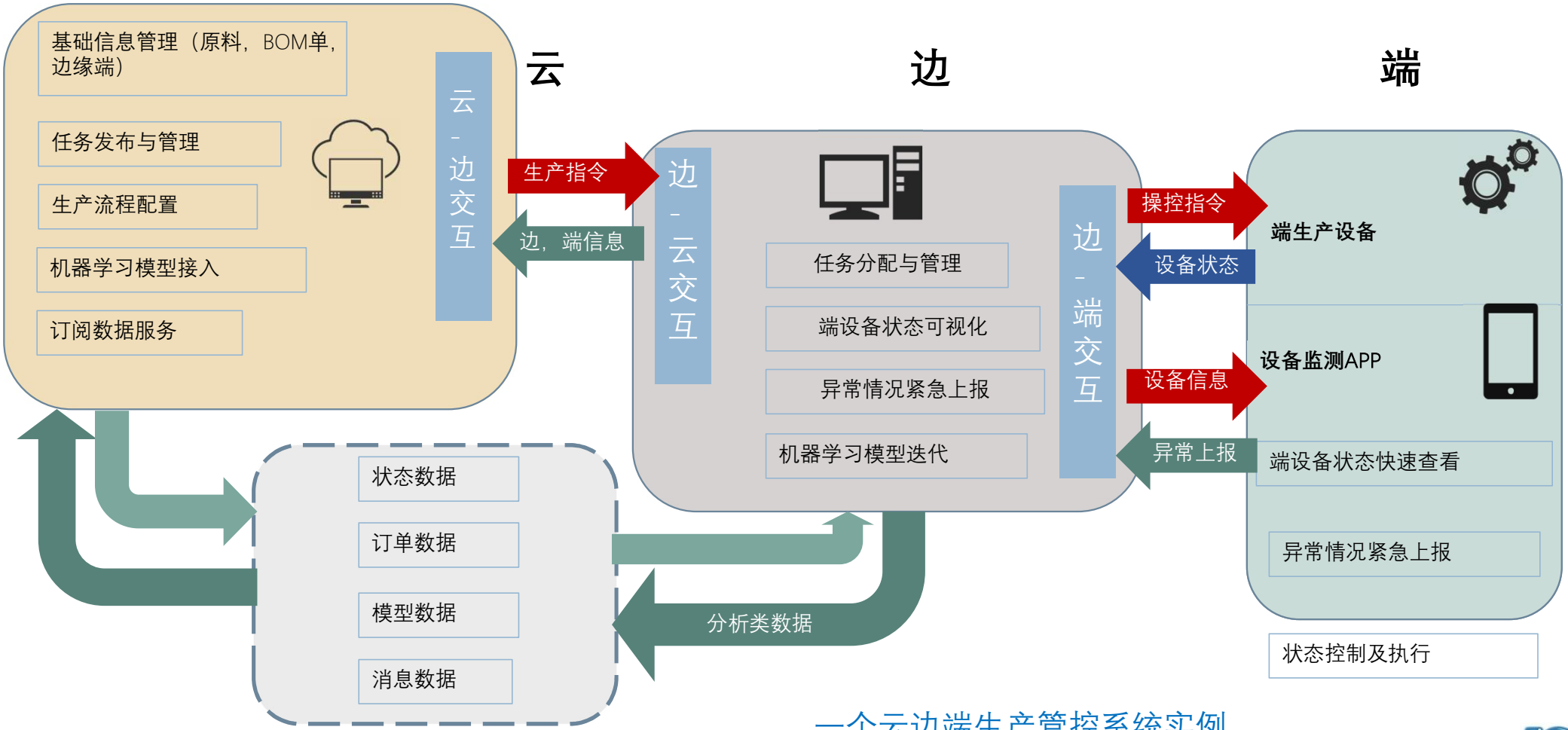
目录

4 基于状态的过程建模方法

状态精确描述了系统的条件。基于状态的建模能精确地描述系统的特点和变化方式。



一个云边端生产管控系统实例

4.1 离散事件动态系统DEDS

- **离散事件动态系统（Discrete Event Dynamic System, DEDS）是由异步、突发的事件驱动状态演化的动态系统。**
  - 这种系统的状态通常只取有限个离散值，对应于系统部件的好坏、忙闲及待处理工件个数等可能的物理状况，或计划制定、作业调度等宏观管理的状况。
  - 而这些状态的变化则由于诸如某些环境条件的出现或消失、系统操作的启动或完成等各种事件的发生而引起。
- **主要涉及到：**
  - **离散事件动态系统**DEDS
  - **经典**Petri**网以及高阶**Petri**网**
  - **基于**PNG**的流程建模及仿真**

DEDS系统特点

- **动态性。**系统具有动态性，其动态行为以及转换过程对于应用较为重要。
- **复杂性：**系统的推演过程是由事件驱动而变化的，其过程带有不连续性；系统的性能指标却带有连续特点，例如平均吞吐率等。
- **随机性。**因为有些驱动系统的事件到来带有随机特点，此外任务的完成、任务间的衔接等也都有一定的随机性，所以常常需要使用处理概率事件与随机过程的方法和技术。
- **层次性。**由于这类系统一般是人造复杂系统，所以多半是按层次组织的，因此，常常涉及到各种因素的分解及集成，上下层次之间还必须协调以保持一致性。
- **计算复杂性。**系统的组成单元数目大，事件状态多，所以常有“组合爆炸”的危险，这给分析计算带来了很大困难。

- 离散事件动态系统的系统状态通常只取有限个离散值，是由异步、突发的事件驱动状态演化的动态系统。
- 常见于通信、交通等公共服务设施，机械、电子等各种离散型生产加工过程，多级控制系统，计算机信息处理等重要技术领域。
- 离散事件动态系统的研究自20世纪80年代后发展较快。针对各层次不同角度的问题提出了多种理论模型和分析技术，并被认为是大型复杂信息处理和控制系统分析和设计的重要理论基础，已开始在许多技术领域得到应用。
- 由于其状态空间缺乏易操作的运算结构，难以用传统基于微分或差分方程的方法来研究。对这种系统首先关心的是它的逻辑行为，可用其演化过程的状态序列和事件序列来刻画。

DEDS系统的理论模型

- 离散事件动态系统研究的最基本问题仍是系统动态建模，当前公认的理论框架包含以下三种模型。
  - (1) 逻辑层次模型：只涉及物理状态和事件之间的关系，属于确定性模型，主要包括形式语言/有限自动机和Petri网，用于定性分析。近年的动向是在确定模型中引入随机因素和时间因素，其中计时Petri网和随机Petri网比较重要；
  - (2) 时间层次模型：不仅涉及事件和状态之间的关系，而且要在物理的时间级上刻画与分析演化过程，主要方法为双子代数，网络演算属于这一层次；
  - (3) 统计性能层次模型：起源于对随机服务系统的研究，主要方法是排队论和排队网络，理论分析的基础是过程的马尔可夫性。
- 而Petri网具有严格的数学基础和规范化语义，可描述异步、同步、并行的逻辑关系，是描述、分析和控制DEDS的最有效和应用最广泛的方法；

4.2 经典Petri网理论

- 经典的Petri网是由德国 Carl Adam Petri在 1962年的博士论文 < Kommunikation mit Automaten >中提出的。
- 他发现当时已有理论，如有限自动状态机FSM和形式化语言都不适合描述物理系统的真实状态（并行、冲突等），于是就提出了Petri网。
- Petri网发展也经历了三个阶段:
  - 最先集中在孤立网系统研究；
  - 1970年至1985年，开始通用网系统研究，但主要被用于理论界；
  - 自从80年中期后，实际的应用越来越多，这主要是由于引入高阶 Petri网和许多工具；

Petri网主要用途

主要用途包括：

- 系统仿真：系统分析与评估的系统仿真。
- 数字分析：可通过结构变化描述系统的变化，支持DEDS形式的数学描述与分析；
- 系统性能分析：如制造系统设备使用率、生产率、可靠性等。
- 系统控制：直接从可视化模型中产生DEDS监控编码，进行系统实施控制。
- 还可以转化为其它的DEDS模型，如马可夫链等。

Petri网在计算机软件中可以应用到：

- 工作流建模及管理
- 并行程序设计
- 协议验证
- 运行状态的数据分析
- 软件逻辑设计（MDA，LCD，RPA等技术）
- …

Petri网的主要特点

- Petri网的主要特点：
  - 从控制和管理的角度模拟系统，不涉及系统所依赖的具体专业原理，简化某些细节，易于理解。
  - 精确描述系统中有关条件的依赖关系和不依赖关系
  - 具有统一的语言描述系统结构和行为，方便建模及仿真
  - 比其他图形建模工具更适于描述并发和冲突。

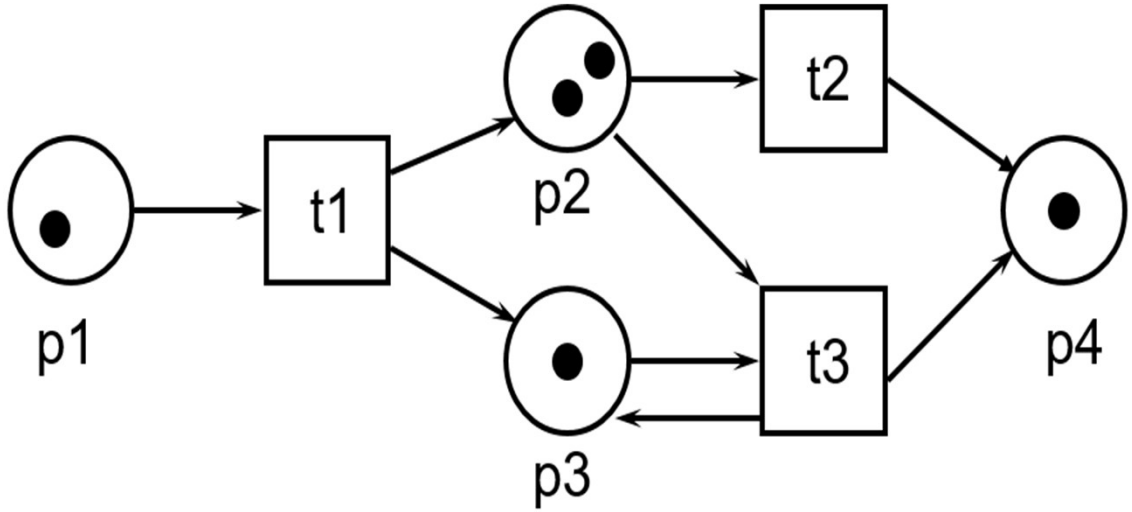
从建模角度来说

- Petri网是从状态及变化过程出发，为复杂系统的描述与分析设计提供的一种有效的建模工具，能自然的描述并发、冲突、资源争用等系统特性；
- 并带有执行控制机制，同时还具备形式化步骤及图论支持的理论严密性。
- Petri网的图形表达的直观性和便于编程实现的技术特点，使得它已经成为目前工作流流建模的主要工具之一。
- 从建模角度——可视化图形描述却被形式化数学方法支持。

- 研究领域趋向认为Petri网是所有流程定义语言之母。

经典Petri网构成

- 经典Petri网的**结构元素**：
  - 库所（Place）圆形节点
  - 转移（Transition）方形节点
  - 连接（Connection）是库所和转移之间的有向边，具有方向，用有向弧表示。
  - 托肯（Token）是库所中的动态对象，可以从一个库所移动到另一个库所，用实心小圆点表示。



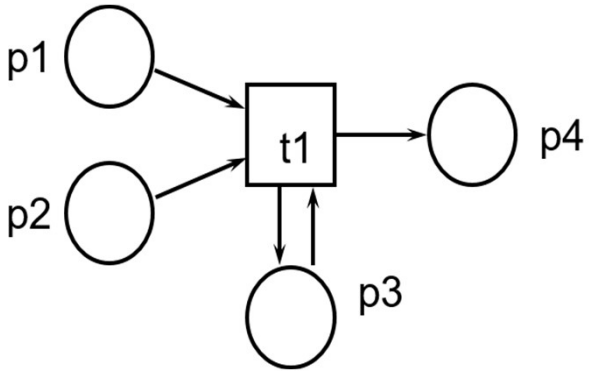
经典Petri网理论

- Petri网的**规则**：
  - 连接是有方向的，其上可以标出权重；
  - 两个库所或转移之间不允许有边，且不应该有孤立节点；
  - 库所可以拥有任意数量的托肯；
- Petri网的描述定义：
  - **输入库所**：以转移为基础，连接到转移的库所为该转移的输入库所，对应该转移的前条件。
  - **输出库所**：以转移为基础，转移连接出的库所为该转移的输出库所，对应该转移的后条件。

经典Petri网理论

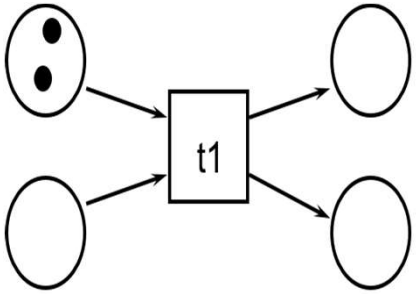
转移t1的输入输出库所

- t1具有三个输入库所 (p1，p2 , p3) 和两个输出库所 (p3 , p4)；库所p3既是t1的输入库所又是它的输出库所

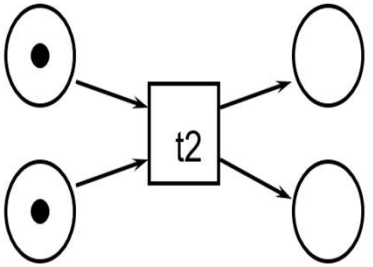


转移的使能条件

使能条件为:如果输入库所都包含了托肯，那么转移就被激活



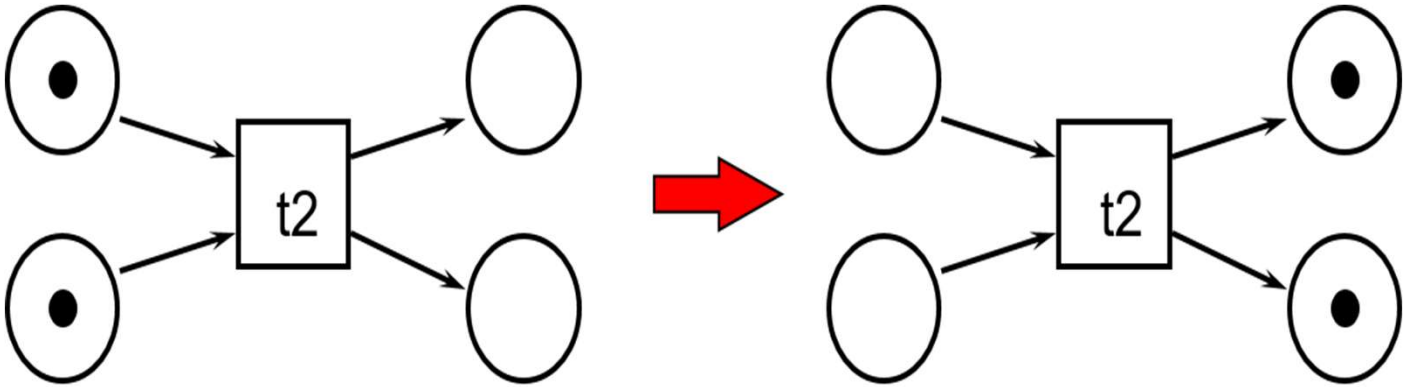
Transition t1 is not enabled,



transition t2 is enabled.

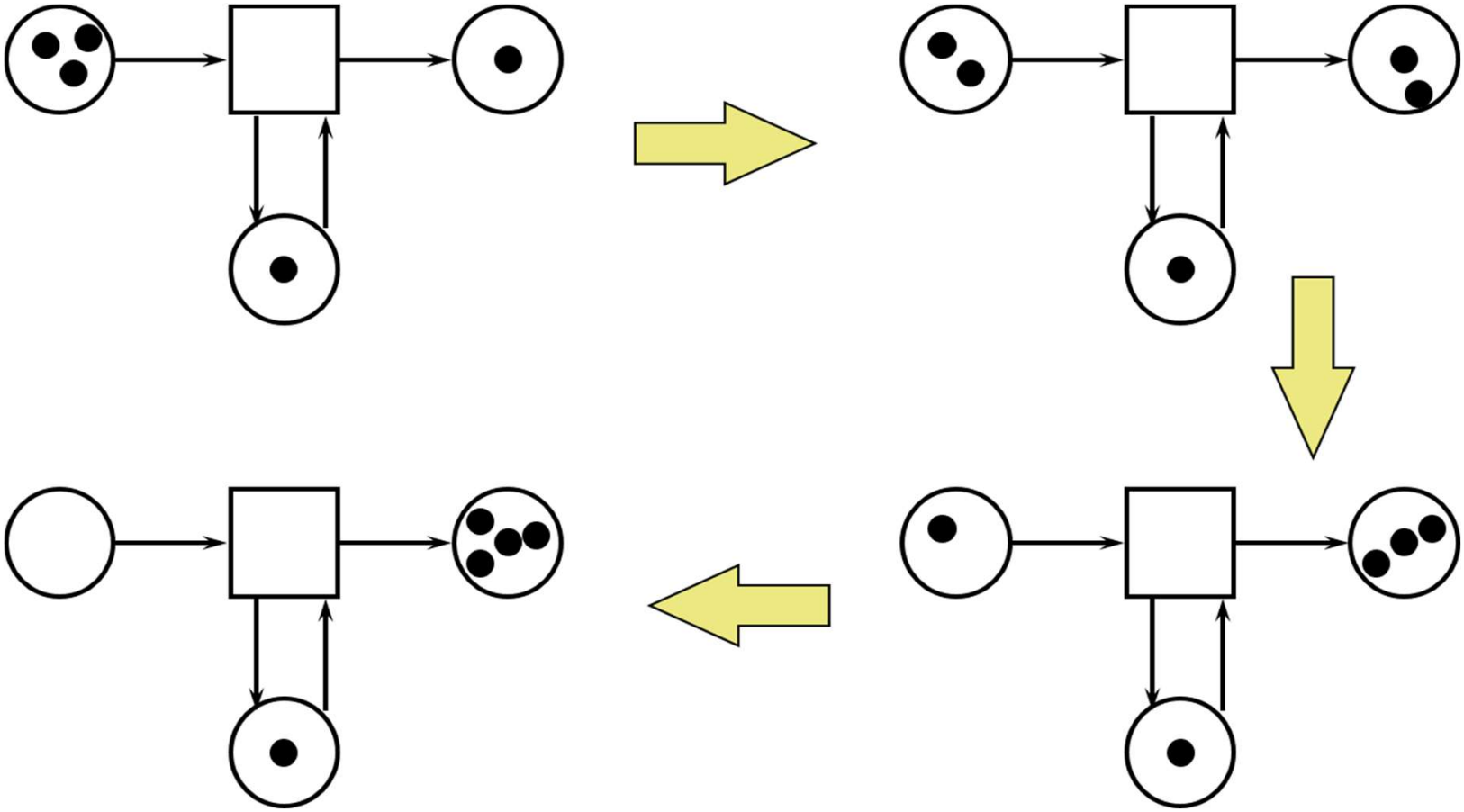
- 转移的激活叫做点火

点火将消耗输入库所的托肯，并为输出库所产生托肯



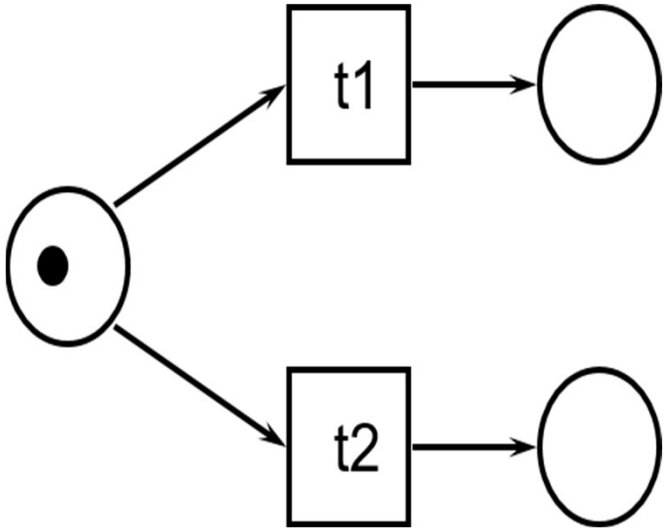
自动点火

一个托肯迁移的例子



经典Petri网理论

- Petri网中，托肯的点火具有的冲突与不确定性，两个转移竞争同一个托肯，产生冲突；
- 即使有两个或者多个托肯，依然存在冲突。



Petri网的行为规则

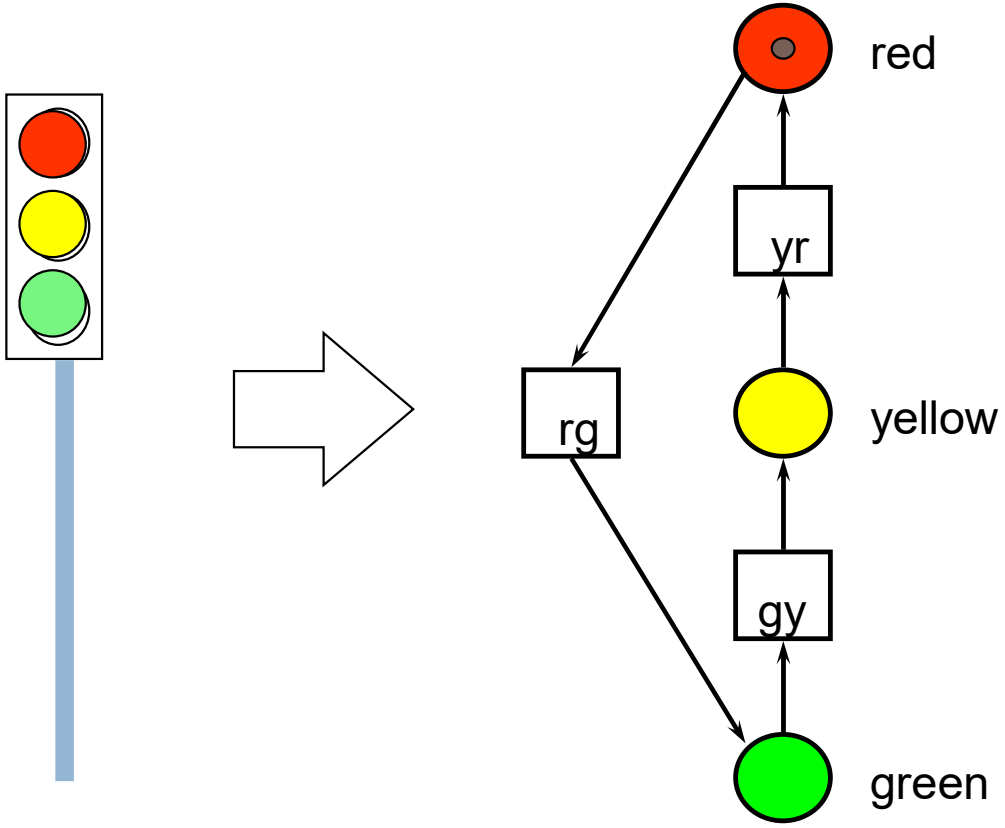
- 一个转移点火的规则是，如果一个转移的每个输入库所都拥有托肯，该转移即为被允许(enable)。点火时，输入库所的托肯被消耗，同时为输出库所产生托肯。
- 转移的发生必须是完整的，也就是说，没有转移只发生了一半的可能性。
- 一个转移出现其输入库所的个数与输出库所的个数不相等时，托肯的个数将发生变化，也就是说，托肯数目不守恒。
- 如网络中有两个或多个转移都被允许的可能，这种情况下转移发生的顺序没有定义，可认为是并发的。
- 两个转移争夺一个托肯的情形被称之为冲突。当发生冲突的时候，由于Petri网的时序是不确定的，因此具体哪个转移得以发生也是不确定的。
- Petri网络是静态的，也就是说，网络拓补结构是静态的，不存在发生了一个转移之后忽然冒出另一个转移或者库所，从而改变Petri网结构的可能。
- Petri网的状态由托肯在库所的分布决定。也就是说，转移发生完毕、下一个转移等待发生的时候才有确定的状态，正在发生转移的时候是没有一个确定的状态的。

基于Petri网的流程建模

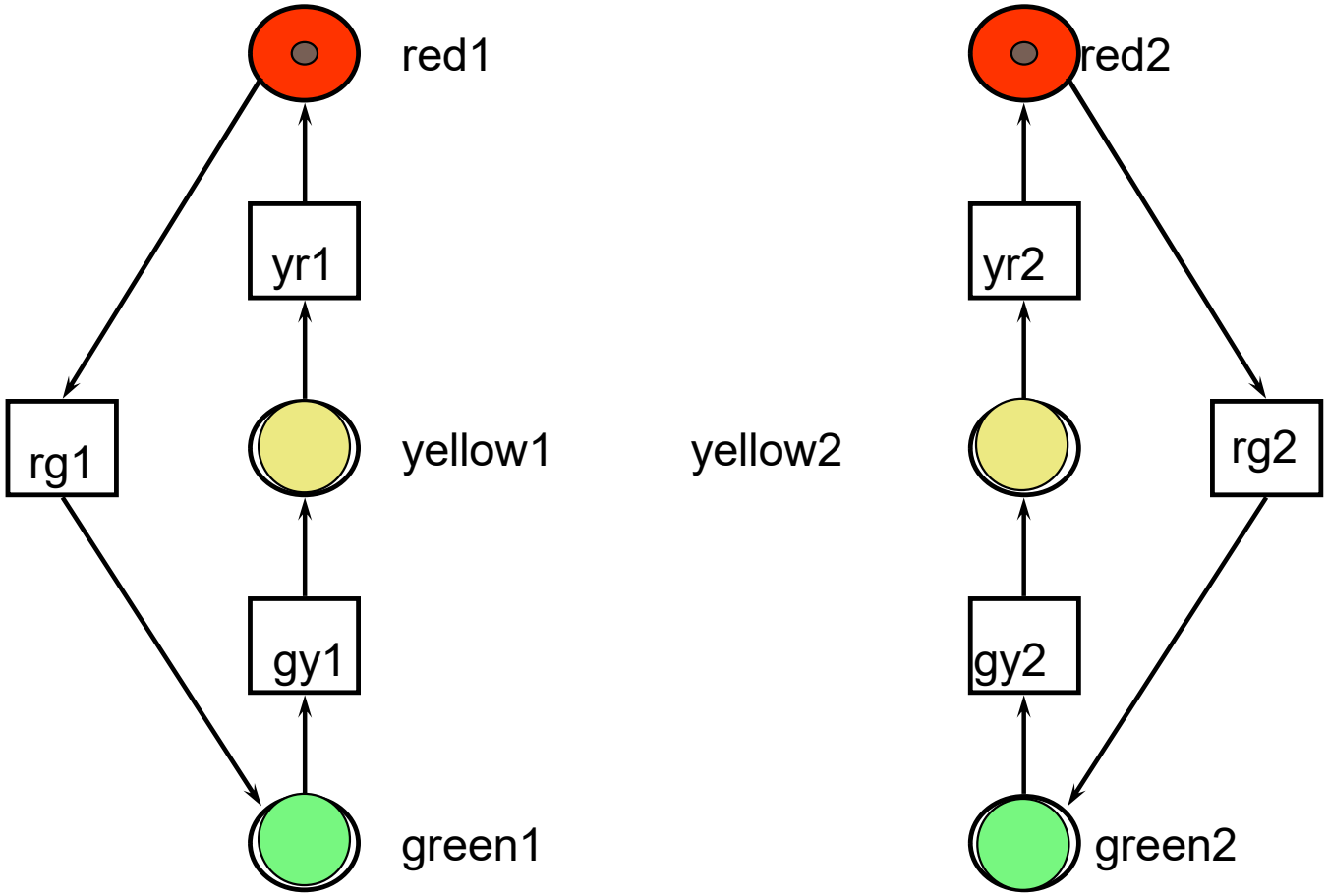
- 库所代表条件，缓存，渠道，地理位置，或者状态
- 转移代表活动、动作，传输或者转换
- 托肯表示对象 (humans, goods, machines), 信息或者对象的状态
- 过程的状态用位于库所的托肯来表示，状态之间的变换用转移来表示

- 讨论：一个航空公司的运营系统如何建模？飞机，航线，机场（位置），如何表示？

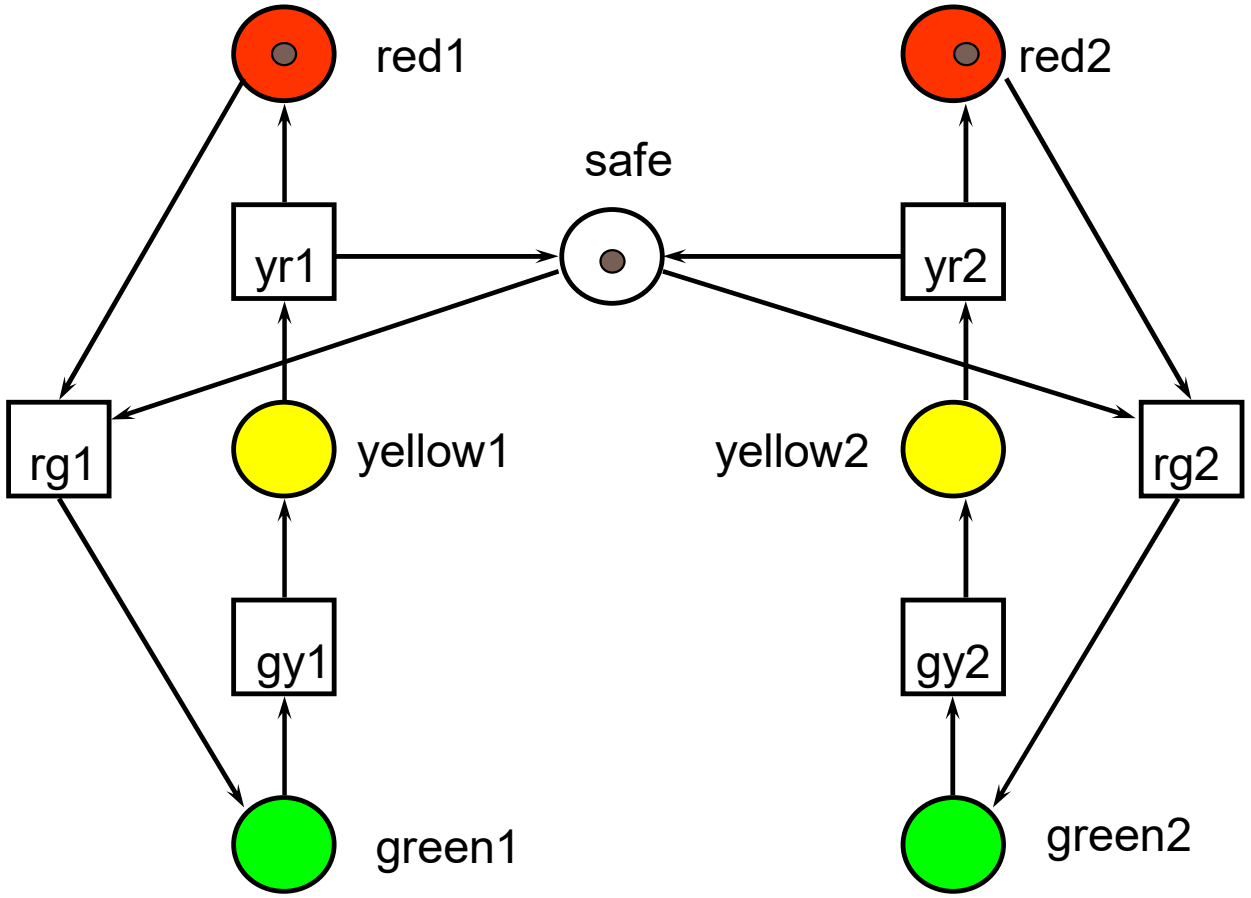
交通灯例子



两个交通灯


red1
yr1
rg1
yellow1
gy1
green1
yellow2
red2
yr2
rg2
gy2
green2

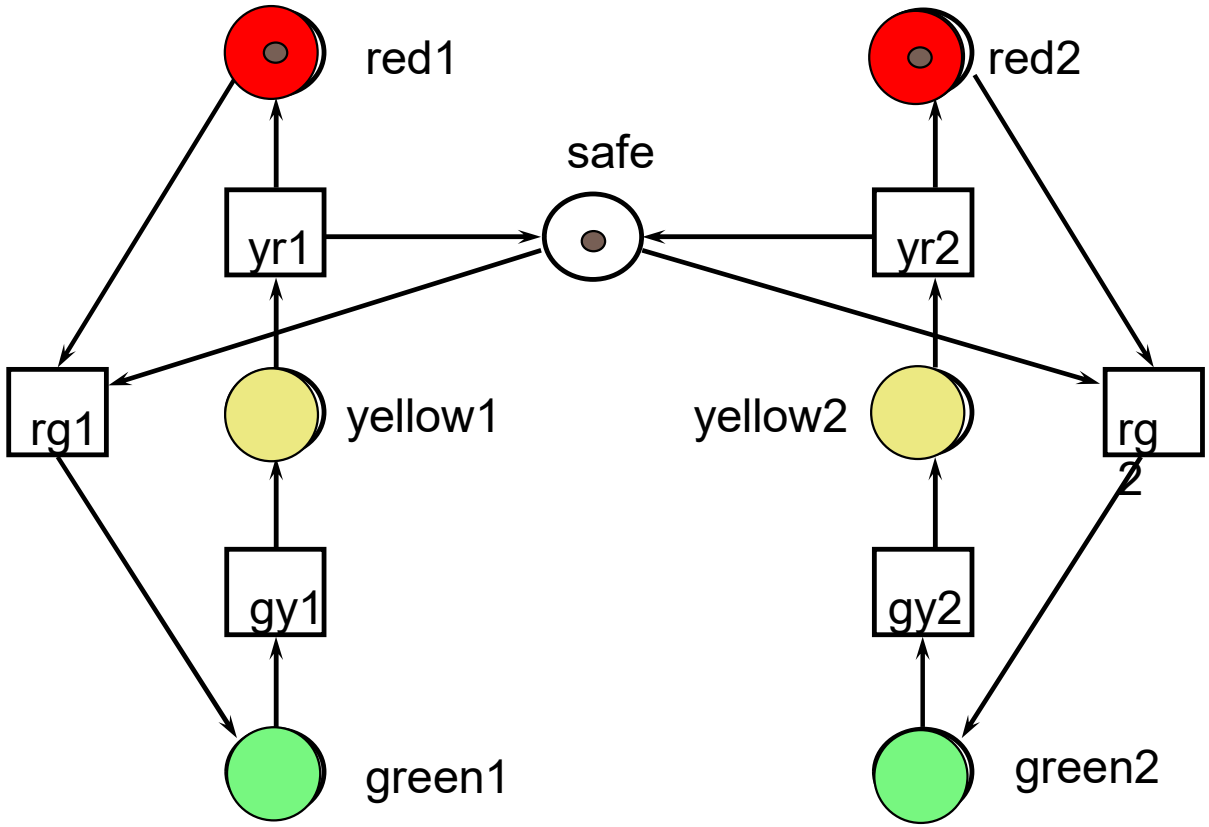
两个安全的交通灯


red1
red2
safe
yr1
yr2
rg1
yellow1
yellow2
rg2
gy1
gy2
green1
green2

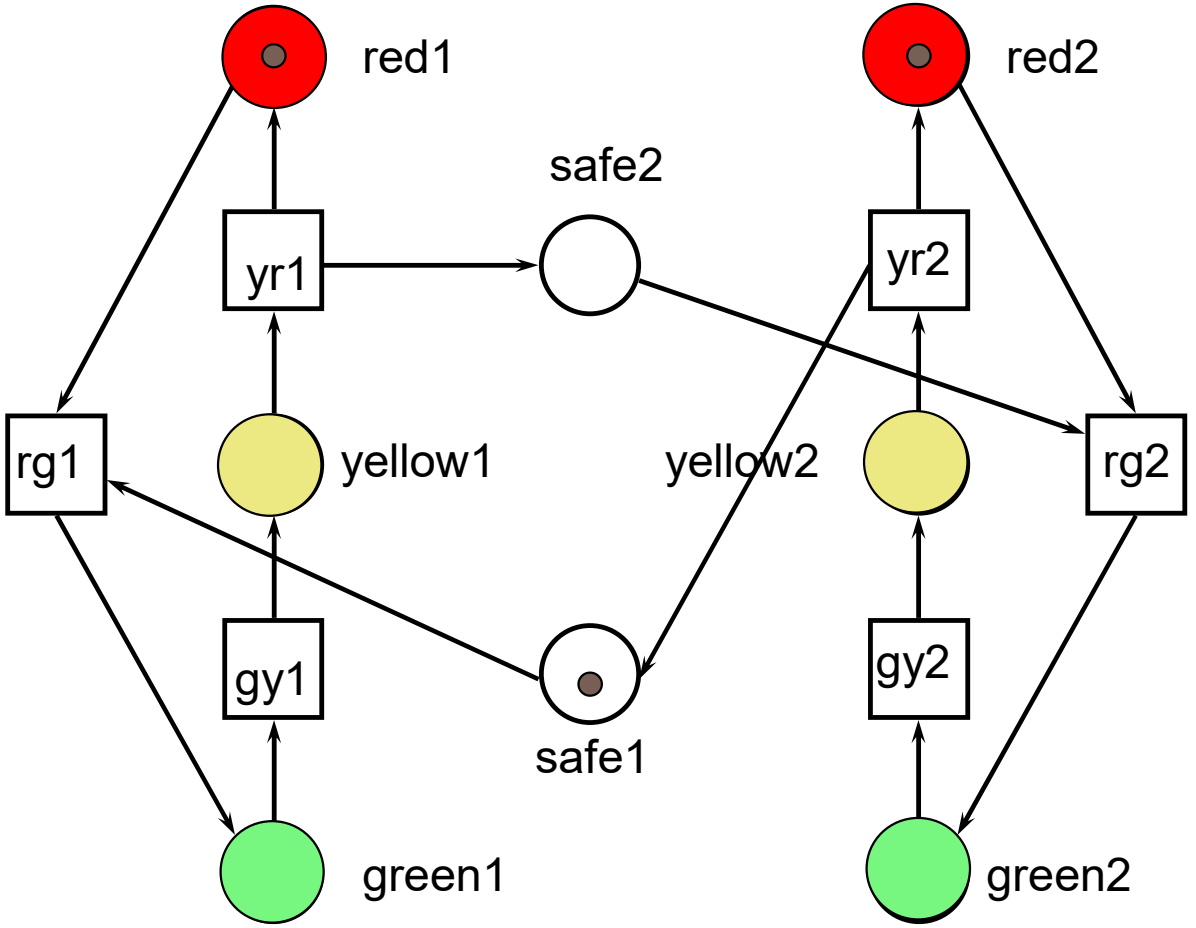
Petri网的可达性

- Petri网的状态：
  - 当前状态：Petri网的状态由库所中托肯的当前分布情况确定
  - 可达状态：通过一系列激活的转移的点火，从当前状态可以达到的状态
  - 不可达状态：从当前状态可以不能点火达到的状态
  - 死状态：没有转移能够激活的状态

交通灯的可达图练习

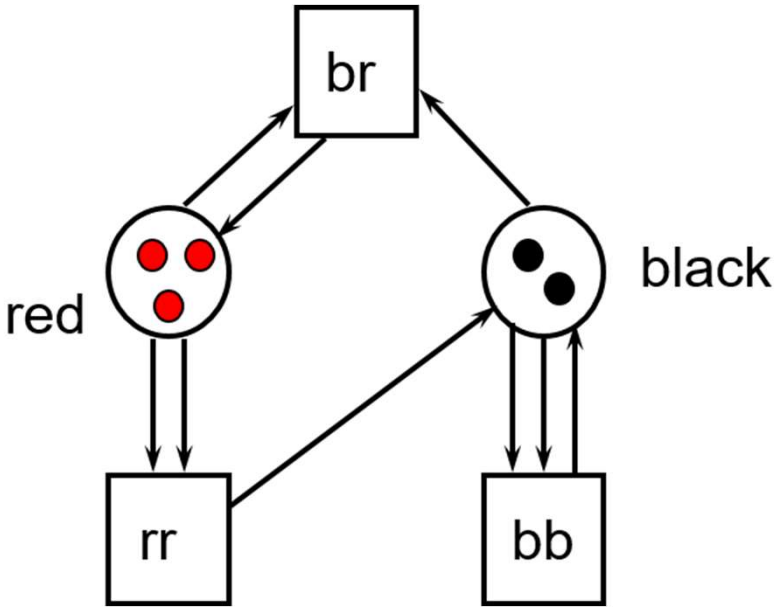

red1
red2
safe
yr1
yr2
rg1
yellow1
yellow2
rg
2
gy1
gy2
green1
green2

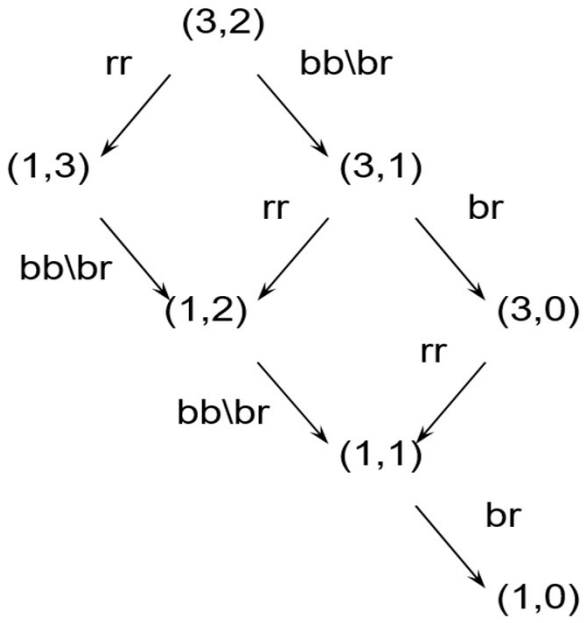
安全而公平的交通灯


red1
red2
safe2
yr1
yr2
rg1
yellow1
yellow2
rg2
gy1
gy2
safe1
green1
green2

小游戏

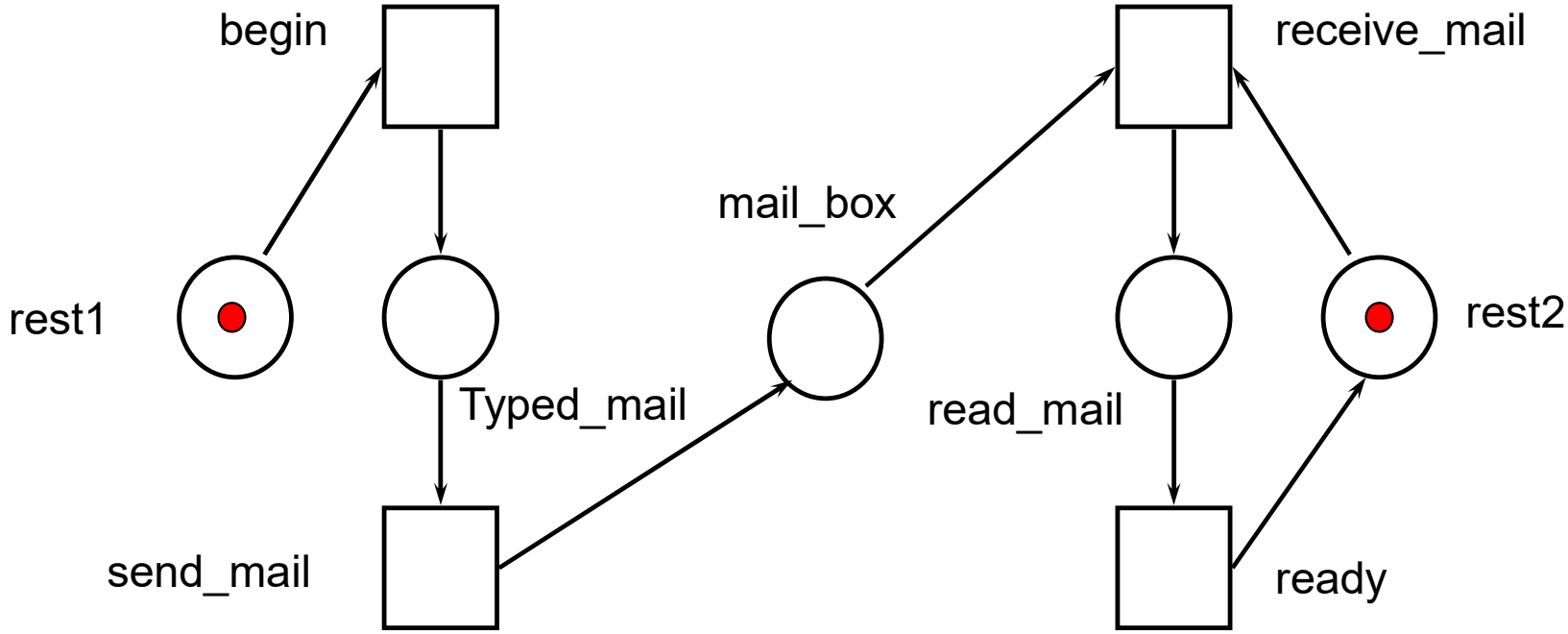
- 布袋中有一些球，拿到2黑或2红放回1黑；拿到黑红各1放回1红。其PNG及可达图如下





从可达图可以看出，有7 可达状态，有 1 死状态

练习

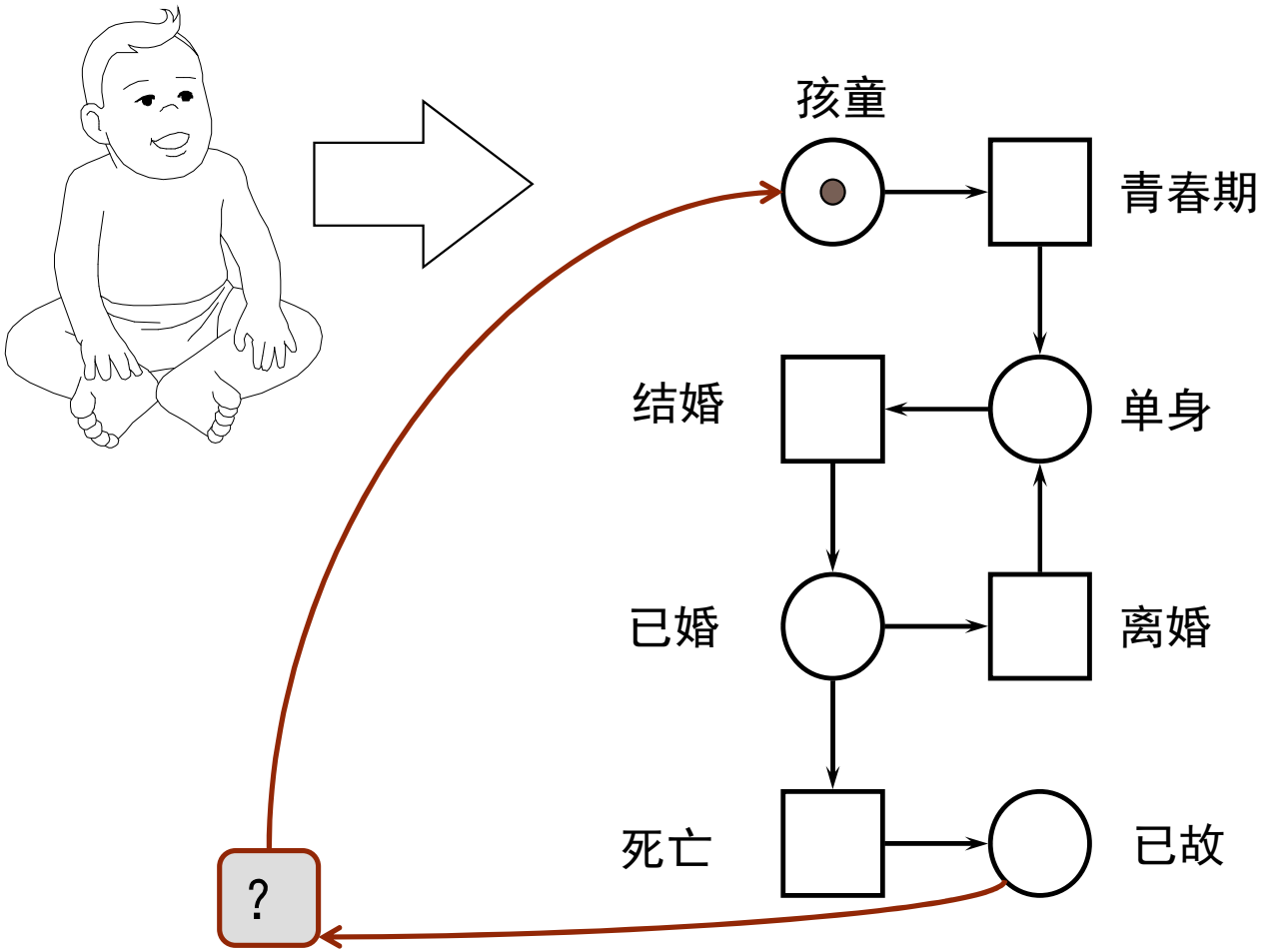


- 画出可达图
- 多少个可达状态?
- 有无死状态?

Petri网的形式化定义

- Petri网定义为五元组(库所，转移，输入函数，输出函数，初始状态)。任何图都可以映射到这样一个五元组上∑ =（P， T，F， K， M0）
  - P 为位置的集合，用圆圈代表， 表示系统的状态;
  - T 为转移的集合， 用空心矩形代表， 表示系统中的动作或者操作;
  - F 称为P->T的流关系， 其规定资源的输入流;
  - K 称为T->P的流关系， 其规定资源的输出流;
  - M0 称为Petri网 ∑ 的初始标识。
  - Token表示工作对象，转移是网络中的控制点。

人的一生



基于petri网的工作流网构造

$$PN = (P, T, F, W, M)$$

- (P, T, F)是基网，此时F为不区分F，K的统一流关系的表述；
- W是有向弧的权重函数，
- M为状态，是一个多维向量集合，$M(p)$表示对应库所P中的token数
- 对于$t \in T$，用$\bullet t = \{p \in P | (p, t) \in F\}$来表示转移t所有输入库所的集合；
- 对于$t \in T$，用$t \bullet = \{p \in P | (t, p) \in F\}$来表示转移t所有输出库所的集合。

在PN终止库所o和开始库所i之间插入一个转移t^*并添加相应连接弧，构成PN的扩展网络。

工作流网

- 构成了一个工作流网，用$\overline{\mathrm{PN}}$表示，其形式化表述为：
- $\overline{\mathrm{PN}} = (\mathrm{P}, \mathrm{T} \cup \{t^*\}, \mathrm{F} \cup \{(\mathrm{o}, t^*), (t^*, \mathrm{i})\}, \mathrm{W}, \mathrm{M})$

辅助转移　终止库所　初始库所

- PN有两种特殊的库所：i和o，i是初始库所：$\bullet \mathrm{i} = \emptyset$，o是终止库所：$\mathrm{o} \bullet = \emptyset$；
- $\overline{\mathrm{PN}}$是强连通的；

Petri网是自由选择网当且仅当对于每个转移t_1和t_2，当满足 $\bullet t_1 \cap \bullet t_2 \neq \emptyset$，则$\bullet t_1 = \bullet t_2$。

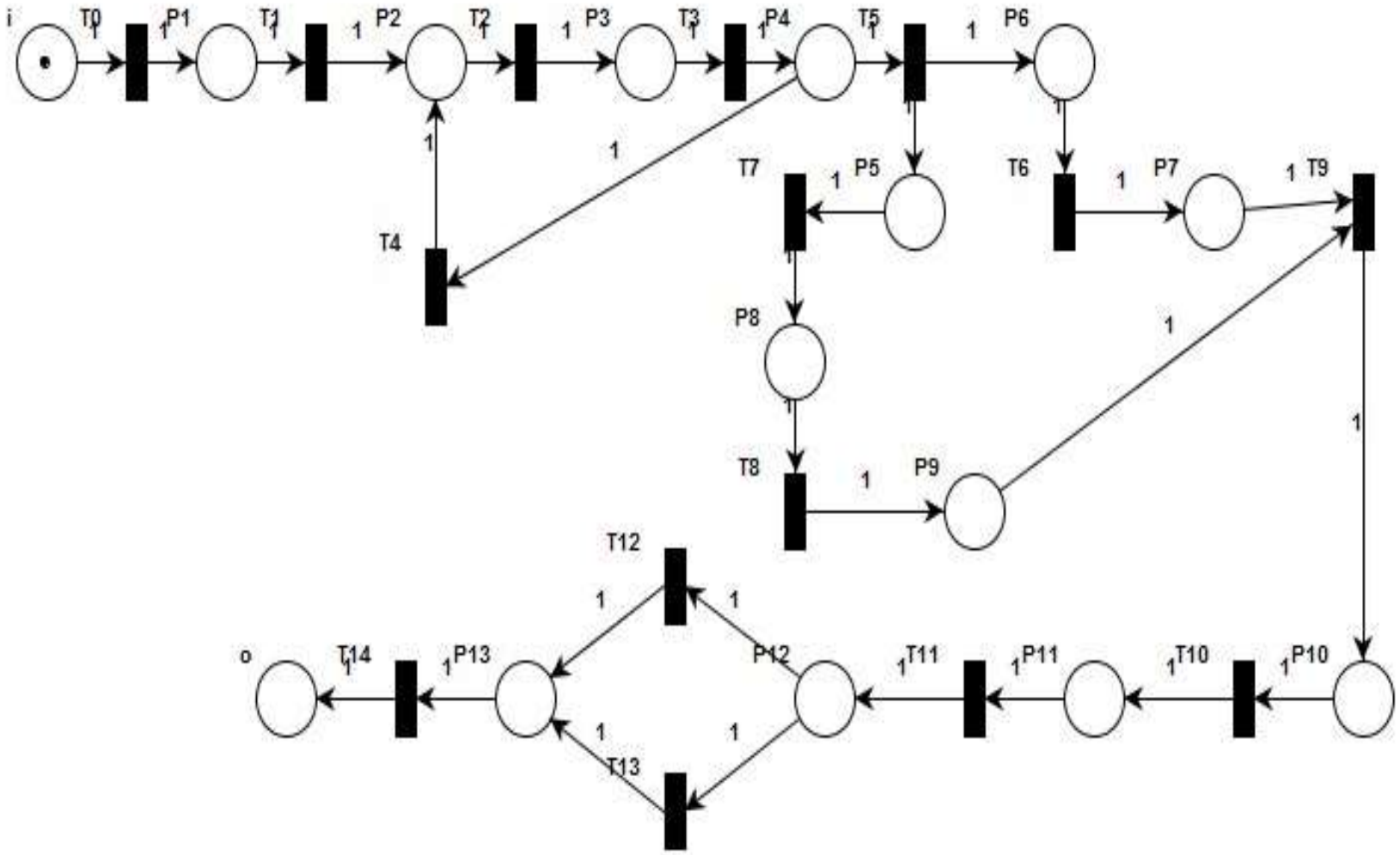
MES系统（订单拉动）流程

- MES系统从上游ERP系统接收到订单，提出加工请求申请，
- 在加工请求申请得到批准之后生成一个生产计划，
- 根据生产计划进行相应的工艺流程设计，之后对工艺流程进行审核，如果审核不通过则退回进行重新设计。
- 如果审核通过，生产流程将分两步走，一方面进行工艺工时定额，另一方面进行物料需求申请、备料。待两方面都准备好之后才能进行生产排产，
- 然后进行生产，生产结束之后将生产出的产品进行相应的质检，如质检通过则进行入库登记管理，否则将进入废品处理阶段，
- 最后将各阶段的生产资料数据进行归档统计生成报表。

要素定义

库所P	内容	转移T	业务环节定义	权重w
i	开始	T0	接受加工订单	10
P1	请求执行	T1	生成生产计划	5
P2	开始工艺设计	T2	设计工艺流程	1
P3	工艺设计结束	T3	审核工艺流程	2
P4	审核开始	T4	发回重新设计	1
P5	文件处理完毕	T5	确认工艺文件	2
P6	文件处理完毕	T6	计算工时定额	2
P7	工时定额完成	T7	产生物料需求	10
P8	需求申请结束	T8	准备物料（备料）	2
P9	备料结束	T9	安排生产（排产）	5
P10	生产排产结束	T10	产品生产	1
P11	生产完成	T11	产品质检	2
P12	质检完成	T12	废品处理	10
P13	库房	T13	成品入库	5
o	结束	T14	归档统计报表	2

生产流程的工作流网模型

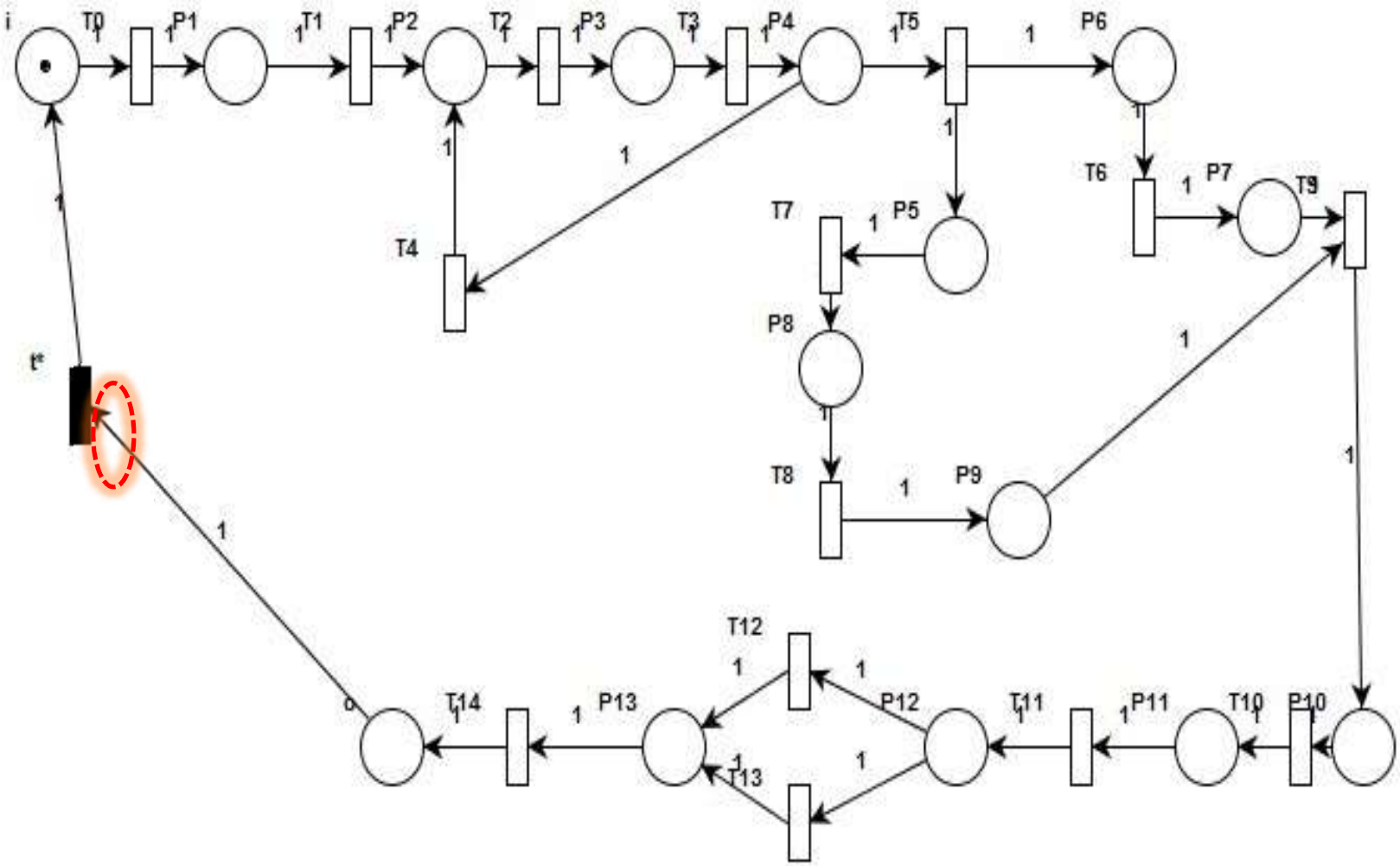


关联矩阵（流关系）

	T0	T1	T10	T11	T12	T13	T14	T2	T3	T4	T5	T6	T7	T8	T9	t*
i	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
o	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	-1
P1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P10	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0
P11	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0
P12	0	0	0	1	-1	-1	0	0	0	0	0	0	0	0	0	0
P13	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
P2	0	1	0	0	0	0	0	-1	0	1	0	0	0	0	0	0
P3	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0
P4	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	0	0
P5	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0
P6	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0
P7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0
P8	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0
P9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0

1代表p为t的输出库所，-1代表p为t的输入库所，0代表无联系

随机petri网


i
T0
P1
T1
P2
T2
P3
T3
P4
T5
P6
T6
P7
T7
P5
T4
P8
T8
P9
T12
P12
T11
P11
T10
P10
T14
P13
T13
1

转移矩阵（各状态的转移）

M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}
$-\lambda_0$	λ_0													
	$-\lambda_1$	λ_1												
		$-\lambda_2$	λ_2											
			$-\lambda_3$	λ_3										
		λ_2		$-\lambda_{4-}$ λ_5	λ_5									
					$-\lambda_{6-}$ λ_7	λ_7	λ_6							
						$-\lambda_{6-}$ λ_8	λ_8	λ_6						
							$-\lambda_7$	λ_7						
								$-\lambda_6$		λ_6				
									$-\lambda_8$	λ_8				
										$-\lambda_9$	λ_9			
											$-\lambda_{10}$	λ_{10}		
												$-\lambda_{11}$	λ_{13}	
													$-\lambda_{13}$	λ_{14}
														$-\lambda_{14}$

$$SPN = (P, T, F, W, M, \lambda)$$

维度λ(其中转移t*是瞬时转移)，代表转移在单位时间内可实施的次数，

模型应用

- 结构特性分析：可达状态分析，状态方程方法；
- 性能分析：加入延迟时间，随机数模拟；
- 仿真：标记概率密度函数，位置中的平均标记数，转移利用率，转移的标记流速；
- …

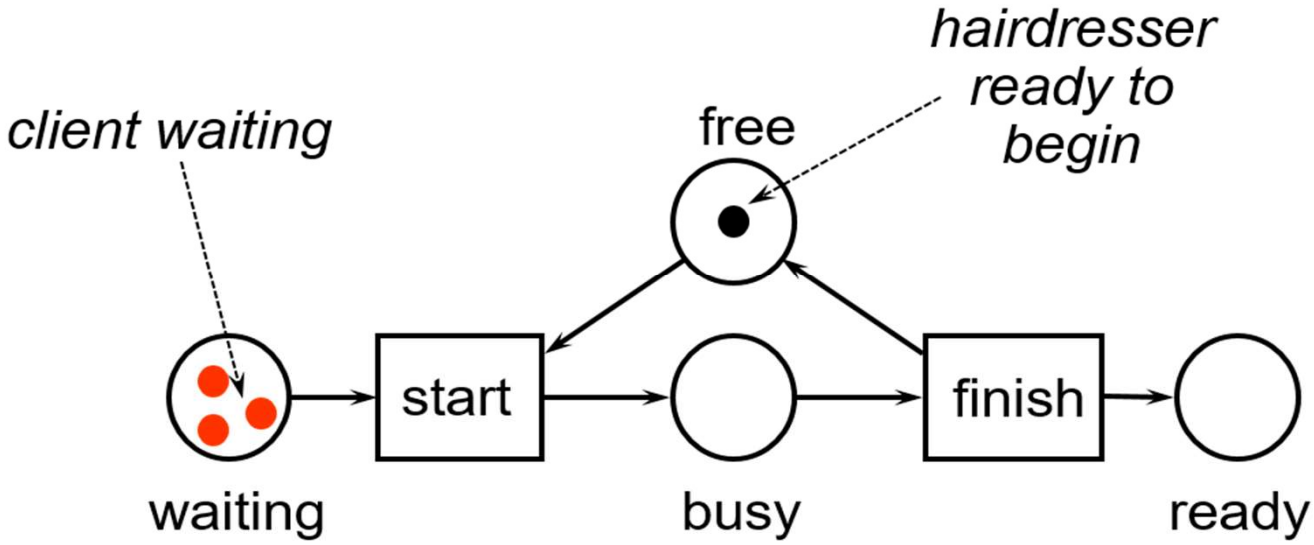
4.3 高阶Petri网

- 经典Petri网建模方法的缺点
  - 没有测试库所中零托肯的能力
  - 模型容易变得很庞大
  - 模型不能反映时间方面的内容
  - 不支持构造大规模的模型，如自顶向下或自底向上

- 高阶Petri网对以下方面进行了扩展：
  - 托肯着色：托肯拥有值（颜色）代表由托肯建模的对象的具体特征
  - 时间：每一个托肯拥有一个时间戳，转移决定生产出的托肯的延迟。并增加时序逻辑的定义
  - 层次化：构造一个与分层数据流图类似的复杂性建模机制

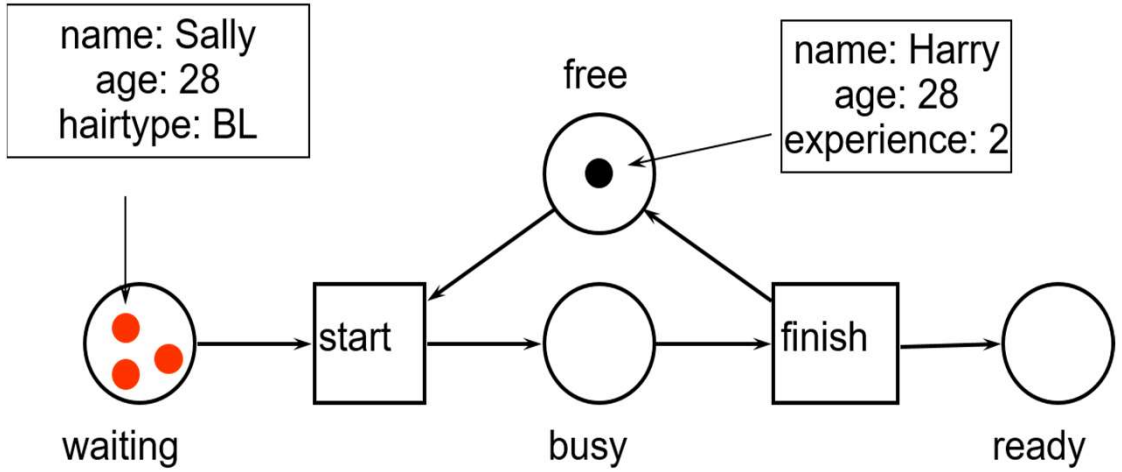
Petri网描述案例

- 高阶Petri网的扩展步骤：
  - 采用颜色进行扩展，加入产生的托肯数目、托肯的值 ，一个前提条件(可选)
  - 时间的扩展：每一个托肯都有一个时间戳
  - 对复杂的Petri网添加结构信息的方法
- 经典petri网描述一个理发厅案例：



Petri网描述案例

- 颜色的扩展

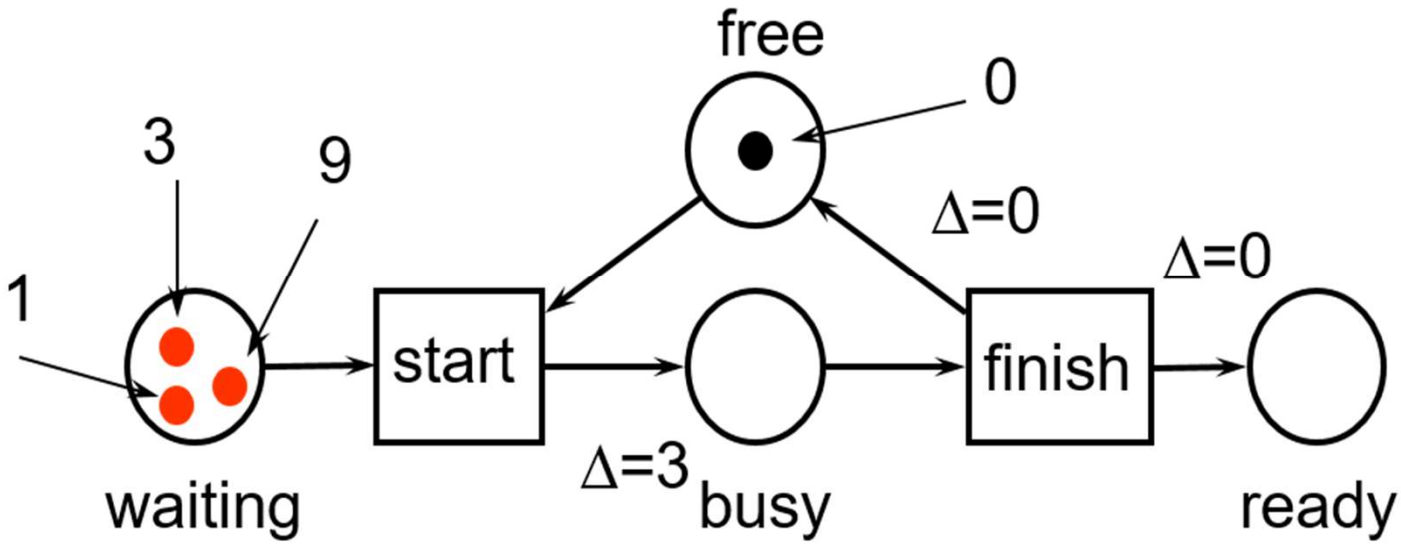


每一个转移可以有显式或者隐式的描述，包括:

- 产生的托肯数目
- 这些托肯的值
- 和(可选) 的一个前提条件

这样，网络的复杂性被分解到转移内和托肯的结构上。
这种处理产生了紧凑、可管理和自然的过程描述。

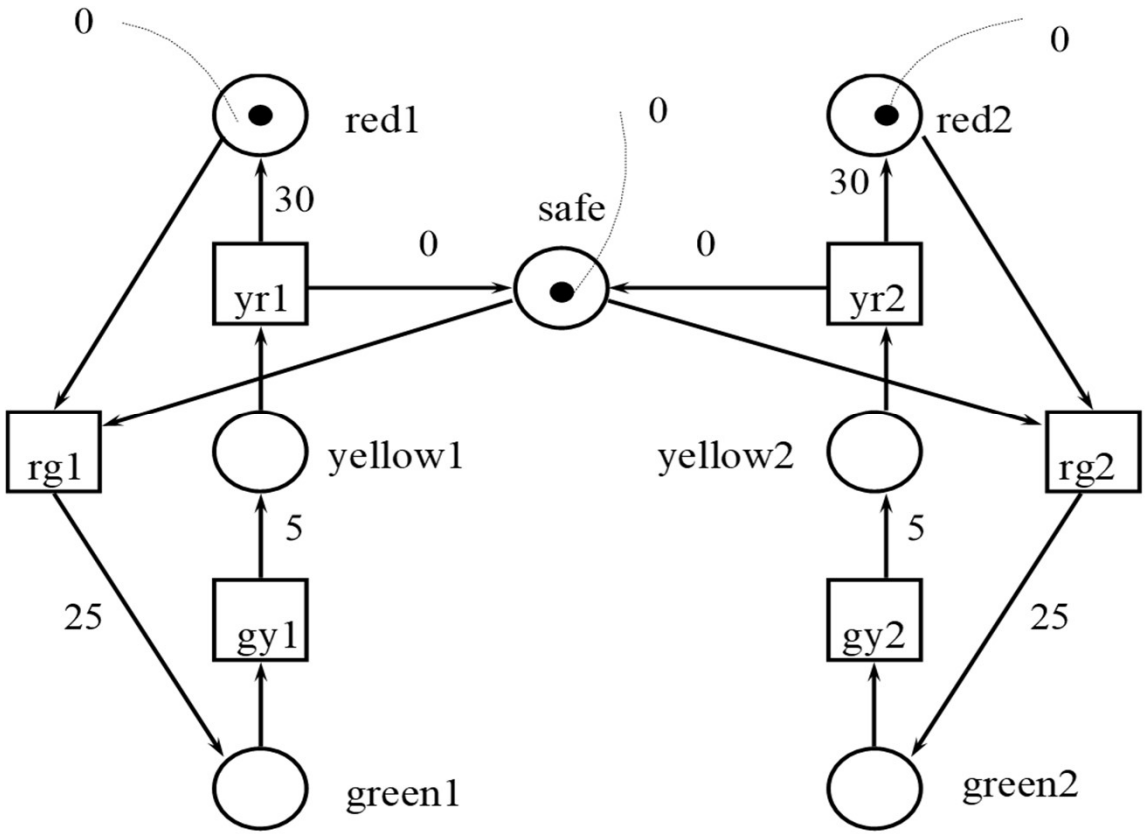
时间的扩展



为了进行性能分析，需要对持续时间，延迟等的时间概念进行建模。
扩展后，每一个托肯都有一个时间戳，而转移确定了产生一个托肯的延迟。

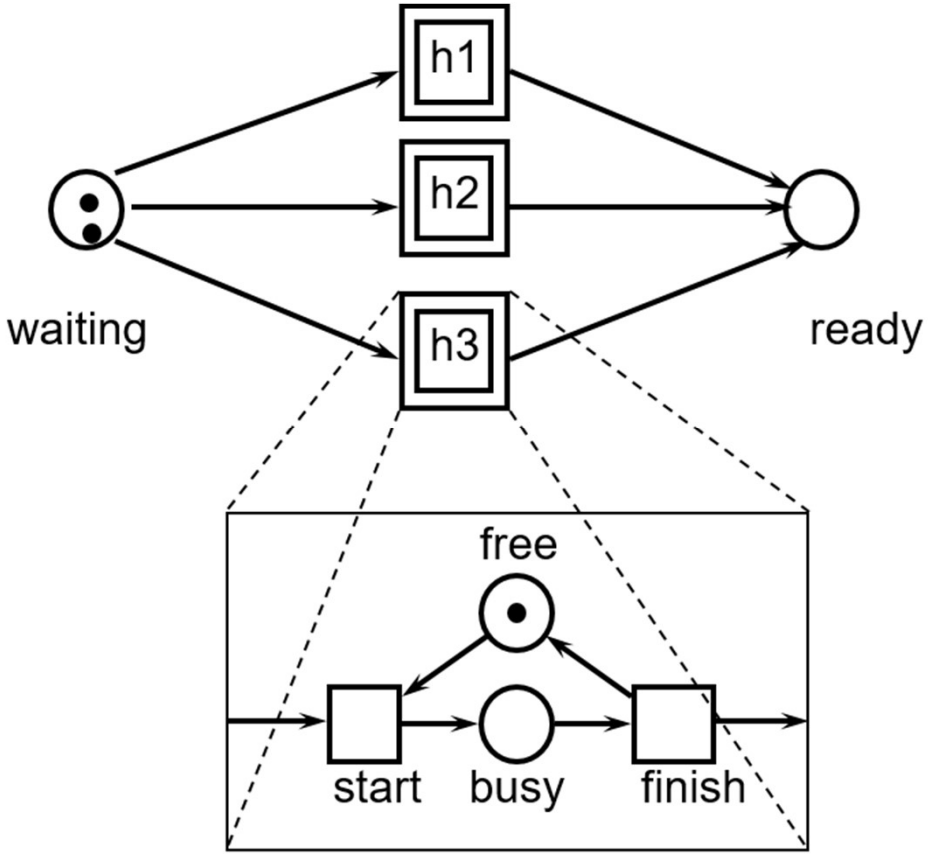
Petri网描述案例－交通灯

- 思考：本例子中的时间标注，是否合理？

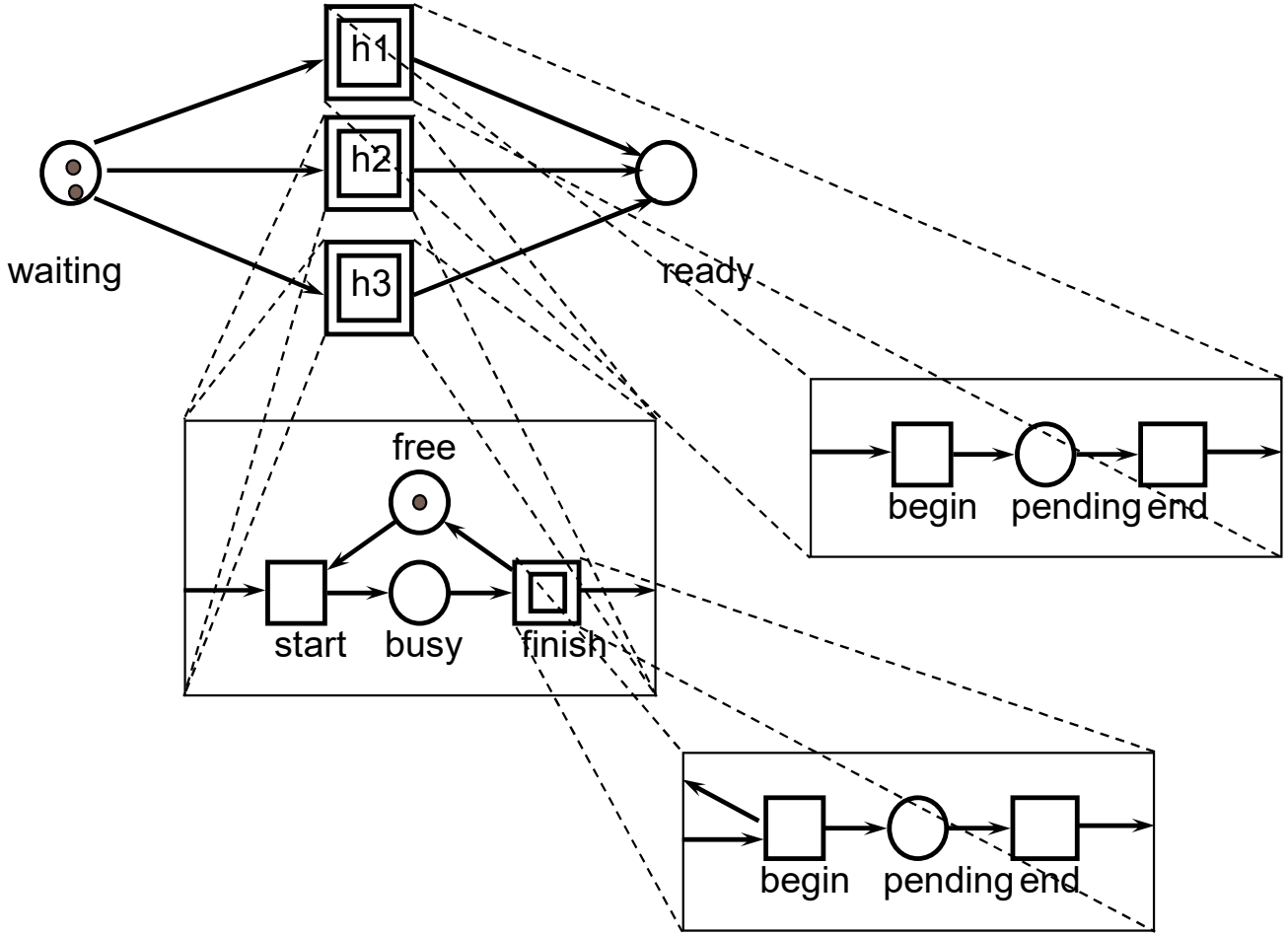


Petri网描述案例

- 对复杂的Petri网添加结构信息的方法，一个子网是对库所，转移和子网的扩展



练习：把层次去掉


h1
h2
h3
waiting
ready
free
start
busy
finish
begin
pending
end
begin
pending
end

4.4 基于Petri网的过程建模方法

- Petri网建立步骤：
  - 根据状态与事件的定义，确定系统的状态集和事件集。
  - 确定系统中状态与事件的关系。
  - 将库所和转移对应起来，建立Petri网模型图。
  - 根据系统情况，决定Petri网模型图的初始状态，确定初始状态的下个状态的Token数。
  - 基于初始状态判断那些事件可被激发，当模型激活后， Petri网模型图的状态将变化，又引起一些事件被触发。

案例－电子商务网站交易过程建模

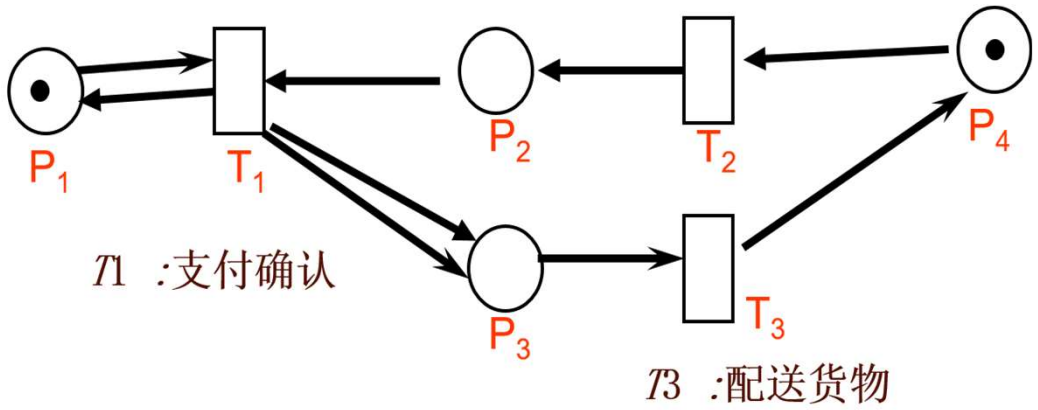
- **电子商务交易流程可用描述为：**
  - 客户通过浏览信息向商家提交订单意向，商家接到提交的订单意向后，通过查看库存信息形成可供订单；
  - 对可供订单确认后，客户进行支付，得到银行的支付确认后，商家将可供订单转为有效订单，同时产生库存信息变更；
  - 商家按有效订单配送货物，并修改库存信息。接着商家进行下次交易的处理。
- **相应的状态集和事件集如下：**
  - **转移 (活动) 可概括为**:
    - *T2* :**查看库存**; *T1* :**支付确认**; *T3* :**配送货物。**
  - **状态 (库所) 可概括为**:
    - *P1*:(**客户**)**订单**; *P2*:**可供订单**;
    - *P3* :**有效订单和库存信息**; *P4*(**商家**)**库存**

建模案例－电子商务网站交易过程

- 转移与状态的关系为

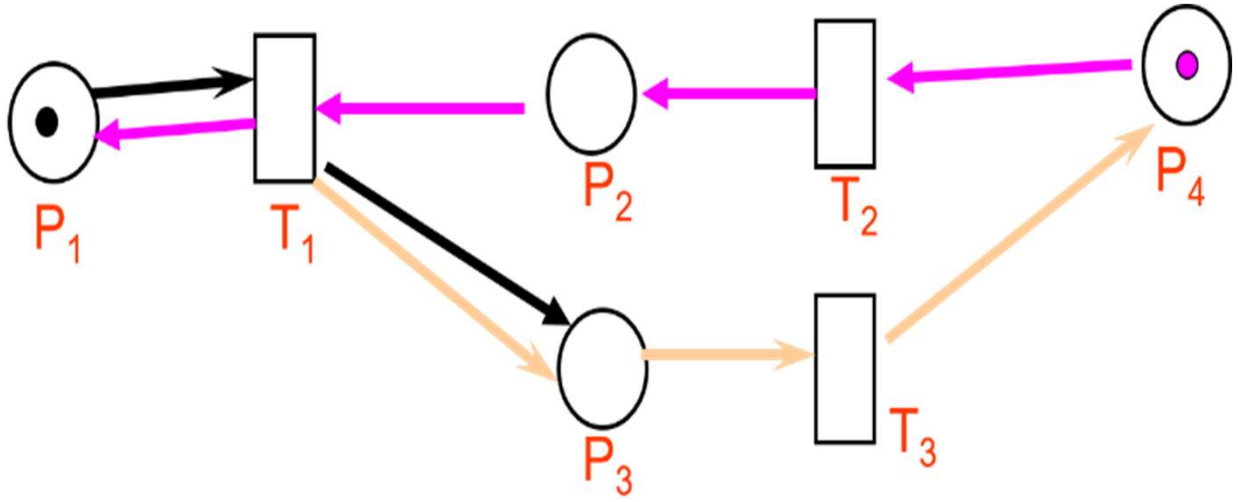
事件	前条件	后条件
支付确认 T1	***P1***：（客户）订单；***P2:***可供订单	***P1***：（客户）订单；***P2:***可供订单
查看库存 T2	***P4***（商家）库存	***P2:***可供订单
配送货物 T3	***P3*** **:**有效订单和库存信息	***P4***（商家）库存

- 设定初始时客户、商家库存的托肯均为1，Petri网模型图为



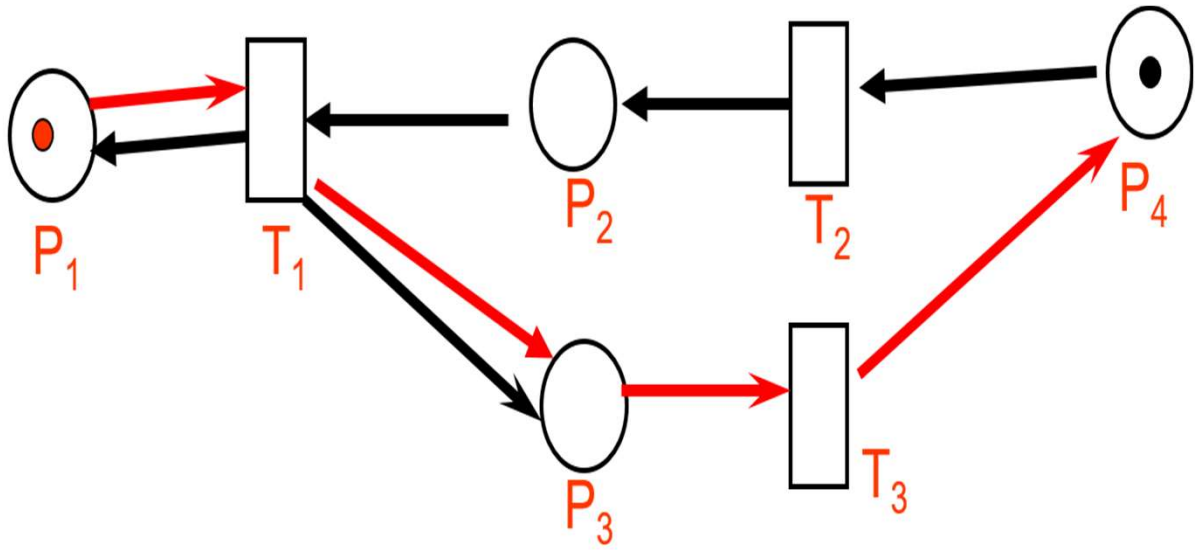
建模案例－电子商务网站交易过程

- **以商家的库存变化为例，该Petri网的状态变化为：**
  - Token ① **（商品）库存的流动过程**P4——〉P2——〉P1——〉P3——〉P4;
  - **执行转移：** T2，T1，T3



建模案例－电子商务网站交易过程

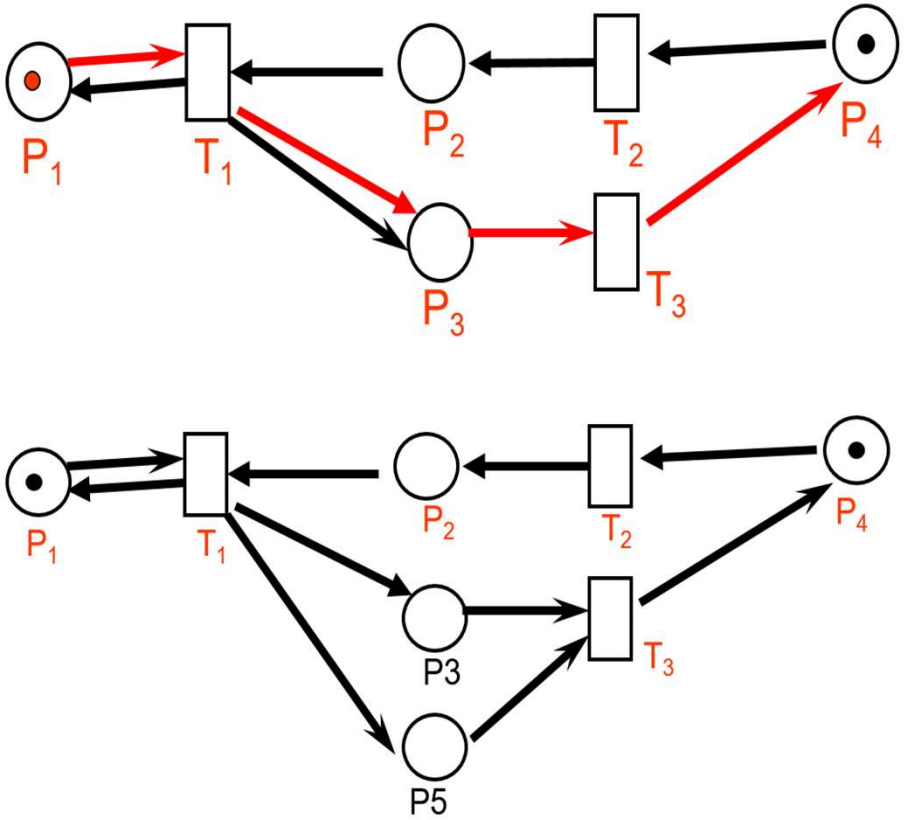
- **以用户的订单变化为例，该Petri网的状态变化为：**
  - Token ② **（客户）订单的流动过程；**P1——〉P3——〉P4
  - **执行转移：** T1，T3



基于Petri网的过程建模方法

两种PNG的比较

- P3是有效订单和可用库存
- 分解为P3有效订单和P4可用库存



再次理解：PNG的状态是静态条件Place和动态条件Token的结合。

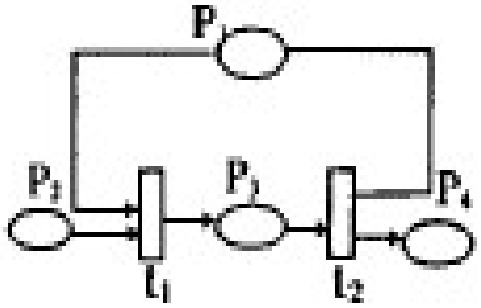
4.5 基于Petri网的流程建模及仿真实例

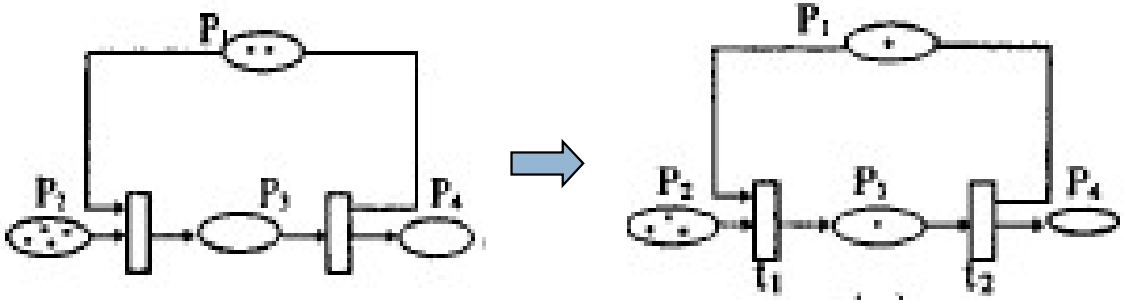
基于Petri网的仓储物流系统建模与仿真

- 一个典型的现代仓储物流系统包含入库台、输送带、AGV 自动导引小车、堆垛机、托盘、货架、空托盘堆放区、分拣区、出库台等。
- 这个物流系统的运作流程主要有出库及入库流程：
  - 入库流程: 仓库收到入库消息→货物进入库台→到达输送带→在输送的过程中向等待的AGV 发出请求→AGV 取货→取货的过程中向堆垛机发出请求→堆垛机送货到各个货位。
  - 出库流程: 按照出库单查询出库货物的具体库位→堆垛机取货→货物进入分拣系统输送带→分拣系统拣货→出库→ (空托盘清理)

一个取货环节的建模实例

- P1 :空闲AGV，待命出发
- P2: 缓冲站上货物等待运送
- P3:AGV取货送往仓库
- P4 :货物一卸载在入库台
- t1 :AGV取缓冲站上货物事件
- t2 :AGV上货物卸载在入库台事件



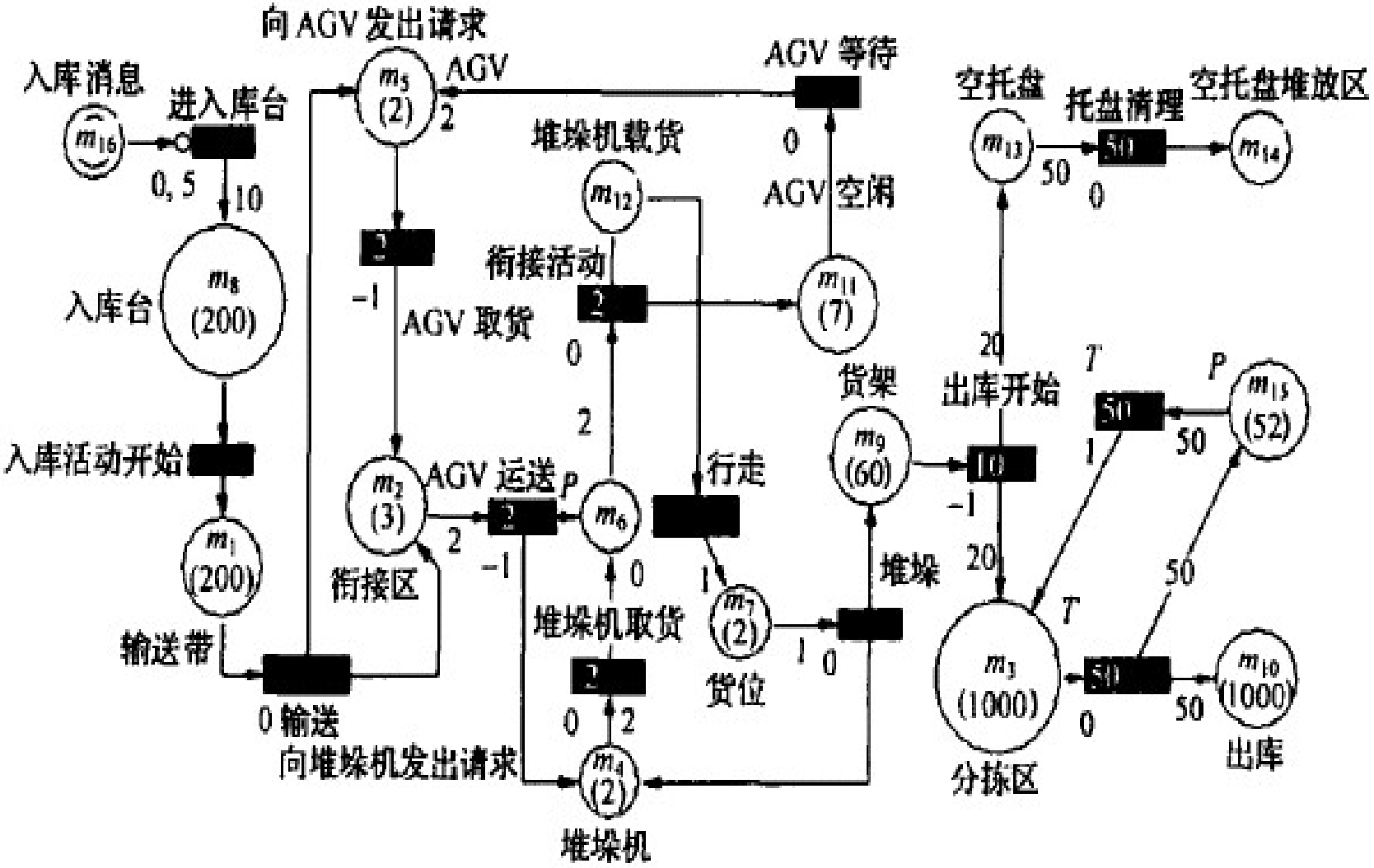


2辆AGV,4箱成品货物,t1触发，1箱货物转移。

参数设置

- 其中入库台的容量设为200,AGV 的容量设为2, 衔接区的容量为3, 货架的容量为60, 货架上每个货位的容量为2, 分拣区和出库台的容量均为1000, 输送带为可积放式, 其容量为20。
- 位置的容量可以代表标识数(或token 数)。每条边上的数据均为权数, 未加权数的边其权数都默认为1, 权数为- 1 的边的权数为任意。
- 当每个位置的标识数达到了其输出边的权数, 则T 即事件发生。
- 设货物进入库台的延迟时间为30 s。

完整流程


向AGV发出请求
入库消息
进入库台
AGV
AGV 等待
堆垛机载货
空托盘
托盘清理
空托盘堆放区
0, 5
10
2
0
50
0
AGV 空闲
衔接活动
入库台
-1
AGV 取货
0
2
货架
20
出库开始
T
P
入库活动开始
AGV 运送
P
行走
堆垛
1
-1
20
50
50
输送带
衔接区
堆垛机取货
货位
T
0 输送
向堆垛机发出请求
0
2
1
0
分拣区
出库
堆垛机

仿真过程

- 仿真开始后，当位置“入库消息”收到了10 个token 量的货物时，事件“进入库台”便启动发生,后续位置“入库台”接收到了10 个token 量的货物。
- 接着后续事件启动，货物到达“输送带”，token不断传递，各个后续事件逐一被启动。
- 其中AGV 小车在收到输送请求及空闲AGV 的token 为1 时才触发，并在衔接区取货,且同时向空闲堆垛机输出一个token，在堆垛机取货后,AGV 空车返回空闲处等待。
- 同样，堆垛机在载货行走至货位、堆垛后返回原处。
- 在分拣区,可依所需的包装数量进行设置其token、转移的容量。此时设置的转移的容量均为50。
- 空托盘的清理按定量方式进行，例中，其容量定为50 个token，即50 个托盘作一次处理，处理后在托盘堆放区增加一个token。

仿真结果

- 在仿真运行到20 m in 左右时, 入库台的容量达到饱和。入库台不再按设置的延迟时间每隔30s 收10 个token 的货物, 而是在相差大约15个token的货物时, 自动传递10 个token。
- 当仿真运行到25 m in 12 s 时, 入库台的token数为192 个, 散放空托盘积累了30 个, 空托盘堆放区已有9 堆(每堆是50 个) , 出库货物为450 个token的量;
- 当仿真运行到50 m in36 s 的时候, 入库台的token 数为187 个, 没有散放的空托盘, 空托盘堆放区已有20 堆, 出库货物为950 个托肯的量。

仿真讨论

- 基于Petri 网的物流系统仿真能够清楚地展现在逻辑时序下整个仓储系统的工作流程及系统特性.
- 从仿真结果数据来看, 入库量与出库量持平, 系统运行良好。
- 然而, 用Petri 网仿真, 只能模拟得出流程关系, 而难以模拟出其平面关系, 因此, 它的直接表现性还很不够。
- 实际过程是任意一个受概率支配的随机过程。
- 仿真给出了流程状态的变化情况。但状态随着时间而改变，相关研究有一些特定变化的数学模型，如马尔可夫过程、高斯过程等。