“十二五”普通高等教育本科国家级规划教材

北京高等教育精品教材

21世纪软件工程专业规划教材

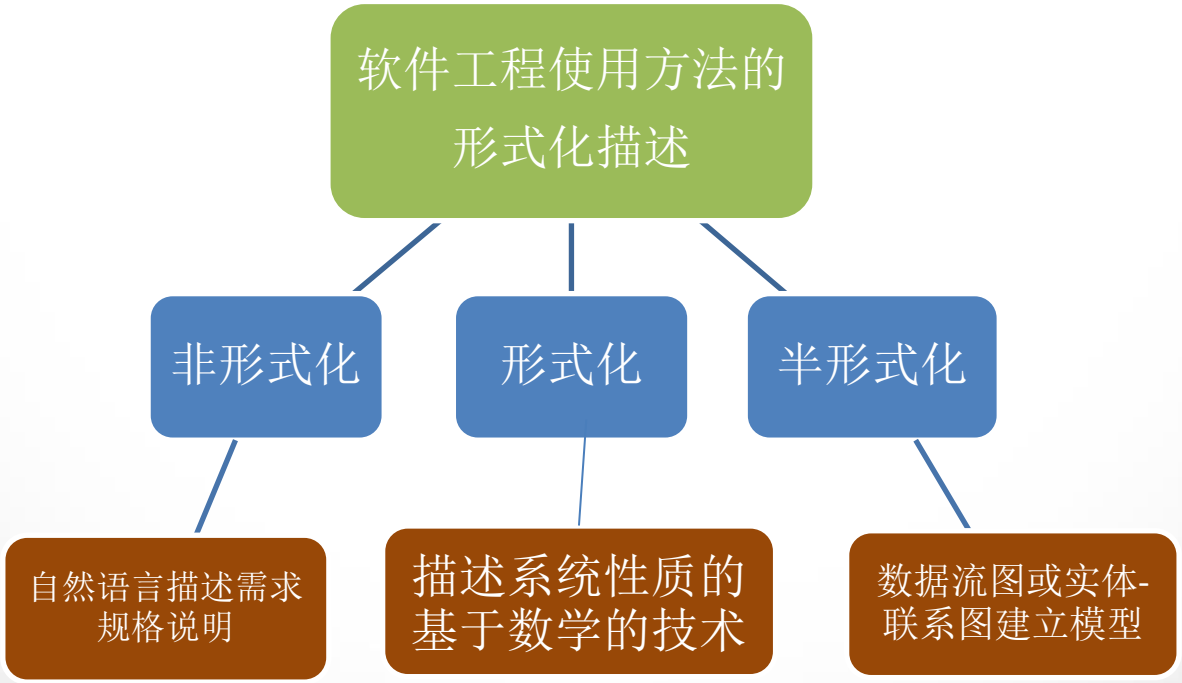
软件工程导论（第6版）

第4章　形式化说明技术

张海藩，牟永敏编著

清华大学出版社

本章将介绍 形式化技术在软件工程中有效的提高了开发的效率、改进了软件开发的质量、减少了开发费用。形式化的技术容易在软件的规约上取得一致性，它属于一种非常有效的交流方式。



主要内容

4.1 概述

4.2 有穷状态机

4.3 Petri网

4.4 Z语言

主要内容

4.1 概述

4.1.1 非形式化方法的缺点

用自然语言书写的系统规格说明书，可能存在矛盾、二义性、含糊性、不完整性及抽象层次混乱等问题。

① 矛盾是指一组相互冲突的陈述。

② 二义性是指读者可以用不同方式理解的陈述。

③ 含糊性

④ 不完整性

⑤ 抽象层次混乱是指在非常抽象的陈述中混进了一些关于细节的低层次陈述。

4.1 概述

4.1.2 形式化方法的优点

为了克服非形式化方法的缺点，人们把数学引入软件开发过程，创造了基于数学的形式化方法。

① 数学能够简洁准确地描述物理现象、对象或动作的结果，因此是理想的建模工具。

② 数学以在不同的软件工程活动之间平滑地过渡。

③ 数学提供了高层确认的手段。

4.1 概述

4.1.3 应用形式化方法的准则

① 应该选用适当的表示方法。

② 应该形式化，但不要过分形式化。

③ 应该估算成本。

④ 应该有形式化方法顾问随时提供咨询。

⑤ 不应该放弃传统的开发方法。

4.1 概述

4.1.3 应用形式化方法的准则

⑥ 应该建立详尽的文档。

⑦ 不应该放弃质量标准。

⑧ 不应该盲目依赖形式化方法。

⑨ 应该测试、测试再测试。

⑩ 应该重用。

主要内容

4.1 概述

4.2 有穷状态机

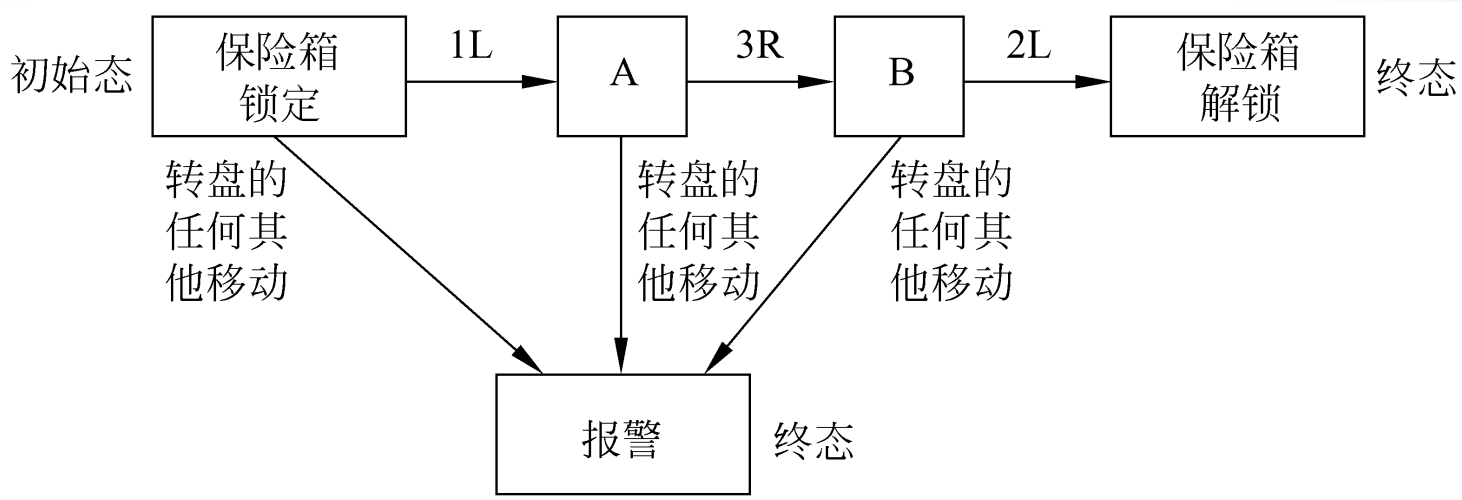
4.3 Petri网

4.4 Z语言

4.2 有穷状态机

4.2.1 概念

例子：一个保险箱上装了一个复合锁，锁有3个位置，分别标记为1、2、3，转盘可向左(L)或向右(R)转动。这样，在任意时刻转盘都有6种可能的运动，即1L、1R、2L、2R、3L和3R。保险箱的组合密码是1L、3R、2L，转盘的任何其他运动都将引起报警。



4.2 有穷状态机

一个初始态：保险箱锁定状态。
表4.1保险箱的状态装换表

当前状态 / 次态 / 转盘动作	保险箱锁定	A	B
1L	A	报警	报警
1R	报警	报警	报警
2L	报警	报警	保险箱解锁
2R	报警	报警	报警
3L	报警	报警	报警
3R	报警	B	报警

4.2 有穷状态机

从上面这个简单例子可以看出，一个有穷状态机包括下述5个部分：状态集J、输入集K、由当前状态和当前输入确定下一个状态(次态)的转换函数T、初始态S和终态集F。

对于保险箱的例子，相应的有穷状态机的各部分如下：

状态集J：｛保险箱锁定，A，B，保险箱解锁，报警｝。

输入集K：｛1L，1R，2L，2R，3L，3R｝。

转换函数T：如表所示。

初始态S：保险箱锁定。

终态集F：｛保险箱解锁，报警｝。

4.2 有穷状态机

如果使用更形式化的术语，一个有穷状态机可以表示为一个5元组(J，K，T，S，F)，其中：J是一个有穷的非空状态集；

K是一个有穷的非空输入集；

T是一个从(J-F)×K到J的转换函数；

S∈J，是一个初始状态；

F　J，是终态集。

4.2 有穷状态机

有穷状态机的概念在计算机系统中应用得非常广泛。

例如，每个菜单驱动的用户界面都是一个有穷状态机的实现。一个菜单的显示和一个状态相对应，键盘输入或用鼠标选择一个图标是使系统进入其他状态的一个事件。状态的每个转换都具有下面的形式：

当前状态〔菜单〕+事件〔所选择的项〕=>下个状态。

4.2 有穷状态机

为了对一个系统进行规格说明，通常都需要对有穷状态机做一个很有用的扩展，即在前述的5元组中加入第6个组件——谓词集P，从而把有穷状态机扩展为一个6元组，其中每个谓词都是系统全局状态Y的函数。转换函数T现在是一个从(J-F)×K×P到J的函数。现在的转换规则形式如下：

当前状态〔菜单〕+事件〔所选择的项〕+谓词=>下个状态。

4.2 有穷状态机

4.2.2 例子

有穷状态机技术表达系统的规格说明，现在给出电梯系统的规格说明。

用自然语言描述的对电梯系统的需求。

在一幢m层的大厦中需要一套控制n部电梯的产品，要求这n部电梯按照约束条件C1，C2和C3在楼层间移动。

4.2 有穷状态机

C1：每部电梯内有m个按钮，每个按钮代表一个楼层。当按下一个按钮时该按钮指示灯亮，同时电梯驶向相应的楼层，到达按钮指定的楼层时指示灯熄灭。

C2：除了大厦的最低层和最高层之外，每层楼都有两个按钮分别请求电梯上行和下行。这两个按钮之一被按下时相应的指示灯亮，当电梯到达此楼层时灯熄灭，电梯向要求的方向移动。

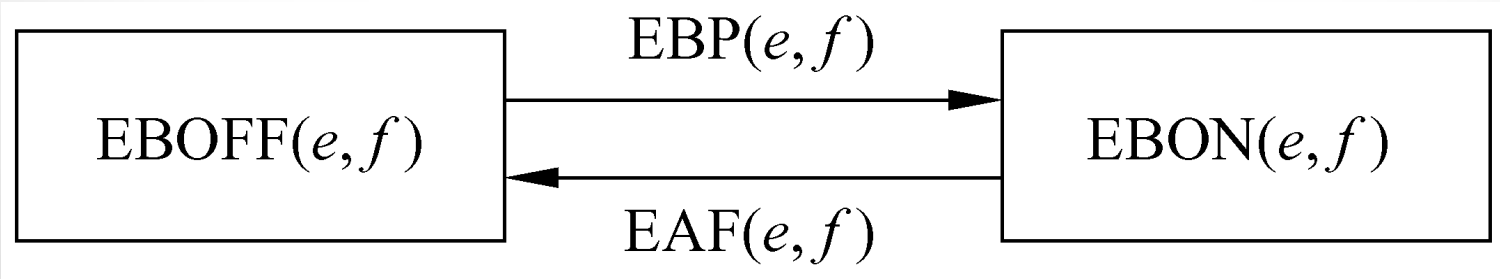
C3：当对电梯没有请求时，它关门并停在当前楼层。

4.2 有穷状态机

使用一个扩展的有穷状态机对本产品进行规格说明.

分析：问题中有两个按钮集。n部电梯中的每一部都有m个按钮，一个按钮对应一个楼层。因为这m×n个按钮都在电梯中，为电梯按钮。每层楼有两个按钮，一个请求向上，另一个请求向下，称为楼层按钮。

电梯按钮的状态转换图如图4.2所示。



4.2 有穷状态机

令EB(e,f)表示按下电梯e内的按钮并请求到f层去。EB(e,f)有两个状态，分别是按钮发光(打开)和不发光(关闭)。状态是：

EBON(e,f)：电梯按钮(e,f)打开

EBOFF(e,f)：电梯按钮(e,f)关闭

如果电梯按钮(e,f)发光且电梯到达f层，该按钮将熄灭。相反如果按钮熄灭，则按下它时，按钮将发光。两个事件，它们分别是：

EBP(e,f)：电梯按钮(e,f)被按下

EAF(e,f)：电梯e到达f层

4.2 有穷状态机

为了定义与这些事件和状态相联系的状态转换规则，需要一个谓词V(e,f)，它的含义如下：

V(e,f)：电梯e停在f层

如果电梯按钮(e,f)处于关闭状态〔当前状态〕，而且电梯按钮(e,f)被按下〔事件〕，而且电梯e不在f层〔谓词〕，则该电梯按钮打开发光〔下个状态〕。

4.2 有穷状态机

状态转换规则的形式化描述如下：

EBOFF(e,f)+EBP(e,f)+not V(e,f)=>EBON(e,f)

反之，如果电梯到达f层，而且电梯按钮是打开的，于是它就会熄灭。

这条转换规则可以形式化地表示为：

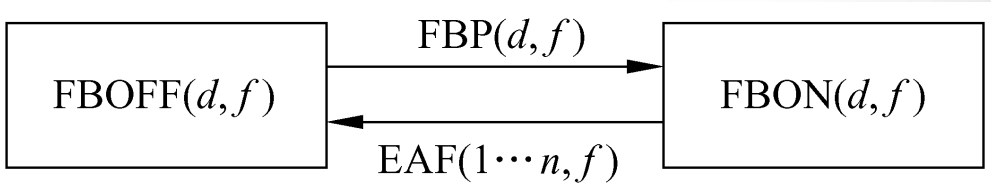
EBON(e,f)+EAF(e,f) =>EBOFF(e,f)

4.2 有穷状态机

考虑楼层按钮。令FB(d,f)表示f层请求电梯向d方向运动的按钮，楼层按钮FB(d,f)的状态转换图如图所示,楼层按钮的状态如下。

FBON(d,f)：楼层按钮(d,f)打开

FBOFF(d,f)：楼层按钮(d,f)关闭



楼层按钮已经打开，一部电梯到达f层则按钮关闭。

楼层按钮原来是关闭的，被按下后该按钮将打开。

FBP(d,f)：楼层按钮(d,f)被按下

EAF(1…n,f)：电梯1或…或n到达f层

其中，1…n表示或为1或为2…或为n。

4.2 有穷状态机

为了定义与这些事件和状态相联系的状态转换规则，需要一个谓词S(d,e,f)，定义如下：S(d,e,f)：电梯e停在f层并且移动方向由d确定为向上(d=U)或向下(d=D)或待定(d=N)。

这个谓词是一个状态，形式化方法允许把事件和状态作为谓词对待。

使用谓词S(d,e,f)，形式化转换规则为：

FBOFF(d,f)+FBP(d,f)+not S(d,1…n,f) =>FBON(d,f)

FBON(d,f)+EAF(1…n,f)+S(d,1…n,f) =>FBOFF(d,f)

其中，d=UorD。

4.2 有穷状态机

电梯按钮状态转换规则时定义的谓词V(e,f)，可以用谓词S(d,e,f)重新定义如下：**V(e,f)=S(U,e,f)or S(D,e,f)or S(N,e,f)**

电梯的状态及其转换规则，一个电梯状态实质上包含许多子状态(如，电梯减速、停止、开门、在一段时间后自动关门)。

下面定义电梯的3个状态。

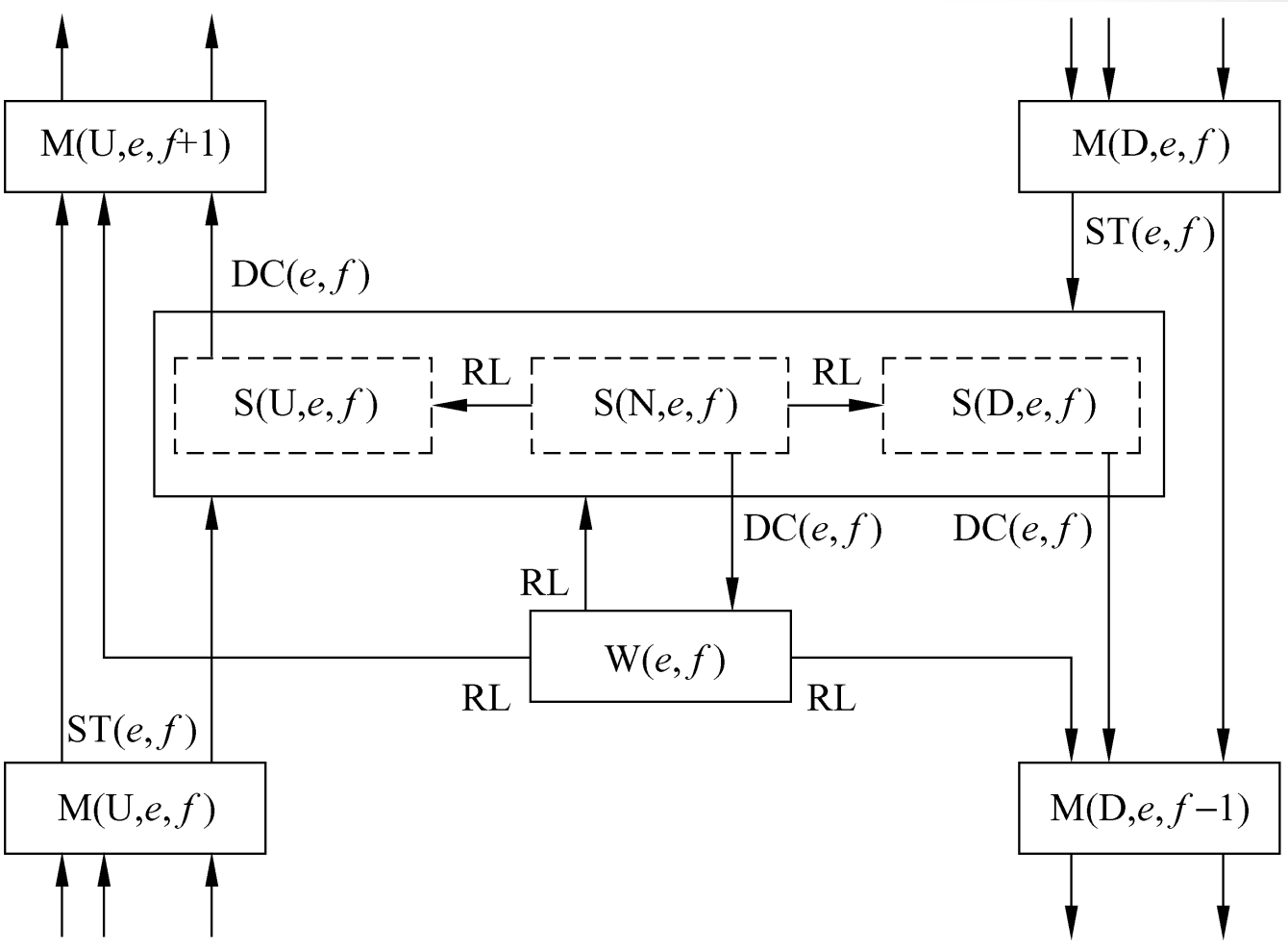
M(d,e,f)：电梯e正沿d方向移动，即将到达的是第f层

S(d,e,f)：电梯e停在f层，将朝d方向移动(尚未关门)

W(e,f)：电梯e在f层等待(已关门)其中，S(d,e,f)状态已在讨论楼层按钮时定义过，但是，现在的定义更完备。

4.2 有穷状态机

图是电梯的状态转换图。3个电梯停止状态S(U，e,f)、S(N，e,f)和S(D，e,f)已组合成一个大的状态，目的是减少状态总数以简化流图。



4.2 有穷状态机

图4.4中包含了下述3个可触发状态发生改变的事件。

DC(e,f)：电梯e在楼层f关上门

ST(e,f)：电梯e靠近f层时触发传感器，电梯控制器决定在当前楼层电梯是否停下

RL：电梯按钮或楼层按钮被按下进入打开状态，登录需求

4.2 有穷状态机

电梯的状态转换规则。这里给出的规则仅发生在关门之时。

S(U，e,f)+DC(e,f) => M(U，e,f+1)

S(D，e,f)+DC(e,f) => M(D，e,f-1)

S(N，e,f)+DC(e,f) =>W(e,f)

第一条规则表明，如果电梯e停在f层准备向上移动，且门已经关闭，则电梯将向上一楼层移动。

第二条和第三条规则，分别对应于电梯即将下降或者没有待处理的请求的情况。

4.2 有穷状态机

4.2.3 评价

有穷状态机方法采用了一种简单的格式来描述规格说明：

当前状态+事件+谓词=>下个状态

这种形式的规格说明易于书写、易于验证，而且可以比较容易地把它转变成设计或程序代码。事实上，可以开发一个CASE工具把一个有穷状态机规格说明直接转变为源代码。

4.2 有穷状态机

有穷状态机方法比数据流图技术更精确，而且和它一样易于理解。

它也有缺点：在开发一个大系统时三元组(即状态、事件、谓词)的数量会迅速增长。此外，和数据流图方法一样，形式化的有穷状态机方法也没有处理定时需求。

主要内容

4.1 概述

4.2 有穷状态机

4.3 Petri网

4.4 Z语言

4.3 Petri网

4.3.1 概念

Petri网由来：

并发系统中遇到的一个主要问题是定时问题。这个问题可以表现为多种形式，如同步问题、竞争条件以及死锁问题。

用于确定系统中隐含的定时问题的一种有效技术是Petri网，这种技术的一个很大的优点是它也可以用于设计中。

Petri网是由*Carl Adam Petri*发明的。在性能评价、操作系统和软件工程等领域，Petri网应用得都比较广泛。特别是已经证明，用Petri网可以有效地描述并发活动。

4.3 Petri网

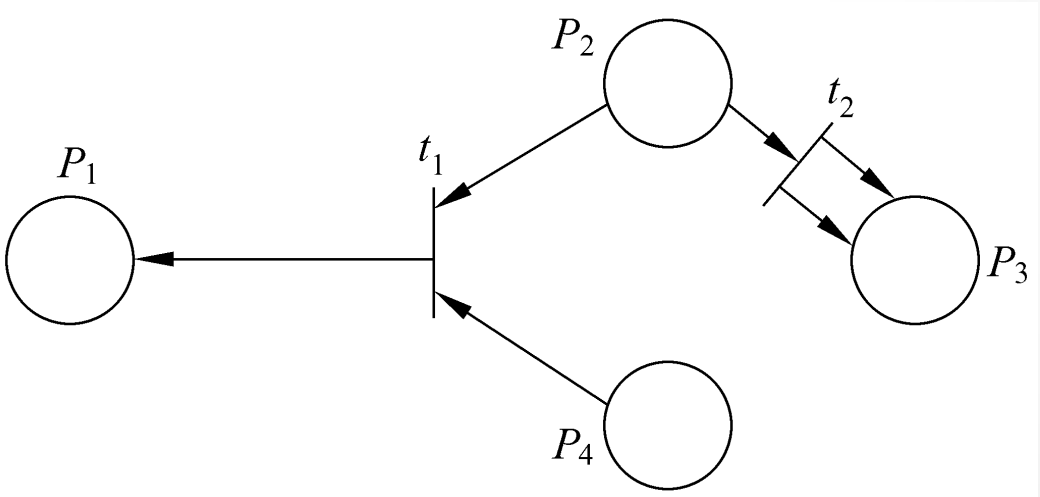
Petri网包含4种元素：

一组位置P、

一组转换T、

输入函数I、

输出函数O。



其中：

一组位置P为｛P1，P2，P3，P4｝，在图中用圆圈代表位置。

一组转换T为｛t1，t2｝，在图中用短直线表示转换。

4. 3 Petri网

两个用于转换的输入函数，用由位置指向转换的箭头表示，是：

$$I(t_1)=\{P_2，P_4\}$$

$$I(t_2)=\{P_2\}$$

两个用于转换的输出函数，用由转换指向位置的箭头表示，是：

$$O(t_1)=\{P_1\}$$

$$O(t_2)=\{P_3，P_3\}$$

注意，输出函数$O(t_2)$中有两个P_3，是因为有两个箭头由t_2指向P_3。

4. 3 Petri网

更形式化的Petri网结构，是一个四元组C=(P,T,I,O)。

其中：P=｛P1，…，Pn｝是一个有穷位置集，n≥0。

T=｛t1，…，tm｝是一个有穷转换集，m≥0，且T和P不相交。

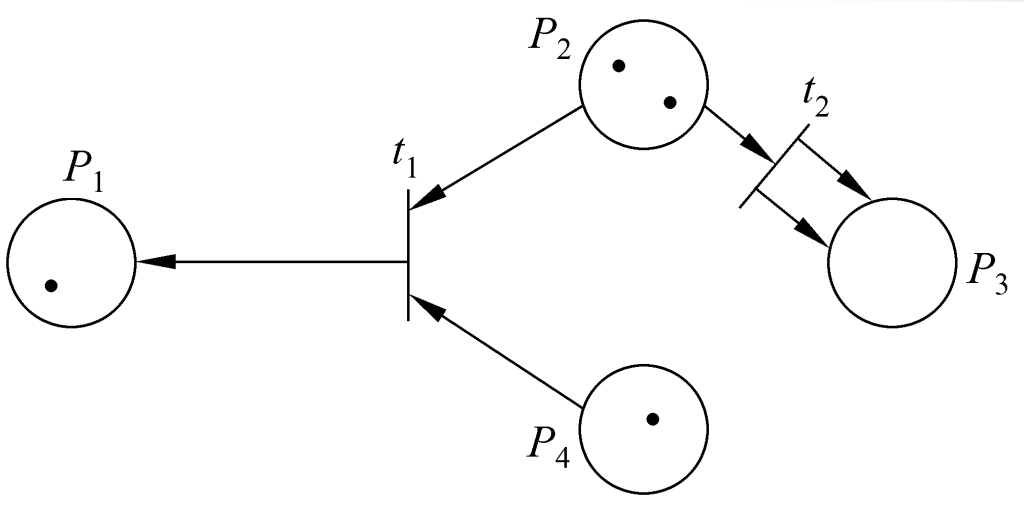
I：T→P∞为输入函数，是由转换到位置无序单位组(bags)的映射。

O：T→P∞为输出函数，是由转换到位置无序单位组的映射。

4. 3 Petri网

一个无序单位组或多重组是允许一个元素有多个实例的广义集。

Petri网的标记是在Petri网中权标(token)的分配。例如，在图4.6中有4个权标，其中一个在P_1中，两个在P_2中，P_3中没有，还有一个在P_4中。上述标记可以用向量(1，2，0，1)表示。由于P_2和P_4中有权标，因此t_1启动(即被激发)。



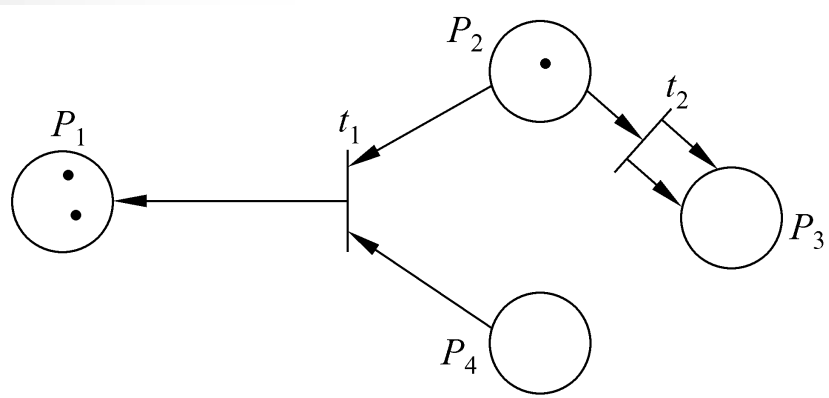
4.3 Petri网

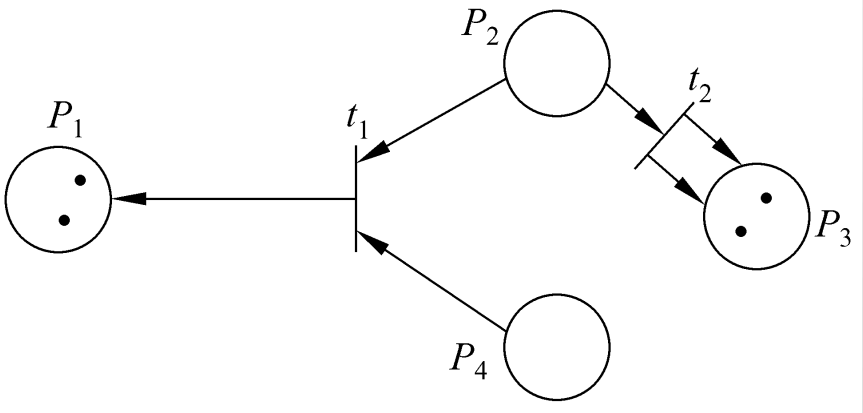
通常，当每个输入位置所拥有的权标数大于等于从该位置到转换的线数时，就允许转换。当t_1被激发时，P_2和P_4上各有一个权标被移出，而P_1上则增加一个权标。Petri网中权标总数不是固定的，在这个例子中两个权标被移出，而P_1上只能增加一个权标。

在图4.6中P_2上有权标，因此t_2也可以被激发。当t_2被激发时，P_2上将移走一个权标，而P_3上新增加两个权标。Petri网具有非确定性，也就是说，如果数个转换都达到了激发条件，则其中任意一个都可以被激发。

4. 3 Petri网

左图所示Petri网的标记为(1，2，0，1)，t_1和t_2都可以被激发。假设t_1被激发了，则结果如左图所示，标记为(2，1，0，0)。此时，只有t_2可以被激发。如果t_2也被激发了，则权标从P_2中移出，两个新权标被放在P_3上，结果如右图所示，标记为(2，0，2，0)。





4.3 Petri网

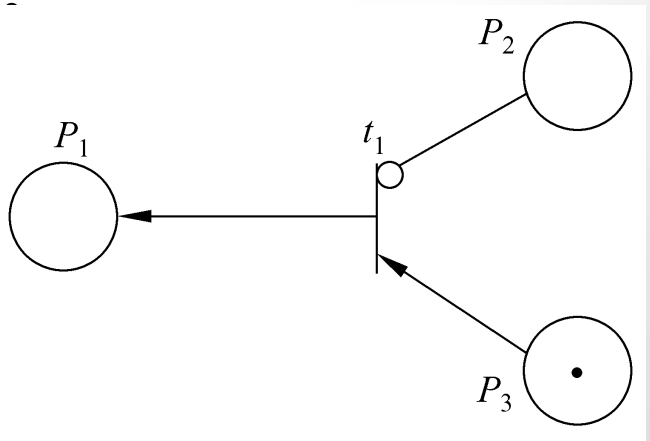
更形式化地说，Petri网C=(P，T，I，O)中的标记M，是由一组位置P到一组非负整数的映射：

M：P→{0，1，2，…}

带有标记的Petri网成为一个五元组(P，T，I，O，M)。

对Petri网的一个重要扩充是加入禁止线。

如图所示，禁止线是用一个小圆圈而不是用箭头标记的输入线。



4.3 Petri网

4.3.2 例子

现在把Petri网应用于上一节讨论过的电梯问题。当用Petri网表示电梯系统的规格说明时，每个楼层用一个位置F_f代表(1≤f≤m)，在Petri网中电梯是用一个权标代表的。在位置F_f上有权标，表示在楼层f上有电梯。

4. 3 Petri网

1. 电梯按钮

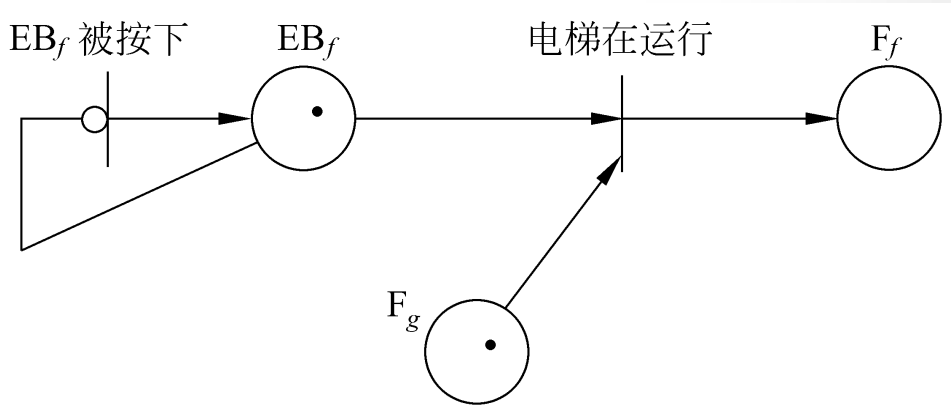
电梯问题的第一个约束条件描述了电梯按钮的行为。

第一条约束C1：每部电梯有m个按钮，每层对应一个按钮。当按下一个按钮时该按钮指示灯亮，指示电梯移往相应的楼层。当电梯到达指定的楼层时，按钮将熄灭。

为了用Petri网表达电梯按钮的规格说明，在Petri网中还必须设置其他的位置。电梯中楼层f的按钮，在Petri网中用位置EB_f表示($1 \le f \le m$)。在EB_f上有一个权标，就表示电梯内楼层f的按钮被按下了。

4. 3 Petri网

电梯按钮只有在第一次被按下时才会由暗变亮，以后再按它则只会被忽略。如图所示的Petri网描述了电梯按钮的行为规律。首先，假设按钮没有发亮，在位置EB_f上没有权标，在存在禁止线的情况下，转换“EB_f被按下”是允许发生的。现在按下按钮，则转换被激发并在EB_f上放置了一个权标，如图所示。因此，位置EB_f上的权标数不会多于1。



4.3 Petri网

假设电梯由g层驶向f层，因为电梯在g层，如上图所示，位置Fg上有一个权标。由于每条输入线上各有一个权标，转换“电梯在运行”被激发，从而EBf和Fg上的权标被移走，按钮EBf被关闭，在位置Ff上出现一个新权标，即转换的激发使电梯由g层驶到f层。

事实上，电梯由g层移到f层是需要时间的，为处理这个情况及其他类似的问题，Petri网模型中必须加入时限。也就是说，在标准Petri网中转换是瞬时完成的，而在现实情况下就需要时间控制Petri网，以使转换与非零时间相联系。

4. 3 Petri网

2. 楼层按钮

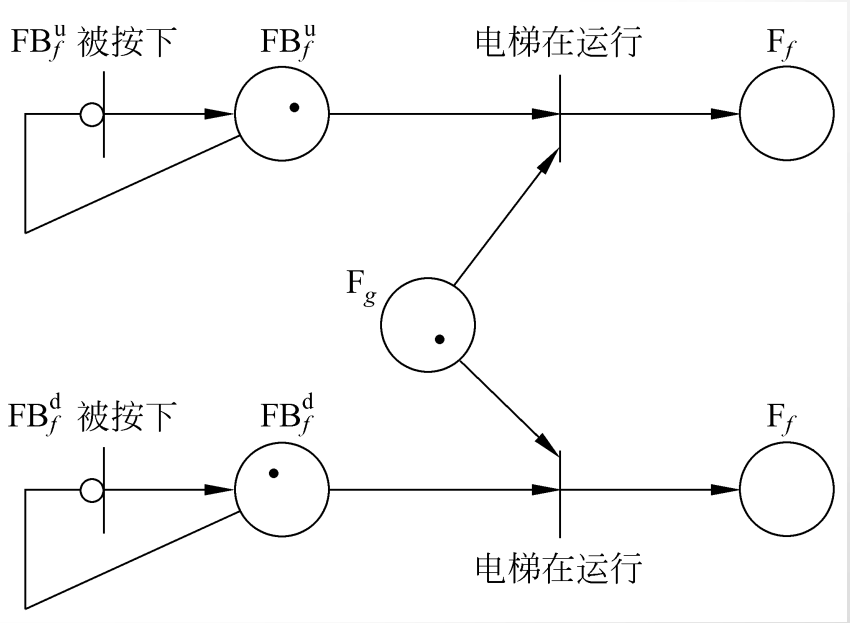
在第二个约束条件中描述了楼层按钮的行为。

第二条约束C2：除了第一层与顶层之外，每个楼层都有两个按钮，一个要求电梯上行，另一个要求电梯下行。这些按钮在按下时发亮，当电梯到达该层并将向指定方向移动时，相应的按钮才会熄灭。

在Petri网中楼层按钮用位置FB_f^u和FB_f^d表示，分别代表f楼层请求电梯上行和下行的按钮。底层的按钮为FB_1^u，最高层的按钮为FB_m^d，中间每一层有两个按钮FB_f^u和FB_f^d (1＜f＜m)。

4.3 Petri网

图所示的情况为电梯由g层驶向f层。根据电梯乘客的要求，某一个楼层按钮亮或两个楼层按钮都亮。如果两个按钮都亮了，则只有一个按钮熄灭。图所示的Petri网可以保证，当两个按钮都亮了的时候，只有一个按钮熄灭。但是要保证按钮熄灭正确，则需要更复杂的Petri网模型，对此不做更进一步的介绍。



4. 3 Petri网

第三条约束C3：当电梯没有收到请求时，它将停留在当前楼层并关门。

这条约束很容易实现，如图4.11所示，当没有请求(FB_f^u和FB_f^d上无权标）时，任何一个转换“电梯在运行”都不能被激发。

主要内容

4.1 概述

4.2 有穷状态机

4.3 Petri网

4.4 Z语言

4.4 Z语言

4.4.1 简介

用Z语言描述的、最简单的形式化规格说明含有下述4个部分。

- 给定的集合、数据类型及常数。
- 状态定义。
- 初始状态。
- 操作。

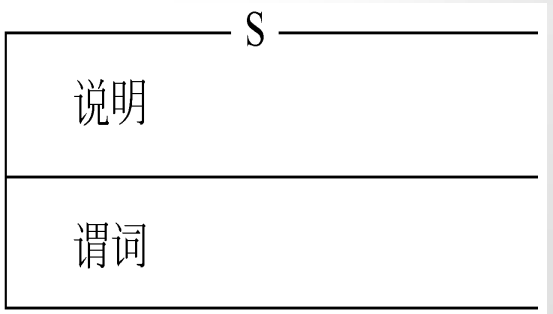
4.4 Z语言

1. 给定的集合

一个Z规格说明从一系列给定的初始化集合开始。所谓初始化集合就是不需要详细定义的集合，这种集合用带方括号的形式表示。对于电梯问题，给定的初始化集合称为Button，即所有按钮的集合，因此，Z规格说明开始于：〔Button〕

2. 状态定义

一个Z规格说明由若干个“格(schema)”组成，每个格含有一组变量说明和一系列限定变量取值范围的谓词。例如，格S的格式如图所示。



4.4 Z语言

在电梯问题中，Button有4个子集，即floor_buttons(楼层按钮的集合)、elevator_buttons(电梯按钮的集合)、buttons(电梯问题中所有按钮的集合)以及pushed(所有被按的按钮的集合，即所有处于打开状态的按钮的集合)。

描述了格Button_State，符号P表示幂集(即给定集的所有子集)图。约束条件声明，floor_buttons集与elevator_buttons集不相交，而且它们共同组成buttons集

Button_State

$$\begin{aligned} &floor\_buttons, elevator\_buttons : \mathbb{P}\ Button \\ &buttons : \mathbb{P}\ Button \\ &pushed : \mathbb{P}\ Button \end{aligned}$$

$$floor\_buttons \cap elevator\_buttons = \Phi$$

$$floor\_buttons \cup elevator\_buttons = buttons$$

4.4 Z语言

3. 初始状态

抽象的初始状态是指系统第一次开启时的状态。

对于电梯问题来说，抽象的初始状态为：

$$Button\_Init \triangleq [Button\_State \mid pushed = \Phi]$$

上式表示，当系统首次开启时pushed集为空，即所有按钮都处于关闭状态。

4.4 Z语言

4. 操作

如果一个原来处于关闭状态的按钮被按下，则该按钮开启，这个按钮就被添加到pushed集中。图定义了操作Push_Button(按按钮)。

Push_Button

ΔButton_State

button?:Button

$$
\begin{aligned}
&(button? \in buttons) \land \\
&(((button? \notin pushed) \land (pushed' = pushed \cup \{button?\})) \lor \\
&((button? \in pushed) \land (pushed' = pushed)))
\end{aligned}
$$

4.4 Z语言

操作的谓词部分，包含了一组调用操作的前置条件，以及操作完全结束后的后置条件。

图4.14中的第一个前置条件规定，“button?”必须是buttons的一个元素，而buttons是电梯系统中所有按钮的集合。如果第二个前置条件button? ∉pushed得到满足(即按钮没有开启)，则更新pushed按钮集，使之包含刚开启的按钮“button?”。Z语言中，当一个变量的值发生改变时，就用符号“′”表示。后置条件是当执行完操作Push_Button之后，“button?”将被加入到pushed集中。无须打开按钮，使“button?”变成pushed中的一个元素即可。

4.4 Z语言

另一种可能性是，被按的按钮原先已经打开了。由于button?∈pushed，根据第3个前置条件，将没有任何事情发生，这可以用pushed′=pushed来表示，即pushed的新状态和旧状态一样。注意，如果没有第3个前置条件，规格说明将不能说明在一个按钮已被按过之后又被按了一次的情况下将发生什么事，因此，结果将是不可预测的。

4.4 Z语言

电梯到达了某楼层，如果相应的楼层按钮已经打开，则此时会关闭；同样，如果相应的电梯按钮已经打开，则此时它也会关闭。也就是说，如果“button?”属于pushed集，则将它移出该集合，如图所示(符号\表示集合差运算)。但是，如果按钮“button?”原先没有打开，则pushed集合不发生变化。

Floor_Arrival

$\triangle$Button_State

button? : Button

$(button? \in buttons) \land$
$((button? \in pushed) \land (pushed' = pushed \setminus \{button?\})) \lor$
$((button? \notin pushed) \land (pushed' = pushed))$

4.4 Z语言

4.4.2 评价

Z也许是应用得最广泛的形式化语言，

(1) 可以比较容易地发现用Z写的规格说明的错误，特别是在自己审查规格说明，及根据形式化的规格说明来审查设计与代码时，情况更是如此。

(2) 用Z写规格说明时，要求作者十分精确地使用Z说明符

(3) Z是一种形式化语言，在需要时开发者可以严格地验证规格说明的正确性。

4.4 Z语言

(4) 虽然完全学会Z语言相当困难，但是，经验表明，只学过中学数学的软件开发人员仍然可以只用比较短的时间就学会编写Z规格说明，当然，这些人还没有能力证明规格说明的结果是否正确。

(5) 使用Z语言可以降低软件开发费用。

(6) 虽然用户无法理解用Z写的规格说明，但是，可以依据Z规格说明用自然语言重写规格说明。经验证明，这样得到的自然语言规格说明，比直接用自然语言写出的非形式化规格说明更清楚、更正确。

4.4 本章小结

1. 形式化技术有优点也有缺点
2. 介绍了有穷状态机的概念，并举例并评价
3.简要介绍了Petri网的概念，并举例并评价
4.简要介绍了Z语言的概念及评价

本章结束