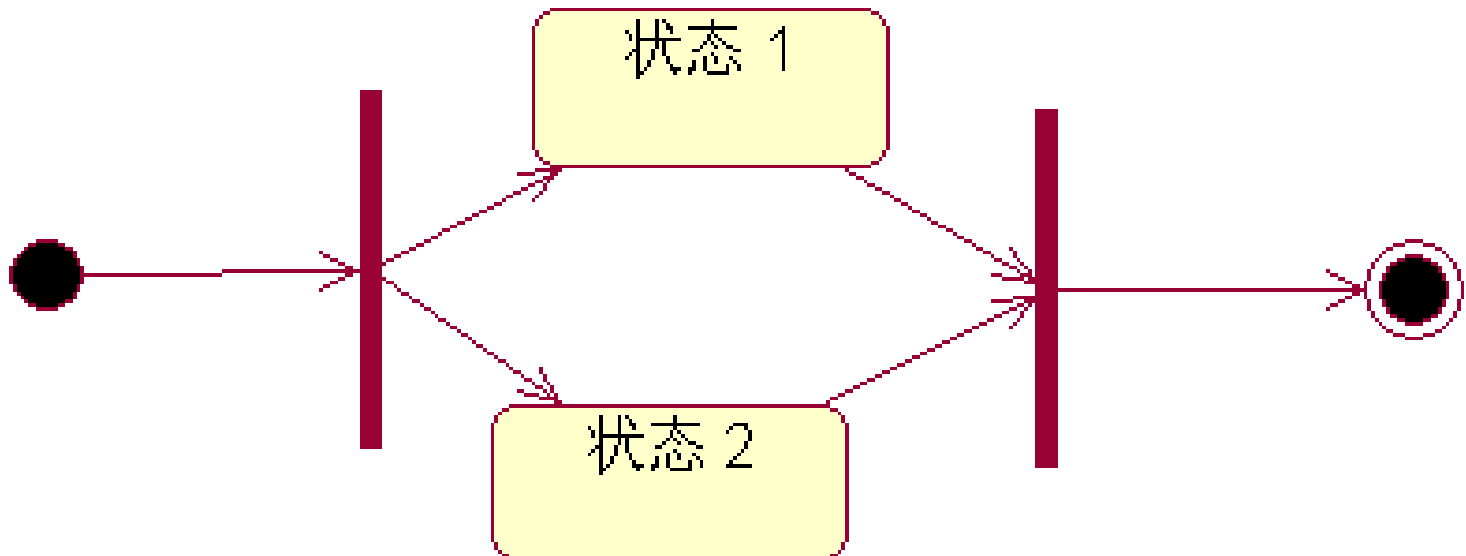


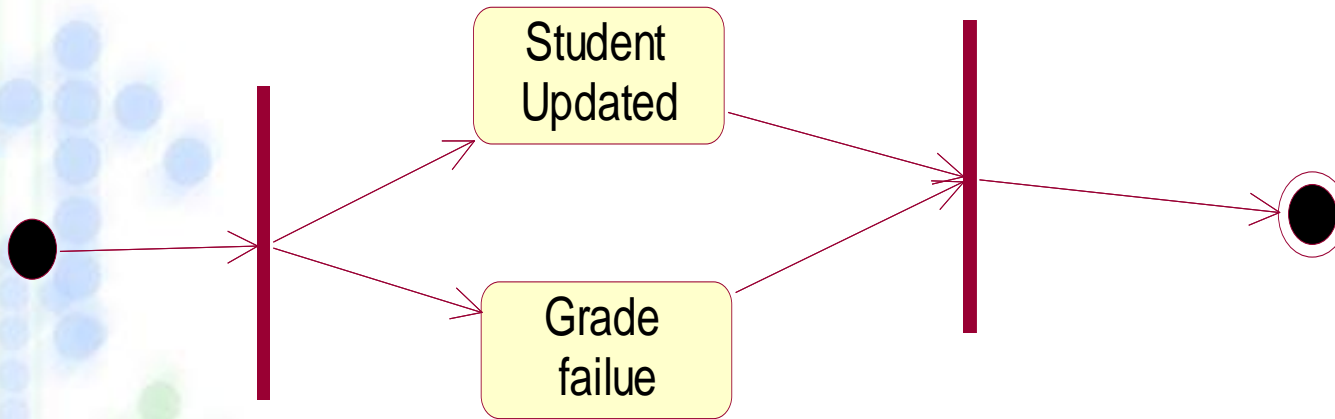


2.4 同步

№ 10

- ⊕ 同步条是用来显示并发的状态,即不同的状态在同步条处需要同时满足开始或者结束的要求
- ⊕ 同步条必须配对使用







2.5 状态的事件

№ 12

- ⊕ 事件用来指示什么触发了状态的转移
- ⊕ 标记符就是在转移箭头上的文字



PEREC



2.6 动作

№ 13

⊕ 动作用来说明当事件发生时会产生什么情况

⊕ 初始状态可以有以下5种基本动作类型:

Entry :用来指定进入状态时发生的动作

Exit : 用来指定状态被另一个状态取代时发生的动作

Do : 用来指定出状态时发生的活动

Include : 调用由另一个状态图表示的子机器

Event : 用来指定当特定的事件触发时发生的动作

PEREC



动作有两种标记符

№ 14

- ⊕ 第一种用于entry,exit,do和include动作类型,形如:基本动作类型/动作

例如: entry/numberOfStudents=0

exit/classes->include(this)

do/refreshStudentList

include/performSomeSubtask

PEREC



⊕ 第二种标记符用于Event动作类型,这种情况下事件会触发动作,形如:

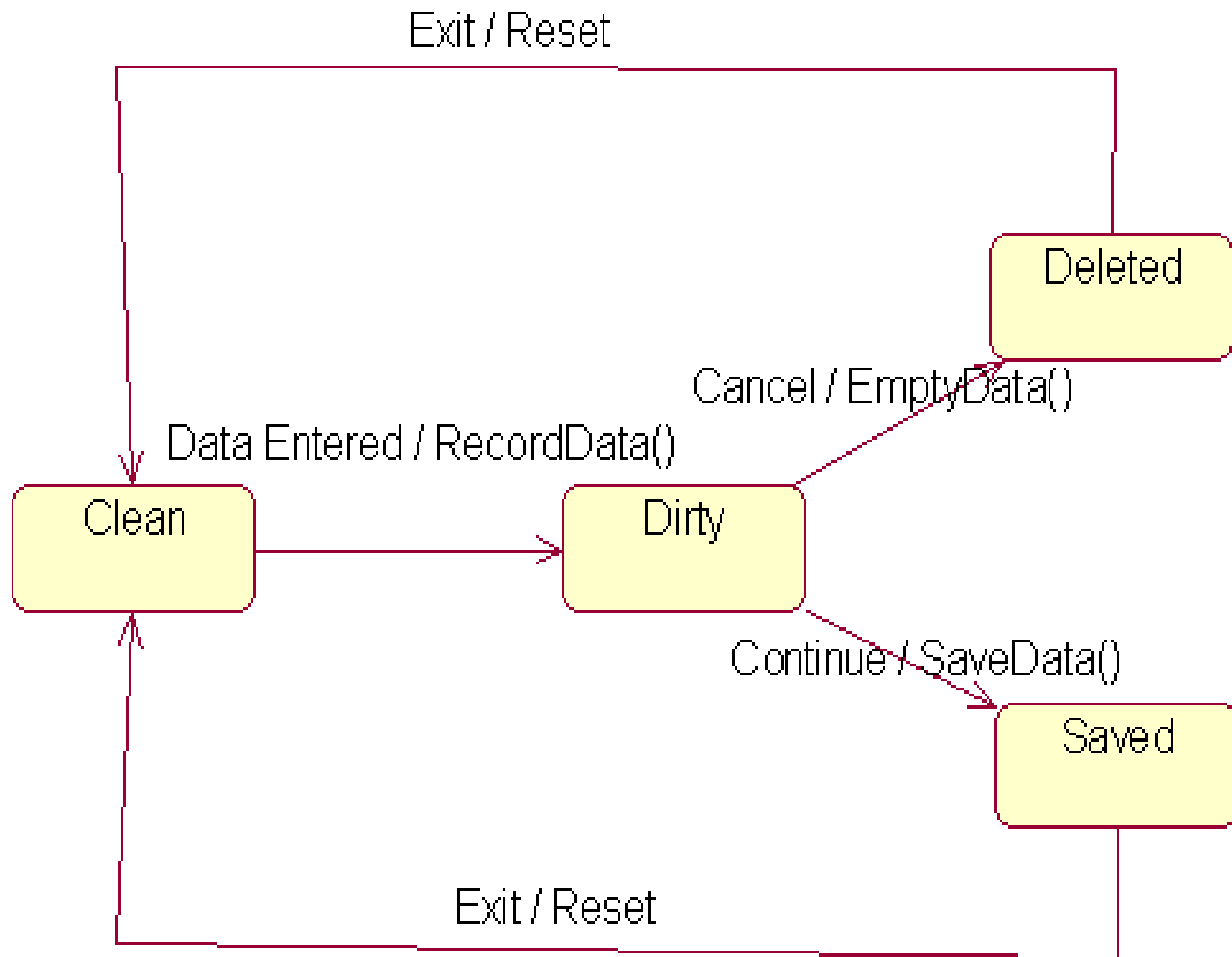
event-name(parameters)[guard-condition]/action

事件名(参数)[条件]/动作

例如:

ResetUsers(users)[users->forall(open=false)]/users->empty

PEREC





三 学习如何使用合成状态

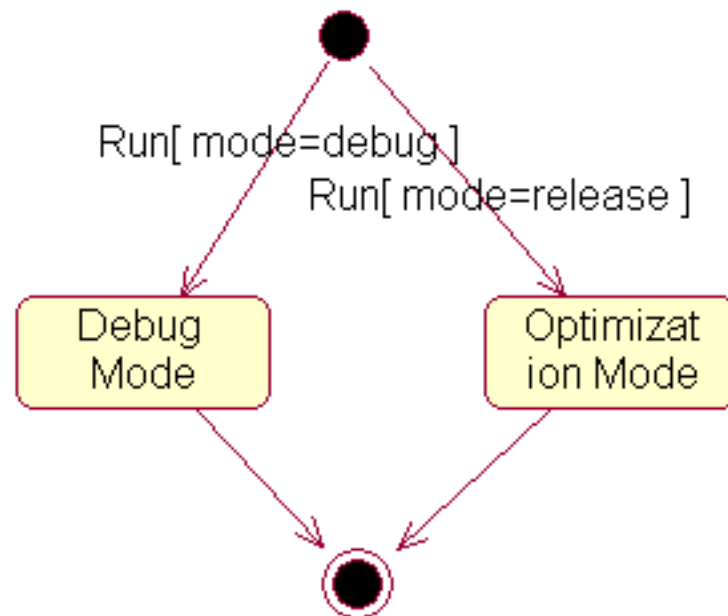
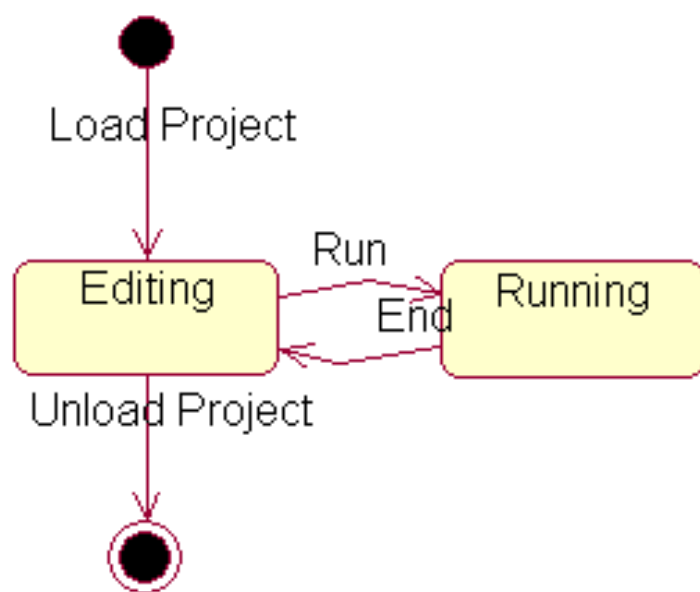
№ 17

- ⊕ 一个对象可以通过同步操作同时拥有多个状态
- ⊕ 有时对象还可以拥有层次结构的状态

3.1 子状态

是状态图中的某个状态独有的状态,可以使用另一个单个状态图来建模,即用一个状态图进一步详细的建模一个状态

PEREC

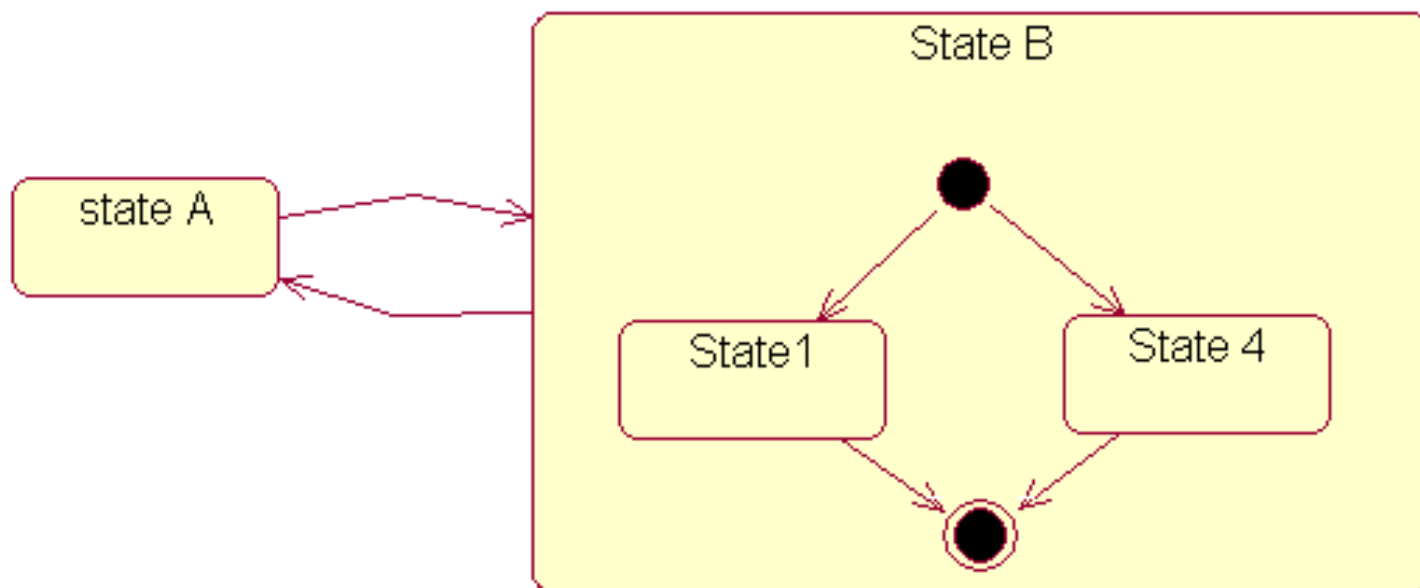




3.2 合成状态

合成状态通过创建一个较大的状态,然后在其中嵌入子状态来建模

PEREC





四 如何建模状态图

No 21

- ⊕ 标识出需要进一步建模的实体
- ⊕ 标识出每一个实体的开始和结束状态
- ⊕ 确定与每一个实体相关的事件
- ⊕ 从开始状态开始创建状态图
- ⊕ 如果必要则指定建立合成状态

PEREC



状态图作业

№ 22

考察一个课程对象(Course)的状态变化过程.

课程对象被创建,添加到数据库中,管理员可以删除,修改课程信息,在某个学期,开设课程,如果选修人数超过指定人数,就不再允许学生选这门课.学期结束,课程状态中止.

PEREC

