arXiv:2603.21742v1 [cs.FL] 23 Mar 2026

用于并发离散事件控制的解释型高维自动机

Dylan Bellier[2], Gregory Faraut[1], Yan Monier[1], Philipp Schlehuber-Caissier[2]

摘要——近年来，从理论角度来看，高维自动机（HDA）理论取得了重大进展，体现了标准自动机理论的发展（例如 [7]）。此外，也有人开始尝试将高维自动机所提供的数学框架应用于已知问题，尤其是 Petri 网分析（见 [14], [3]）。然而，现实世界的应用仍然缺乏；在控制器生成领域，诸如并发等现实世界系统问题仍未得到解决。本文展示了如何调整高维自动机框架，以帮助将以解释型 Petri 网（IPN）表示的控制器转化为实际的闭环控制器，以及高维自动机如何帮助识别 IPN 中仍然“隐藏”的模糊甚至相互矛盾的规范。我们通过将生成的控制器连接到用于闭环控制的虚拟环境来展示其可行性，并以一个工业实例为例。

I. 引言

现代信息物理生产系统（CPPS）日益呈现分布式和并发特征，这使得采用离散事件系统（DES）形式化方法进行控制器设计与验证更具挑战性 [2]。尽管形式化的离散事件系统方法能够为验证、诊断和综合提供有力保证，但其实际有效性取决于所选语义在多大程度上反映了执行时间实际发生的情况 [13]。在许多工业场景中（PLC扫描周期、异步输入/输出、通信延迟），模型中被视为瞬时发生的动作实际上可能同时发生，或者其发生顺序并不受设计者控制 [4]。

一种常见的建模方式是通过交错来表示并发：将并发事件串行化为序列 [4]。这种抽象通常足以处理可达性类问题，但可能掩盖由并发引发的不一致，而这类不一致只有在多个事件“同时”发生时才会显现。输出赋值就是一个典型例子：网中彼此独立（因此在现实中可能并发）的两个事件，可能会施加相互冲突的输出更新；然而，如果强制这些事件按某种顺序发生，这种冲突就可能无法显现。当规范不完整或约束宽松时，这类问题尤为棘手，因为多个实现可能都满足交错模型，但在并发执行时表现却各不相同 [11]。

本文借助高维自动机（HDA） [14], 提出一种用于控制规范的真并发视角：高维胞腔明确表示并发步骤，而非将其编码为任意交错。本文介绍一种解释型变体，称为解释高维自动机（IHDA），并基于解释型 Petri 网（IPN） [5], 纳入输入/输出，使模型可直接用于自动化控制。在 IHDA 中，并发不仅得到表示，还成为可分析的对象，从而能够在同时动作的层面进行形式化验证。

本文的主要贡献是提出一种从解释型 Petri 网（IPN）到 IHDA 的转换，该转换在保留标准顺序行为的同时，为可容许的并发激发显式生成高维胞腔。在此构造的基础上，我们提出一种简单机制，用于检测并发情况下的输出不一致：如果并发步骤导致不兼容的输出要求（例如，同时进行相反的更新），这种不一致就会明确体现于相应的胞腔约束中，并可提取出一个见证执行。我们通过一个具有代表性的制造示例展示了该方法，该示例在仿真工具中实现，说明 IHDA 如何帮助弥合采用常见形式化方法（自动机或 Petri 网）设计的离散事件系统控制器与物理执行层所引发的并发效应之间的差距。

本文其余部分安排如下：第二节回顾解释型 Petri 网的相关背景，第三节回顾高维自动机的相关背景。第四节介绍 IHDA 的定义以及从 IPN 到 IHDA 的转换。本节还介绍不一致性检测原理。最后，第五节展示一个使用仿真工具设计的制造系统案例研究。

II. 解释型佩特里网

解释型 Petri 网（IPN）是控制与自动化领域中一种广为人知的形式化方法，可用于将 Petri 网用作有效的被控对象控制器。为此，Petri 网扩展了输入和输出信号，以便与物理对象 [5], [12] 进行接口连接。

IPN 的输入和输出通常直接关联到传感器或执行器。本文采用稍微更通用的方法，使用原子命题集合

[1]巴黎萨克雷大学，巴黎萨克雷高等师范学校，LURPA，法国。firstname.lastname@ens-paris-saclay.fr[2] SAMOVAR，巴黎电信南方高等学校，巴黎综合理工学院，法国。部分经费由学术与研究讲席“复杂系统架构”提供，资助方包括达索航空、海军集团、达索系统、KNDS法国、国防创新局、巴黎综合理工学院

（AP）作为输入和输出，这在处理（输入-输出）自动机 [6], [9] 或反应式综合 [8] 时是标准做法。一个原子命题对应一个布尔变量，自动机中的变迁可以用这些命题上的布尔函数而非简单字母来标记。虽然这不会改变自动机的表达能力，却能大幅提升其实用性。

这里我们采用类似的方法来定义 IPN 的输入和输出。为使概念简洁，我们将使用一类称为立方项的布尔函数子集。立方项是仅使用合取、原子命题及其否定（即不含析取）的布尔函数。它们由语法 $c ::= ap \mid \neg ap \mid c \wedge c$ 生成，其中 c 是立方项，ap 是原子命题。最后，我们将包含所有原子命题（每个原子命题都与 $\top$ 或 $\bot$ 关联）的立方项称为赋值。这一限制使表述更为简洁，不过将来很容易取消。

形式化地，我们有：

定义 *1*：一个解释型 *Petri* 网 *(IPN)* $Q = \{G, M_0, I, O, \lambda_{pn}\}$，其中 (G, M_0) 是一个标准的带标记 Petri 网，$G = (P, T)$ 是其库所和变迁，并且

- I 是一个有限的原子输入命题集合，
- O 是一个有限的原子输出命题集合，
- λ_{pn} 为标记函数。

标记函数 λ_{pn} 将原始网中的每个库所关联到一个关于输出的立方项，并将每个变迁关联到一对立方项，一个关于输入，另一个关于输出，分别记作 $\lambda_{pn}(t)[0]$/$\lambda_{pn}(t)[1]$，表示输入/输出条件。

在此语境下，标准 Petri 网不带权、不着色，也不包含抑制弧或任何其他扩展。这样有意选择是为了使定义保持简洁。不过，对许多此类扩展进行直接改造是可行的，因为只需修改相应高维自动机的构造（见第 III 节），而且文献 [3] 中介绍了许多这类技术的专门构造。

此类 Petri 网中的变迁发射受输入条件约束。也就是说，变迁 t 只有在（与常规 Petri 网一样）所有前置库所 $^\bullet t$ 中至少各有一个托肯，并且当前输入赋值与 $\lambda_{pn}(t)[0]$ 兼容时，才能发射。形式上，输入表示为输入命题上的一个赋值，记作 i；当且仅当 $i \models \lambda_{pn}(t)[0]$ 时，该输入与变迁 t 兼容。

对此类 Petri 网的一次带解释执行，会将输入赋值流转换为输出赋值流。我们将这两个流组合起来解释为无限字：$w = (i_0,\ o_0)(i_1,\ o_1)(i_2,\ o_2)\ \ldots \in (\mathbb{B}^{|I|},\ \mathbb{B}^{|O|})^\omega$。如果存在该 IPN Q 上的一次运行 $r = m_0 m_1 m_2 m_3 \ldots \in \mathbb{N}^{|P|^\omega}$，使得在两个标识之间要么发生一次变迁，要么由于当前标识和/或当前输入赋值而没有变迁可触发，因而标识保持不变，那么我们就称这样的词 w 与 IPN Q 兼容。此外，输出也必须兼容。

形式化地，我们有

- 等待步 k：$m_k = m_{k+1}$ 因此不存在变迁 $t \in T$ 使得 $m_k \geq {}^\bullet t$ 和 $i_k \models \lambda_{pn}(t)[0]$ 同时成立，并且我们有 $o_k \models \bigwedge_{p \in P:\ m_k[p]>0} \lambda_{pn}(p)$
- 转移步 k：此处执行一个变迁 t（因此必须验证标识和输入约束），并且我们有 $o_k \models \bigwedge_{p \in P:\ m_k[p]>0} \lambda_{pn}(p) \wedge \lambda_{pn}(t)[1]$，且标识根据 $m_{k+1} = m_k - {}^\bullet t + t^\bullet$ 演化。

直观而言，产生的输出对应于库所输出的合取；若发生转移步，则还对应于与所触发变迁相关联的输出约束。

III. 高维自动机

高维自动机（简称 HDA）是一种用于表示系统真实并发的数学框架。直观而言，HDA 是一组胞腔，每个胞腔都带有模型系统中可能并发发生的动作标签。这些胞腔通过面映射相连：面映射是一种函数，用于定义通过终止或启动当前活跃动作的子集，可以到达哪些胞腔。在近期的工作 [3],中，我们展示了如何使用 HDA 对 Petri 网建模，其中上述动作对应于 Petri 网的变迁。由于对 Petri 网中的同一变迁发射两次，会对当前标识产生相同的改变，且无法通过其他方式加以区分[1], ，因此我们无需追踪变迁的身份。它们的行为完全相同，可以互换。因此，我们可以在此采用匿名 HDA，使形式化方法更加简单。在本文其余部分中，提及 HDA 时均指匿名 HDA。

A. 形式化定义

给定字母表 Σ，我们用 $\square(\Sigma)$ 表示 $\square$（当字母表可由上下文明确时），即 Σ 上的多重集集合。这样的 Σ 上的多重集称为并发集，简称并发集。

请回顾，多重集是集合的推广，其中元素可以重复出现。形式上，集合 E 上的多重集 f 是一个函数 $f : E \to \mathbb{N}$。对于满足以下条件的多重集 f，我们采用通常的记号 $f_1 + f_2$：对每个 $e \in E$，都有 $f(e) = f_1(e) + f_2(e)$；当对每个 $e \in f_1$ 都有 $(e) \leq f_2(e)$ 时，我们也记作 $f_1 \leq f_2$；最后，当 $f_1 \leq f_2$ 时，对于满足以下条件的多重集 f，我们记作 $f_2 - f_1$：对每个 $e \in E$，都有 $f(e) = f_2(e) - f_1(e)$

定义 *2*：一个（匿名）预立方集 $\mathcal{X} = \langle X, \mathrm{ev}\ \Delta \rangle$ 由一组胞腔 X、一个函数 $\mathrm{ev} : X \to \square$ 以及一组面映射 Δ 构成。对于并发集 U，我们将 $X[U] = \{x \in X \mid \mathrm{ev}(x) = U\}$ 记为类型为 U 的胞腔。此外，对于每个 U，A，$B \in \square$，若 $A + B \leq U$，则存在面映射 $\delta_{A,\ B;U}$：$X[U] \to X[U - (A + B)]$，它们满足

$$\delta_{C,D;U-(A+B)} \delta_{A,B;U} = \delta_{A+C,B+D;U} \tag{1}$$

对于每个 $U \in \square$、$A + B \leq U$ 和 $C + D \leq U - (A + B)$。

我们将省略面映射中的额外下标“U”，并进一步记作 $\delta^0_A = \delta_{A,\emptyset}$ 和 $\delta^1_B = \delta_{\emptyset,B}$。上 面映射

[1]请参阅 [14]，了解有关个体和集体令牌语义的详细讨论。此处我们采用标准集体令牌语义。

δ_B^1 将单元 x 变换为一个其中 B中的变迁已终止的单元；下面映射 δ_A^0 将 x 变换为一个其中 A 中的变迁尚未开始的单元，也就是说，从 x的角度来看，这些变迁尚未启动。每个面映射$\delta_{A,B}$ 都可以写成复合形式$\delta_{A,B} = \delta_A^0\delta_B^1 =\delta_B^1\delta_A^0$，而预立方恒等式（1）表明这些变换是可交换的。

我们将 $X_n = \{x \in X \mid |\mathrm{ev}(x)| = n\}$ 记作 $n \in \mathbb{N}$，并将其元素称为 X_n n-胞腔。 $x \in X$ 的维数为 $\dim(x) = |\mathrm{ev}(x)| \in \mathbb{N}$； X 的维数为 $\dim(X) = \sup\{\dim(x) \mid x \in X\} \in \mathbb{N} \cup \{\infty\}$。对于 $k \in \mathbb{N}$， X 的 k-截断是预立方集 $X^{\le k} = \{x \in X \mid \dim(x) \le k\}$，其中所有维数高于 k 的胞腔均被移除。

一个高维自动机（*HDA*） $\mathcal{A} = (\Sigma, \mathcal{X}, I)$由有限字母表 Σ、预立方集 $\mathcal{X} = \langle X, \text{以及定义在 } \Sigma \text{ 上的 ev } \Delta\rangle$，和初始胞腔子集 $I \subseteq X$ 组成。本文不需要接受胞腔，通常只考虑一个初始胞腔。一般而言，高维自动机可以是有限的、无限的，甚至是无限维的；不过，我们将范围限定为有限维有限高维自动机，这与 [3] 中的做法相同，足以表示有界（亦称 k-安全）的 Petri 网。我们自然地将预立方集的所有记法推广到高维自动机。

在本研究中，我们在定义计算时，为高维自动机赋予所谓的并发步语义。并发步计算是一个序列

$$\rho = (z_0, x_0, z_1, \ldots, z_{n-1}, x_{n-1}, z_n) \tag{2}$$

其中 z_i 0 $\delta^0_{\mathrm{ev}(x_i)}\ x_i = z_i$ $i \in$ 是 -胞腔，并且对于每个，） $\{0, \ldots, n-1\}$ $\delta^1_{\mathrm{ev}(x_{i-1})}\ x_{i-1} = z_i$ $i \in$ （）；对于每个 $\{1, \ldots, n\}$ $\mathcal{A}$ ，）（）。高维自动机的计算集合记为 Comp(𝒜)。

直观而言， z_i 标记控制周期的变化，而 x_i 对应于发生在 z_i 与 z_{i+1} 之间的所有动作的集合。

如果存在一条从 x_S 到 x_T 的路径 ρ，则称单元 $x_T \in X$ 从 x_S 可达，也就是说，在上文的记号 (2) 中， $z_0 = x_S$ 且 $x_n = x_T$。

B. 从 *Petri* 网到高维自动机

在 [3], 中，Petri 网 $N = \langle S, T, F\rangle$ 可以转换为高维自动机 pn2HDA(N) $= \langle \mathcal{X}, I\rangle$，其中 $\mathcal{X} = \langle X, \text{ev } \Delta\rangle$ 定义如下。令 $\square = \square(T)$，并定义 $X = M \times \square$ 和 ev : $X \to \square$，使得 ev((m, τ)) $= \tau$，其中 M 是 $\mathbb{N}^{|P|}$ 中 Petri 网的标识。对于 $x = (m, \tau)$，若 $\tau = (t_1, \ldots, t_n)$ 非空且 $i \in \{1, \ldots, n\}$，则定义 Δ 为

$$\delta^0_{t_i}((m, \tau)) = (m + {}^\bullet t_i, (\tau - t_i)),$$
$$\delta^1_{t_i}((m, \tau)) = (m + t_i^\bullet, (\tau - t_i)).$$

直观而言， $\mathcal{A}$ 的 0-胞腔和 1-胞腔分别对应于 N 可达图中的顶点和边。高维胞腔表示多个变迁并发触发。

引理 *1*（第 *4* 条 *[3]*）：Petri 网N 的可达图与 pn2HDA(N) 的 1-截断同构： $G^1_N \simeq \mathrm{pn2HDA}(N)^{\le 1}$

原论文给出了向非匿名 HDA 的转换。非匿名 HDA 是一种更一般的设定，允许区分具有相同标签的两个并发动作。形式上，非匿名 HDA 不使用并发集，而是依赖 conclist，即由动作标记的有序事件序列。在处理 Petri 网时，并不需要这种更一般的设定；文献 [3] 通过对动作施加任意的（字典序）顺序解决了这一问题。本文采用文献 [15] 中确立的、更合适的匿名 HDA 形式化方法。

IV. 引入解释型 HDA

A. 定义

定义 *3*：解释型匿名 *HDA*（*IHDA*）由四元组 $\mathcal{A}_I = \{\mathcal{A}, I, O, \lambda_{HDA}\}$ 定义，其中 $\mathcal{A}$是一个标准 HDA，但其胞腔以输入为条件，并与一个输出相关联。形式上，我们有

- I 是原子输入命题的有限集合，
- O 是原子输出命题的有限集合，
- λ_{HDA} 为标记函数。

标记函数 λ_{HDA} 将一对立方项与每个单元 x 关联起来：一个立方项对应输入 $\lambda_{HDA}(x)[0]$，另一个对应输出 $\lambda_{HDA}(x)[1]$。

IHDA 的解释型执行呈现出输入/输出行为，与 IPN 类似，产生形式为 $w = (i_0, o_0)(i_1, o_1)(i_2, o_2) \ldots \in (\mathbb{B}^{|I|}, \mathbb{B}^{|O|})^\omega$ 的词。若存在计算 $\rho = (z_0, x_0, z_1, \ldots)$，并且该计算满足以下兼容条件，我们就称词 w 与 IHDA $\mathcal{A}_I$ 的并发步语义兼容：

我们使用两个计数器 $n_0 = 0$ 和 $n_1 = 0$ 来跟踪 w 和 ρ 中的当前位置。任何时候都必须满足 $i_{n_0} \models \lambda_{HDA}(\rho[n_1])[0]$ 和 $o_{n_0} \models \lambda_{HDA}(\rho[n_1])[1]$。也就是说，当前单元与输入兼容，并产生兼容的输出。最后，读取词 w 时，通过递增 n_0 计数器，我们可以采用以下两条规则之一：

- 只有 n_1 递增（类似于“等待步”），
- n_0 和 n_1 递增（类似于“转移步”）。

B. 从 *IPN* 到 *IHDA*

IHDA 的主要用途是以规范形式表示并发步语义的可能性。因此，我们将讨论如何将 IPN 转换为相应的 IHDA，以及这种转换一方面如何提高表达能力，另一方面如何自动识别输出赋值中的不一致之处。

构造的第一步采用第 III-B 节概述的技术，将标准 Petri 网转换为高维自动机，并附加约束：只构造输入兼容单元，也就是说，对于 $x = (m, c)$，我们有 $\bigwedge_{t \in c} \lambda_{pn}(t)[0] \not\equiv \bot$。请注意，维数为 0 或 1 的单元始终输入兼容，因此，底层 PN 的可达图与底层 IHDA 的 1-截断同构。第二步是以一致的方式，从 IPN 的标记函数推导出高维自动机的标记函数

。给定一个由其标识和并发集定义的单元 $x = (m,\ c)$，我们定义标记函数

$$\lambda_{HDA}(x) = \left(\bigwedge_{t\in c} \lambda_{pn}(t)[0],\ \bigwedge_{p\in P: m[p]>0 \vee p\in {}^\bullet c} \lambda_{pn}(p) \wedge \bigwedge_{t\in c} \lambda_{pn}(t)[1] \right),$$

其中 ${}^\bullet c$ 将前置库所扩展为并发集：${}^\bullet c = \sum_{t\in c} c(t) \times {}^\bullet t$。

由于我们只构造输入兼容单元，因此对于所有单元都有$\lambda_{HDA}(x)[0] \neq \bot$。定义完毕后，我们可以讨论第一个问题：检测输出不一致。单元的并发集对应于 IPN 中输入兼容的可并发触发变迁。也就是说，存在一个可达标识 m 和一个输入赋值，使得 c是一个有效的并发步，即 $m \geq {}^\bullet c$ 和 $\bigwedge_{t\in c} \lambda_{pn}(t)[0] \neq \bot$。$\lambda_{HDA}(x)[1]$ 表示单元生成的输出。如果存在一个单元，使得 $\lambda_{HDA}(x)[1] \equiv \bot$，这意味着不同的输出条件彼此矛盾。这种矛盾，例如要求我们的搬运车同时向左和向右行驶，代表控制架构中的故障，需要予以纠正。IHDA 能够以系统化的方式识别此类情况，并提供导致这些情况的轨迹。

C. 从 *IHDA* 到控制器

在介绍制造示例之前，我们先概述所提出的技术，以及如何从 IPN 得到实际的控制器。

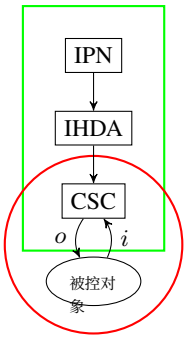

图 1：并发步控制器（CSC）的生成与使用流程。

图 1 展示了所提出方法的流程。绿色方框表示预处理或生成步骤。我们从 IPN 得到 IHDA，随后为其赋予并发步语义，以生成并发步控制器。红色圆圈圈出了物理/仿真的被控对象及控制器。箭头表示控制周期。在每个周期中，被控对象向 CSC 发送当前的输入赋值（i），CSC 随后以输出赋值（o）进行响应。

V. 制造示例

作为本文的贡献，我们在本节中给出一个小型示例，灵感源自 [12], ，用于说明如何使用高维自动机控制制造系统。

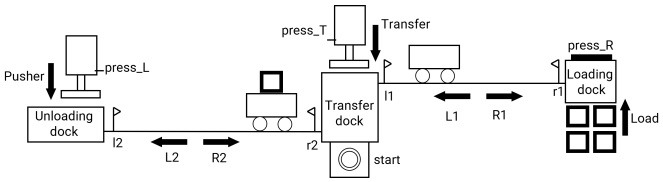

图 2：制造示例

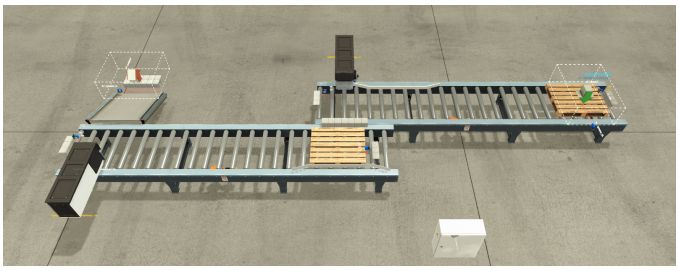

图 3：在仿真工具 Factory I/O 中设计的制造示例

A. 示例介绍

所考虑的系统如图2所示。在该系统中，转运站是核心部分，设有启动 按钮以及两辆运输小车的初始位置（传感器 *l1* 和 *r2*）。上方的小车可向右移动（执行器 **R1**）到达装货站，传感器*r1* 用于指示其所在位置。装载负载后（小车配有压力传感器 *pressR*），小车向左移动（**L1**）返回转运站。与此同时，下方的小车将负载向左运送（**L2**）至卸货站。到达*l2*位置后，由**推杆**驱动的气缸将负载推出。压力传感器*pressL*确认负载已释放，而小车则向右移动（**R2**）返回转运站。两辆小车之间的负载转移由动作**转移** 完成，并在传感器*press T* 激活时进行。

为了仿真该系统，我们在仿真工具“Factory I/O”([1], [10]) 中实现了它。图 3 展示了在该工具中设计的系统。该系统通过 Modbus TCP 和 OPCClient DA/UA 进行互连。这使我们能够用 Python 设计控制律，并通过 Modbus TCP 和 MQTT 连接输入/输出。

该系统的控制律通过解释型 Petri 网（IPN）设计，如图 4 所示。我们有意将该控制律设计得十分简单，以便更好地理解 IHDA 的使用和实现。

B. 转换为 *IHDA*

如上所述，我们首先将 IPN 的底层 Petri 网转换为高维自动机（第 III 节）。然后，添加有关输入和输出的额外信息，将高维自动机转换为与 IPN 对应的 IHDA（第 IV 节）。由此生成的 IHDA 如图5所示，其输入和输出列于表 V-B。举例来说，2-胞腔 x 对应于两辆搬运车分别同时抵达装货站和卸货站

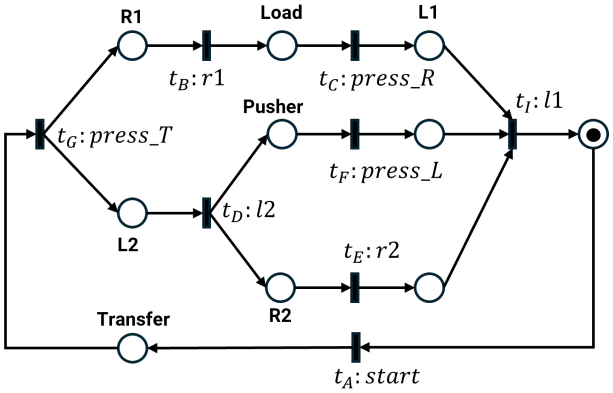

图4：制造系统的解释型 Petri 网设计的控制律

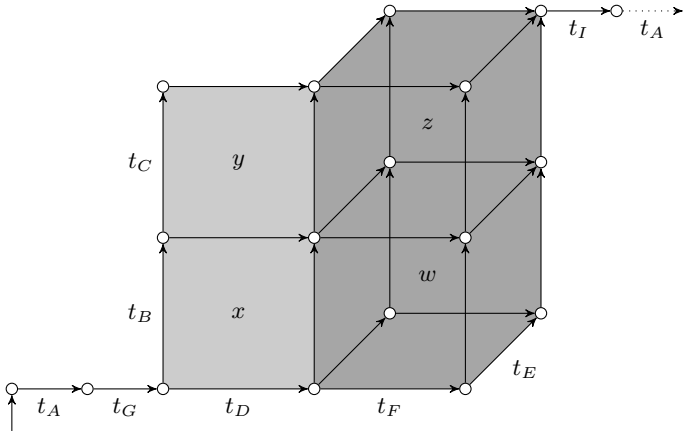

图5：与图4的解释型 Petri 网相对应的高维自动机。

，因此 t_B 和 t_D 同时发生；而 2-胞腔 y 对应于上层搬运车装载了新的负载，同时下层搬运车抵达卸货站。3-胞腔 w 对应于上层搬运车抵达装货站时，下层搬运车已返回初始位置，且推杆正将负载推向卸货站。最后，3-胞腔 z 对应于两辆搬运车返回起始位置时，推杆正在推动负载。0-胞腔的输出由 IPN 中对应的标识给出，即所有至少含有一个托肯的库所的输出条件的合取。对于 1-胞腔，输入是变迁的输入，输出由变迁之前的标识给出。对于更高维的胞腔，其输入/输出如下（为节省空间并提高可读性，我们省略了 w 和 z 的子胞腔）：

Cell	输入	输出
x	$r1 \wedge l2$	**R1** $\wedge$ **L2**
y	按下 $R \wedge l2$	**负载** $\wedge$ **L2**
w	$r1 \wedge$ 按下 $L \wedge r2$	**R1** $\wedge$ **推杆** $\wedge$ **R2**
z	按下 $R \wedge$按下 $L \wedge r2$	**负载** $\wedge$ **推杆** $\wedge$ **R2**

C. 错误检测与纠正

继续上一节的示例，我们可以说明错误检测的工作原理。假设在推杆处于活动状态时，下层小车不得移动。这意味着**L2 / R2** 和**推杆** 不能同时执行。我们可以通过将相应库所的输出标签从**L2** 更改为 **L2** $\wedge \neg$**推杆**，将**R2** 更改为 **R2** $\wedge \neg$**推杆** ，以及将 **推杆**更改为 $\neg$**L2** $\wedge$**R2** $\wedge$**推杆**，直接将这一限制添加到我们的 IPN 中。

另一种方法是在输出上定义全局不变量，即此处的 $\neg$**L2** $\vee \neg$**推杆** 和 $\neg$**R2** $\vee \neg$**推杆**。然后，在计算每个单元的标签时，我们可以检查是否满足该不变量；如果不满足，则引发错误。这种方法也很容易扩展，以涵盖涉及输入命题和输出命题的更复杂不变量。

在这两种情况下，我们都可以验证，胞腔 w 和 z（及其子胞腔）均不符合给定的限制：例如， w 的输出会变为 **R1** $\wedge$ **推杆** $\wedge$**R2** $\wedge \neg$**推杆**。这显然是不可能的，因为推杆不可能同时被激活和停用，因此在构造过程中会报出相应错误。请注意，胞腔 x 和 y 不受此更改影响。例如， x 的输出公式会变为 **R1** $\wedge$**L2** $\wedge \neg$**推杆**，这不会造成问题。

从初始胞开始并到达 w 或 z的某次运行中，我们可以推导出系统的一种执行方式，其中推杆被激活的同时，下层小车向右移动。确保推杆已停止后搬运车才再次开始移动，有多种方法。这里我们只是将它们按“顺序”排列，以避免分支。请注意，在 t_E 和t_F之间添加一个互斥库所并不足够。我们需要防止导致**R2**和 **推杆**的库所同时拥有托肯。修正后的 IPN 如图6所示，相应的 IHDA 如图7所示。

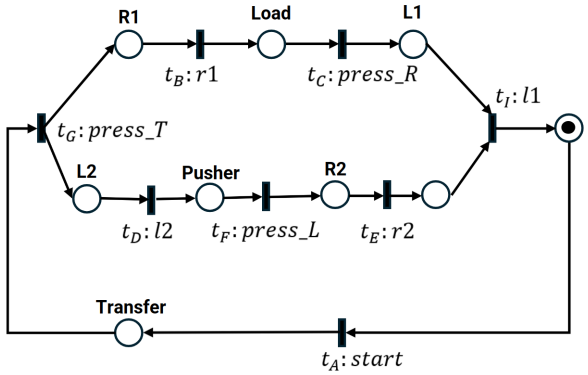

图 6：IPN 为制造系统设计的修正控制律

直观地说，原始高维自动机中每个 3-胞腔 w 和 z 都被分解为两个 2-胞腔（分别为 w'/w'' 和z'/z'' ），从而禁止**推杆**和 **R2**并发激活。

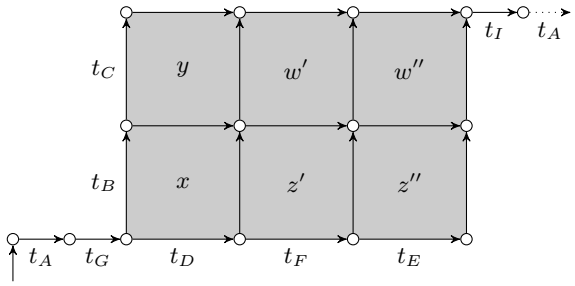

图7：与图6中的解释型 Petri 网对应的高维自动机。

D. 仿真工具中的 *IHDA* 控制器

我们实现了 IHDA，如图5所示，它作为控制器与仿真工具中建模的制造系统相连接。有关由此产生的闭环行为，以及输入和输出格式的其他信息和如何使用解释型佩特里网到控制器的转换，可参见 https://philippschlehubercaissier.github.io/pages/wodes26.html。

视频展示了系统以 1 秒的采样周期运行（输入和输出每秒更新一次）。初始化后，系统等待启动 操作，执行一个完整周期，最后返回初始配置，即一个箱子停在转运站等待。

VI. 结论

本文提出了（匿名）解释型高维自动机的数学框架，用于对具有并发性的离散事件系统进行建模。我们给出了从解释型 Petri 网到 IHDA 的转换，该转换保留了 IPN 所描述的所有可能行为，同时加入了新框架所支持的并发性。随后，我们通过一个示例实例化了这一转换，并展示了如何利用它检测系统设计中的错误。该示例已与仿真工具 Factory I/O 对接并经过测试。

本研究是将高维自动机框架应用于工业问题的初步尝试。我们有意将研究范围限定在高维自动机的并发步语义上，但还有其他更强大的语义，能够帮助处理并发系统中并发的其他方面。值得注意的是，高维自动机支持对非原子事件进行推理——动作不再必然是离散的，也可能需要耗费时间。这使我们能够更细致地分析动作如何开始、终止以及并发运行，从而也能更详细地了解受控过程。

参考文献

[1] Factory I/O - 新一代 PLC 培训。https://factoryio.com/。[2] Rajeev Alur. 《信息物理系统原理》. 麻省理工学院出版社，美国马萨诸塞州剑桥，2015年4月。

[3] Amazigh Amrane、Hugo Bazille、Uli Fahrenberg、Lo¨ıc H´elou¨et 和 Philipp Schlehuber-Caissier。Petri 网与高维自动机。载于 Elvio Gilberto Amparore 和 Lukasz Mikulski（编），《佩特里网与并发的应用和理论——第46届国际会议，PETRI NETS 2025》，法国巴黎，2025年6月22日至27日，论文集，《计算机科学讲义》第15714卷，第18–40页。施普林格，2025年。[4] Matteo Cantarelli 和 Jean-Marc Roussel。基于监督控制理论的反应式控制系统设计：可能性与局限的评估。载于2008年第9届离散事件系统国际研讨会，第200–205页，2008年5月。[5] Ren´e David 和 Hassane Alla。用于动态系统建模的 Petri 网：综述。《自动化》，30(2):175–202，1994年2月。[6] Alexandre Duret-Lutz、Etienne Renault、Maximilien Colange、Flo-rian Renkin、Alexandre Gbaguidi Aisse、Philipp Schlehuber-Caissier、Thomas Medioni、Antoine Martin、J´erˆome Dubois、Cl´ement Gillard 和 Henrich Lauko。从 Spot 2.0 到 Spot 2.10：有哪些新变化？载于 Sharon Shoham 和 Yakir Vizel（编），《计算机辅助验证——第34届国际会议，CAV 2022》，以色列海法，2022年8月7日至10日，论文集第二部分，《计算机科学讲义》第13372卷，第174–187页。施普林格，2022年。[7] Uli Fahrenberg、Christian Johansen、Georg Struth 和 Krzysztof Ziemianski。《高维自动机的语言》。《数学结构与计算机科学》，31(5):575–613，2021年。[8] Swen Jacobs、Guillermo A. P´erez、Remco Abraham、V´eronique Bruy`ere、Micha¨el Cadilhac、Maximilien Colange、Charly Delfosse、Tom van Dijk、Alexandre Duret-Lutz、Peter Faymonville、Bernd Finkbeiner、Ayrat Khalimov、Felix Klein、Michael Luttenberger、Klara J. Meyer、Thibaud Michaud、Adrien Pommellet、Florian Renkin、Philipp Schlehuber-Caissier、Mouhammad Sakr、Salomon Sickert、Ga¨etan Staquet、Cl´ement Tamines、Leander Tentrup 和 Adam Walker。反应式综合竞赛（SYNTCOMP）：2018—2021年。《国际软件工具与技术转移杂志》，26(5):551–567，2024年。[9] Jan Kret´ınsk´y、Tobias Meggendorfer 和 Salomon Sickert。Owl：用于 ω-词、自动机和线性时序逻辑的库。载于 Shuvendu K. Lahiri 和 Chao Wang（编），《验证与分析自动化技术——第16届国际研讨会，ATVA 2018》，美国加利福尼亚州洛杉矶，2018年10月7日至10日，论文集，《计算机科学讲义》第11138卷，第543–550页。施普林格，2018年。[10] B. Riera 和 B. Vig´ario。HOME I/O 和 FACTORY I/O：用于控制教育的虚拟住宅和虚拟工厂。《IFAC在线论文集》，50(1):9144–9149，2017年7月。[11] Jean-Marc Roussel 和 Jean-Jacques Lesage。尽管规范存在不一致，仍可对逻辑控制器进行代数综合。载于第11届离散事件系统国际研讨会，WODES 2012，第307–314页，墨西哥瓜达拉哈拉，2012年10月。[12] Jeremie Saives、Gregory Faraut 和 Jean-Jacques Lesage。并发离散事件系统的自动划分，用于分布式行为识别。《IEEE自动化科学与工程汇刊》，15(2):832–841，2018年。[13] H. Song、D.B. Rawat、S. Jeschke 和 C. Brecher。《信息物理系统：基础、原理与应用》。《信息物理系统：基础、原理与应用》。Morgan Kaufmann，2016年。[14] Rob J. van Glabbeek。论高维自动机的表达能力。《理论计算机科学》，356(3):265–290，2006年。[15] Krzysztof Ziemia´nski。内部报告。未发表。