赋时Petri网的激发规则

定义1：Petri 网结构是一个四元组, 记为$N=(P,T,F,W)$, 其中

- P 是库所集; T 是变迁集;
- $F \subseteq (P \times T) \cup (T \times P)$ 是一个变迁和库所或库所和变迁组成的二元组的集合, 表示库所与变迁的有向弧;
- $W: F \to \mathbb{Z}^+$是一个正整数集, 表示有向弧上与权重的映射关系.

前置关联矩阵 $C^-: P \times T \to \mathbb{Z}$,

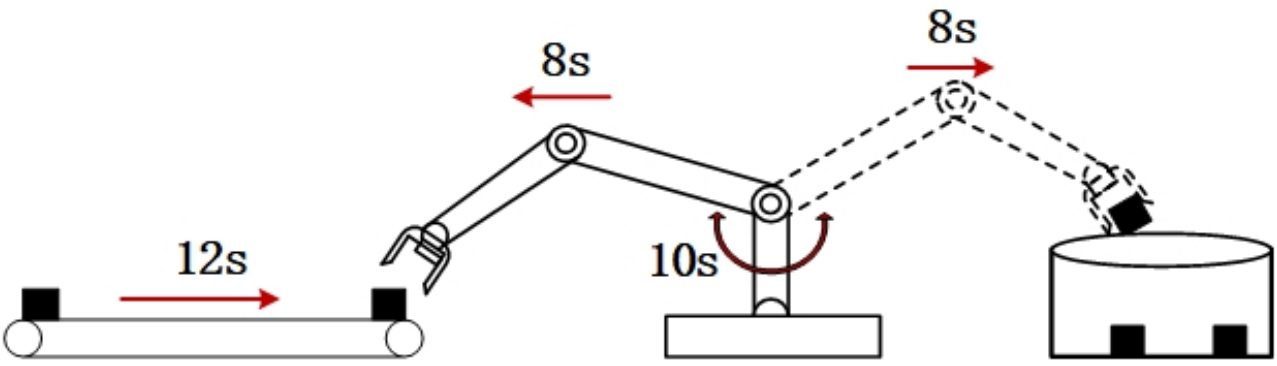
后置关联矩阵 $C^+: T \times P \to \mathbb{Z}$,

关联矩阵 $C = C^+ - C^-$.

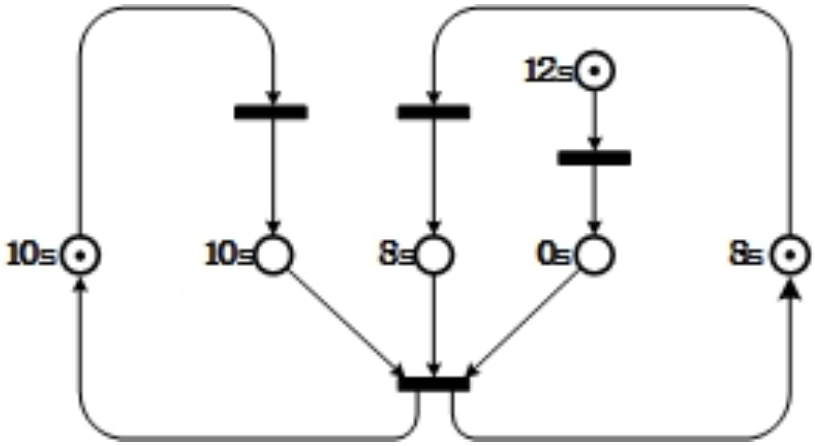
定义2：库所赋时 Petri 网是一个三元组$G=(N,d,m_0)$，其中,

- N 是一个 Petri 网结构,
- 时延 $d: P \to \{0\} \cup \mathbb{R}^+$ 是一个库所集到非负实数集的函数,
- 始标识 $m_0: P \to \mathbb{N}$, 表示初始状态下各库所拥有的托肯数目.

机械臂示例



机械臂的库所赋时Petri网模型



赋时Petri网的执行过程示意图

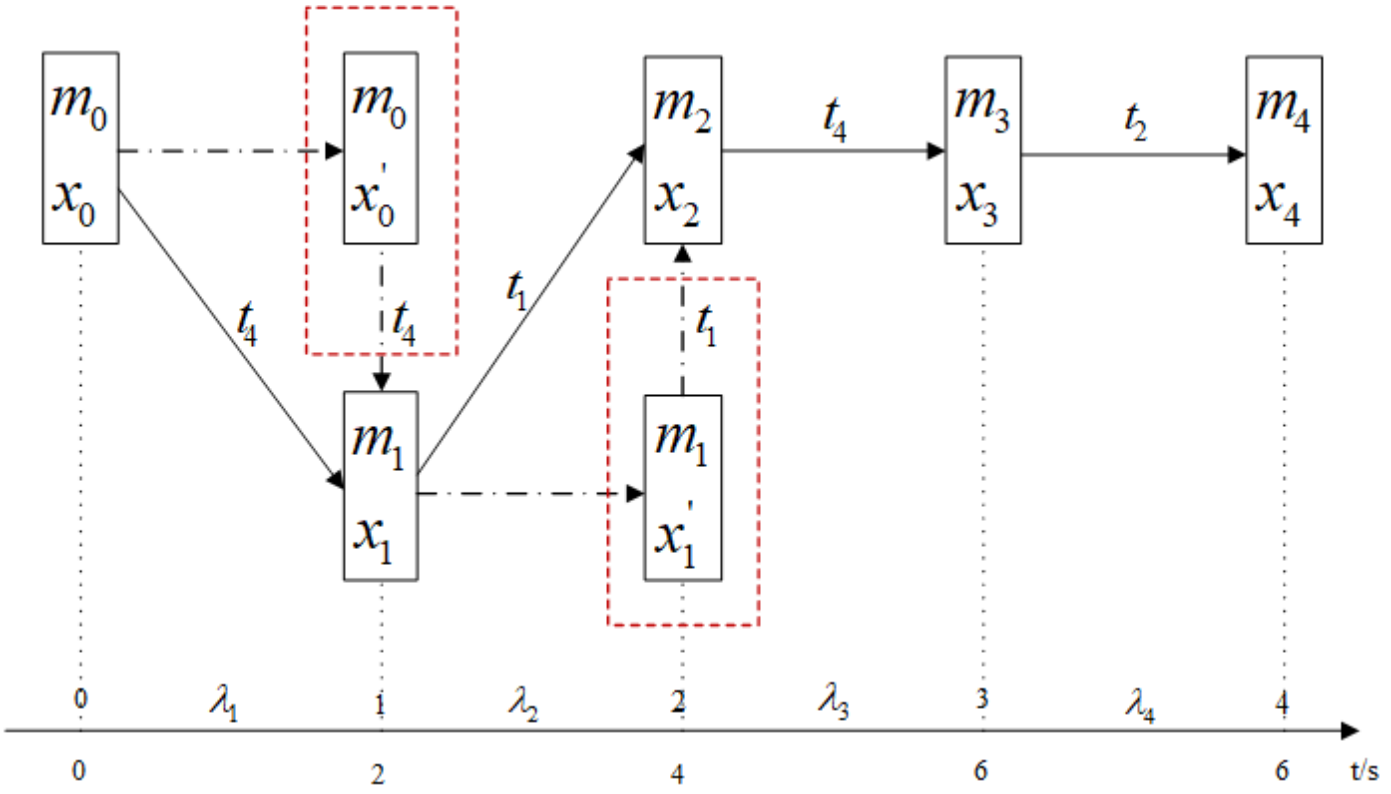
库所赋时Petri网建模规则

建模规则： 如果一个库所具有非零时延，那么该库所最多只能获得1个托肯。

定义3：给定一个库所赋时 Petri 网 $G=(N,d,m_0)$, 三元组 $X_k=(m_k,v_k,g_k)$ 表示它从初始标识开始任意激发$k\in\mathbb{Z}$ 个变迁后到达的状态, 其中：

- m_k 表示 Petri 网经过$k\in\mathbb{Z}$次变迁激发到达的标识，
- $v_k:P\to\{0\}\cup\mathbb{R}^+$ 是一个从库所集到非负实数集的映射，表示托肯在库所中已经等待的时间，
- g_k 表示 Petri 网到达 X_k 时已经消耗时间.

定义4：给定一个库所赋时 Petri 网 $G=(N,d,m_0)$ ， e_k表示它从初识标识开激发的第$k\in\mathbb{Z}$ 个变迁，而 λ_k表示它的第$k-1$和第k变迁激发之间的时间间隔。



定义5：给定一个库所赋时 Petri 网 $G=(N,d,m_0)$，任意状态 $X_k=(m_k,v_k,g_k)$ 和任意变迁t，如果

- $m_k \geq C^-(\cdot,t)$,
- $\forall p \in {}^\bullet t, v_k(p) \geq d(p)$,

那么变迁t在状态X_k下是可以激发的。

定理1：给定一个库所赋时 Petri 网 $G=(N,d,m_0)$，如果库所赋时 Petri 网处于第 k 个激发时刻的状态 X_k， λ_{k+1} 表示激发第 $k+1$ 个变迁还需等待的时间， 那么

$$m_k \geq C^-(\cdot,e_{k+1}) \tag{5}$$

$$\lambda_{k+1}=\max_{p\in {}^\bullet e_{k+1}}(d(p)-v_k(p)) \tag{6}$$

$$m_{k+1}=m_k+C(\cdot,e_{k+1}) \tag{7}$$

$$v_{k+1} = v_k - diag(v_k) \cdot C^-(\cdot, e_{k+1}) + \lambda_{k+1} \cdot (m_{k+1} - C^+(\cdot, e_{k+1}))$$

$$\forall p \in \mathcal{P}, v_{k+1}(p) = \begin{cases} 0, & \text{if } d(p) = 0; \\ 0, & \text{if } d(p) \neq 0 \wedge p \in \left(^\bullet e_{k+1} \cup e_{k+1}^\bullet\right); \\ v_k(p) + \lambda_{k+1}(p), & \text{otherwise.} \end{cases}$$

$$g_{k+1} = g_k + \lambda_{k+1} \tag{8}$$

.

算法 1 赋时 Petri 网可达图算法

输入： C^-, C^+, d, m_0, m_g .

输出： 赋时 Petri 网可达图

1. 初始化 $X_0 = (m_0, v_0, g_0)$, 其中 m_0 为初始标识, $v_0 = 0$, $\lambda_0 = 0$, $g_0 = 0$;
2. 将初始状态 $X_0 = (m_0, v_0, g_0)$ 作为可达树的根结点, 并标记为 new;
3. 如果 $m_0 = m_g$, 将 X_0 标记为 $goal$, 算法退出, 否则继续执行下一步;
4. 从可达树中任意选择一个 new 结点, 表示为 $X_k = (m_k, v_k, g_k)$, 并将该结点标记为 old;
5. 计算 X_k 状态下状态使能的变迁集合 $E_{k+1} = \{t \in T | m_k \geq C^-(\cdot, t)\}$;
6. for all $e_{k+1} \in E_{k+1}$ do
7. 激发 e_{k+1}, 产生新状态 X_{k+1}, 根据公式 (2)-(5) 分别计算 λ_{k+1}、m_{k+1}、v_{k+1} 和 g_{k+1};
8. 如果 X_{k+1} 是新状态, 在可达树中添加一个表示 X_{k+1} 的结点;
9. 画一条从结点 X_k 指向结点 X_{k+1} 的有向弧, 弧上标记激发变迁 e_k;
10. 如果 $m_{k+1} = m_g$, 将 X_{k+1} 对应的结点标记为 $goal$, 否则标记为 new;
11. end for
12. 如果可达树中存在 new 结点, 那么返回第 4 步;