利用多面体约简求解Petri网可达性问题

Nicolas Amat[1], Silvano Dal Zilio[1], 和 Didier Le Botlan[1]

[1]LAAS-CNRS，图卢兹大学，法国国家科学研究中心，图卢兹，法国

arXiv:2302.02686v1 [cs.LO] 6 Feb 2023

摘要

我们提出了一种新方法，利用结构约简加速对Petri网可达性性质的验证。我们的方法依赖于一种称为多面体抽象的状态空间抽象，它结合了结构约简与库所标识之间的线性算术约束集合。我们提出了一种名为令牌流图（TFG）的新数据结构，用以刻画多面体抽象中约束所具有的特殊结构。我们利用TFG高效求解两个可达性问题：首先，检查给定标识是否可达；其次，计算网的并发关系，即所有能够在某个可达标识中同时被标识的库所对。我们的算法已在名为Kong的工具中实现，并在模型检验竞赛2020年届使用的一大批模型上进行了评估。实验表明，即使只应用了适量的约简，该方法也表现良好。

关键词—— Petri网，结构约简，可达性，并发库所

1 引言

我们提出了一种新方法，利用结构约简来加速对佩特里网可达性属性的验证。简而言之，我们从初始带标识网 (N_2, m_2)出发，计算出一个约简网 (N_1, m_1)，并仅通过探索约简网的状态空间来证明关于初始网的性质。与以往结构约简方面的工作（例如 [5], ）不同，我们的方法并非针对某一特定类别的性质量身定制——例如无死锁性质——而是可以应用于更一般的问题。

为展示该方法的通用性，我们将其应用于两个具体问题：首先，检查给定标识的可达性；其次，计算网的并发关系，即所有能在某个可达标识中同时带有标识的库所对。

在理论方面，我们方法的正确性依赖于一种新的状态空间抽象方法，我们在 [1, 2], 中将其称为多面体抽象。该方法涉及一组线性算术约束，用于刻画初始网与约简网中库所标识之间的关系。其思想是定义形如 $(N_1, m_1) \triangleright_E (N_2, m_2)$ 的关系，其中 E 是一个线性方程组，用于关联 N_1 和 N_2 的可能标识。更确切地说，目标是在 E 中保留足够的信息，使我们只需知道 N_2 的可达标识，就能重建 N_1 的可达标识。

在实践中，我们通过从初始网逐步计算结构约简来推导多面体抽象。在这种情况下，我们称之为计算多面体约简。尽管在对 Petri网进行模型检验时，结构约简有诸多益处的例子，但利用方程组（E）追溯约简的影响尚属新颖，我们希望这种方法也能应用于其他问题。

我们的算法依赖一种名为 令牌流图 (TFG) 的新型数据结构，该结构能够捕捉线性系统 E中约束所具有的特定结构 [3], 。我们介绍 TFG，并说明如何利用这种数据结构，加速求解前文提到的两个可达性问题：(1) 标识可达性和 (2) 并发关系。我们使用“加速”一词，以强调 TFG 的“乘法效应”。具体而言，我们提出一个框架：从用于求解问题 (1) 或 (2) 的工具出发，构建该工具的增强版本，使其能够利用约简。对于某个初始实例，例如某个网 N上的实例，增强后的工具可以通过在其化简版本 N上求解，再重构出初始实例的正确解。两种情况下，我们的方法都采用“逆变换”的形式，仅依赖 E ，且无需对约简网进行昂贵的预处理。

对于标识可达性问题，我们通过扩展工具 Sift 来说明我们的方法。Sift 是一种针对佩特里网的显式状态模型检查器，可在线检查可达性属性。对于并发关系，我们扩展了工具 cæsar.bdd；它是 CADP 工具箱的一部分， [9, 18], 使用 BDD技术探索网的状态空间并查找并发库所。

我们证明，我们的方法能够带来显著的加速比，因为约简网的库所数量可能远少于初始网，而且库所数量往往是影响可达性问题复杂度的主要参数。

概要与贡献介绍一些相关工作后，我们将在第3节定义佩特里网的语义以及并发库所的概念。我们将在第4节定义一种简化的“可达性等价”概念，称为多面体抽象。第5节和第6 节介绍我们的主要贡献。我们将在第5节描述令牌流图（TFG），并在第6节证明若干相关结果。这些结果使我们能够通过在令牌流图的节点上进行令牌游戏，推理网中可达的库所。我们将在第7节利用令牌流图为可达性问题定义一种判定过程。接着，我们将在第8节定义一种用于查找并发库所的类似算法，并说明如何将其应用于仅掌握剩余并发关系部分知识的情形。

我们的方法已经实现，计算实验表明，约简对大量模型都十分有效（第9节）。我们的基准测试来自一个独立管理的佩特里网集合，这些网对应于 2020年届模型检验竞赛期间所使用的网 [4]。我们观察到，即使只进行适度的约简（例如可以移除25% 的库所），使用约简计算完整结果也会比不使用约简快得多；通常可提升几个数量级。我们还表明，在关系不完整的情况下，我们同样表现良好，既更快也更准确。

许多结果和定义已经在 [3]中介绍过。本扩展版本包含若干新增内容。首先，我们将基于多面体约简和 TFG 的方法扩展到可达性问题，而 [3] 仅涉及并发关系的计算。针对这一问题，我们为所有关于 TFG 的结果给出了详细证明，从而阐明线性系统 E 的解与网的可达标识之间的密切联系。我们的论文还给出了新公理的定义和证明，这些公理在计算不完整并发关系时非常有用；也就是说，此时我们只掌握关于并发库所的部分知识。最后，我们提供了更多关于工具性能的实验结果。

2 相关工作

我们考察了与本文所研究的两个问题相关的工作，尤其关注并发库所，因为这方面给出了最具原创性的结果。我们还简要讨论了结构约简在模型检验中的应用。

2.1 标识可达性

佩特里网的可达性是一个重要且困难的问题，具有许多实际应用。在本文中，我们考虑如下简单问题：从初始标记 m_1 出发，检查给定标记m_1' 是否可以通过在网 N_1 中触发一系列变迁而达到。

在先前的工作中 [1, 2], , 我们使用多面体抽象和符号模型检验来增强对“广义”可达性性质的验证，即检查是否可以达到满足某个性质 ϕ 的标记；该性质表示为库所之间线性约束的布尔组合，例如 $(p_0 + p_1 = p_2 + 1) \wedge (p_0 \leqslant p_2)$。

这个更一般的问题对应于模型检验竞赛（MCC） [4], 中的一项考题。MCC是面向佩特里网模型检验器的年度竞赛。在此背景下，人们采用了许多优化技术：符号化技术，例如 k-归纳法；用于佩特里网的标准抽象技术，如顽固集和偏序约简；使用“状态方程”；将问题缩减为整数线性规划问题；等等。

假设 (N_2, m_2) 是 (N_1, m_1) 的多面体约简，相关联的方程组为 E。 [1, 2] 的主要结果是，可以构造一个公式 ϕ_E，使得 ϕ 在 N_1 中可达，当且仅当 ϕ_E 在 N_2中可达。这意味着，只要某个模型检验器能够处理广义可达性性质，我们就可以轻松扩展它，使其免费受益于结构约简。

本文使用 TFG 来证明关于标识可达性问题的一个更强性质（见第 7 节）：给定 m_1' 的目标标识 N_1，我们能够有效地计算出 m_2' 的一个标识 N_2，使得 m_1' 在 N_1 中可达，当且仅当 m_2' 在 N_2 中可达。与我们先前的方法相比，这种方法可能效率更高，因为在实践中，变换后的性质 ϕ_E可能相当复杂，而标识可达性性质只是等式约束的简单合取。例如，我们只需使用基本的显式状态模型检查器就能开展实验。

这种多面体约简的应用虽然不如我们关于并发关系的结果那样新颖，但凸显了 TFG 能够提供一种有效的方法来利用约简。它也证明了我们的方法具有多样性。

2.2 并发关系

我们工作的主要成果是提出了一种计算佩特里网并发关系的新方法。这个问题具有实际应用价值，例如可用于将佩特里网分解为并发进程的乘积 [10, 11]。它还提供了一个有趣的安全性示例，巧妙地扩展了死库所的概念；死库所是指在执行过程中永远无法到达的库所。这些问题带来了艰巨的技术挑战，也为测试和改进新的模型检验技术 [12]提供了机会。

自然，可以通过逐一检查每对库所的可达性来计算并发关系。但这相当于求解数量为二次阶的可覆盖性问题——其参数是网中的库所数量——因此，人们会期望找到更巧妙的解法，即便这些解法只适用于某些特定情况。我们也对部分解感兴趣，因为计算整个状态空间并不可行。

多项研究探讨了如何找出或刻画佩特里网中的并发库所。这一概念有多种称谓，例如由标识定义的共存性 [19], 并发图 [28] 或并发关系 [13, 20, 21, 25, 29]。其主要动机在于，并发关系刻画了网中能够同时处于活动状态的子部分。因此，在将网分解为一组独立组件时，它发挥着重要作用。例如， [29], 中的作者将并发库所与“顺序模块”（状态机）的存在联系起来。另一个例子是将网分解为单元安全嵌套单元佩特里网（NUPN） [10, 11], , 其中并发关系的计算是主要瓶颈之一。

我们只知道少数几个支持计算并发关系的工具。近期的一款工具是 Hippo 平台的一部分，可在线获取 [29], 。本文采用的参考工具是 CADP 工具箱中的 cæsar.bdd [9, 18]。它支持计算部分关系，并可使用一种特定的压缩文本格式输出网的“并发矩阵” [12]。我们采用相同的格式，因为我们使用 cæsar.bdd 计算残余网 N_2 的并发关系，并将其作为基准测试中的衡量标准。

2.3 使用约简进行模型检验

关于我们对结构约简的运用，我们的主要结果可被视为约简定理的一个实例：该定理允许根据初始模型（N_1）的性质，推导出更简单、粒度更粗的版本（N_2）的性质。 [23],但我们的约简概念更为复杂，对应于 Berthelot [5], 首创的约简概念，只是增加了线性方程。

有几种工具使用约简来检查可达性属性，但没有一种专门用于计算并发关系。我们可以提到 Tapaal [8], , 这是一款显式状态模型检查器，结合了偏序约简技术和结构约简；或者较新的 ITS-Tools [27], , 它结合了多种技术，包括结构约简以及 SAT 和 SMT 求解器。

在我们的工作中，我们关注能够保留可达状态的约简，并使用“归约方程”来保存初始网与约简后的网之间的可追溯信息。我们的工作是一个三部曲的一部分。

我们的方法最初用于 模型计数 [6, 7], , 以便高效计算可达状态的数量。该方法在名为 Tedd 的符号模型检查器中实现，而 Tedd 是 Tina 工具箱 [22] 的一部分。它还依赖一个名为 Reduce 的辅助工具，该工具会应用结构约简并返回归约方程集合。我们在实验中沿用了这一工具。

我们三部曲的第二部分 [1, 2] 提出了一种方法，将网约简与基于 SMT 的模型检查器结合起来加以利用，并由此开发出一款专用工具 SMPT。这项工作引入了多面体抽象的概念。这里的主要目标是为我们的方法提供一个形式化的

框架，其形式是定义 Petri网之间一种新的语义等价。

最后，本文是 [3], 的扩展版本，后者引入了令牌流图这一概念，并介绍了一种新的应用，即加速并发库所的计算。我们的目标是提供能够利用多面体抽象概念的有效算法。从某种意义上说，本文是 [1, 2]中所发展的理论的实用或算法层面的对应物。这项工作还与一个名为Kong的新工具相关，我们将在第9节介绍该工具。

3 佩特里网

读者应对佩特里网有一定了解。下面回顾一些基本术语。全文中，比较（$=$，$\geqslant$）和算术运算（$-$，$+$）均逐点扩展到函数和元组。

定义 3.1 (佩特里网)。佩特里网 N 是一个元组(P, T, 前置, 后置)，其中：

- $P = \{p_1, \ldots, p_n\}$ 是一个有限的库所集合，
- $T = \{t_1, \ldots, t_k\}$ 是一个有限的变迁集合（与 P 不相交），
- Pre：$T \to (P \to \mathbb{N})$ 和Post：$T \to (P \to \mathbb{N})$ 是前置条件函数和后置条件函数（也称为 N 的流函数）。

我们通常直接写作：当 $p \in P$ 时，p 是 N 的一个库所。网的一个状态 m（也称为标识）是一个全映射 $m: P \to \mathbb{N}$，它为 N 的每个库所分配一定数量的令牌 $m(p)$。带标识网 (N, m_0) 是由一个网及其初始标识 m_0 组成的有序对。

当 $m(p) \geqslant \mathrm{Pre}(t, p)$ 对 P 中所有库所 p 都成立时，变迁 $t \in T$ 在标识 $m \in \mathbb{N}^P$ 下使能。（我们也可以简写为 $m \geqslant \mathrm{Pre}(t)$，其中 $\geqslant$ 表示对标识逐分量进行比较。）若通过发射变迁 t，标识 m' 可从标识 m 到达，记作 $m \xrightarrow{t} m'$，条件为：（1）变迁 t 在 m 下使能；且（2）$m' = m - \mathrm{Pre}(t) + \mathrm{Post}(t)$。当不关心变迁的具体身份时，我们将这一关系简记为 $m \to m'$。更一般地，若存在一个（可能为空的）缩减序列使得 $m \to \cdots \to m'$，则称标识 m' 在 N 中可从 m 到达，记作 $m \to^\star m'$。我们用 $R(N, m_0)$ 表示 N 中从 m_0 可达的标识集合。

当每个库所中的令牌数至多为k 时，标识 m 是 k-有界的；带标识网（N，m_0）是有界的，是指存在一个常数 k，使得所有可达标识都是 k-有界的。虽然我们的许多结果适用于一般情形——即网不一定有界，且对流函数（弧的权重）没有任何限制——但我们的工具以及关于并发关系的实验，聚焦于1有界网，也称为安全网。

给定一个带标识网 (N, m_0)，我们称库所 p、q 是并发的，如果存在一个可达标识 m，使得 p 和 q 均被标识。并发库所问题是枚举所有此类库所对。

定义 3.2（死库所与并发库所）。如果存在 m 属于 $R(N, m_0)$，使得 $m(p) > 0$，则称网（N，m_0）中的库所 p 是非死的。同理，如果存在 m 属于 $R(N, m_0)$，使得$m(p) > 0$ 且 $m(q) > 0$，则称库所 p、q 并发，记作 $p \parallel q$。扩展而言，当 p 非死时，我们使用记号$p \parallel p$。如果 p、q 不并发，则称它们非并发，记作 $p \# q$。

与线性算术约束的关系 佩特里网理论中的许多结果都基于其与线性代数和线性规划技术的关系 [24, 26]。一个著名的例子是，网的潜在可达标识（对可达标识的过近似）

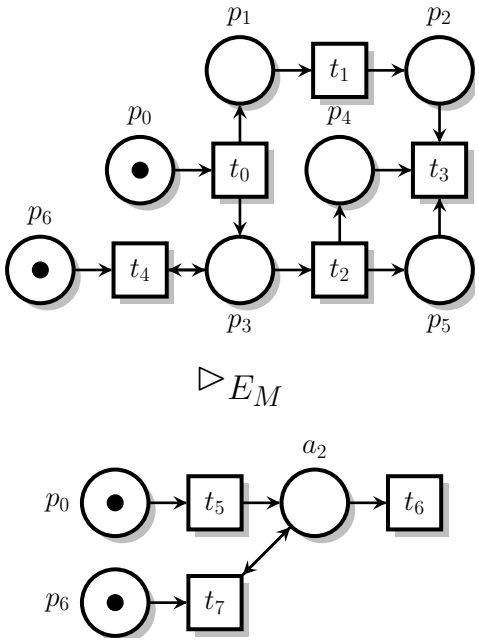


图 1：一个佩特里网示例，M_1（上），以及它的一个多面体抽象，M_2（下），其中包含 $E_M \triangleq (p_5 = p_4)$，$(a_1 = p_1 + p_2)$，$(a_2 = p_3 + p_4)$，$(a_1 = a_2)$。

(N, m_0) 是状态方程问题 $m = I \cdot \sigma + m_0$ 的非负整数解，其中 I 是由 N 的流函数定义的整数矩阵，称为关联矩阵；σ 是 $\mathbb{N}^k$ 中的一个向量。已知，线性方程组 $\sigma^T \cdot I = \vec{0}$ 的解可得出库所不变量 $\sigma^T \cdot m = \sigma^T \cdot m_0$，这些不变量能够提供有关将网分解为非并发库所块的信息，进而提供有关并发关系的信息。

例如，对于网 M_1（图 1），我们可以计算出不变量 $p_4 - p_5 = 0$。只要能够证明库所 p_4 和 p_5中至少有一个是非死的，就足以证明它们是并发的。同样，形如 $p + q = 1$ 的不变量足以证明 p 和 q 是 1-有界的，且不能并发。不幸的是，不变量只能对可达标识集合提供过度近似，而且要判断一个网是否属于已知的少数几类网之一（在这些网中，可达标识集合等于潜在可达标识集合），可能并不容易 [17]。

我们的方法与这类推理有一些相似之处。一个主要区别是，我们将使用方程组来建立两个网的可达标识之间的关系，而不是用来表达单个网内（可能）可达标识所满足的约束。与不变量类似，在许多情况下，这种方法将使我们无需“触发任何变迁”，也就是无需探索状态空间，就能获取有关并发关系的信息。

下文中，我们将经常把库所名称用作变量，并把标识 $m: P \to \mathbb{N}$ 视为一组线性方程的部分解。为简单起见，我们所有的方程都将采用 $x = y_1 + \cdots + y_l$ 或 $y_1 + \cdots + y_l = k$ 的形式（其中 k 是 $\mathbb{N}$中的一个常数）。

给定线性方程组 E，我们用 $fv(E)$ 表示其所有变量的集合。我们只关注 E 的非负整数解。因此，在本例中，E 的解是从 $fv(E)$ 中的变量到 $\mathbb{N}$ 的全映射，且满足 E 中的所有方程。当至少存在一个这样的解时，我们称 E 是相容的。根据这些定义，若方程组 E，$\lfloor m \rfloor$ 相容，其中 $\lfloor m \rfloor$ 是方程序列 $p_1 = m\ p_1 \quad p_n = m$ ( )，$\ldots$，(p_n)，则称映射 $m: \{p_1, \ldots, p_n\} \to \mathbb{N}$ 是 E 的（部分）解。（从某种意义上说，我们将 $\lfloor m \rfloor$ 用作代换。）例如，若方程组 $p = 1 + x$，$q = 1 + y$，$\lfloor m \rfloor$ 相容，则库所 p，q 并发，其中 m 是一个可达标识，并且

x，y 是一些新引入的（松弛）变量。

给定两个标识 $m_1 : P_1 \to \mathbb{N}$ 和 $m_2 : P_2 \to \mathbb{N}$，它们可能来自不同的网，我们称 m_1 和 m_2 是相容的，记作 $m_1 \equiv m_2$，当且仅当它们在共享库所上的标识相同：$m_1(p) = m_2(p)$ 对所有 p 属于 $P_1 \cap P_2$ 均成立。这是方程组 $\lfloor m_1 \rfloor, \lfloor m_2 \rfloor$ 相容的充要条件。

4 多面体抽象

我们最近定义了一种多面体抽象 [1, 2]，其基础是我们先前将结构约简应用于模型计数的工作 [6, 7]。这里我们只需使用这一概念的简化版本，它蕴含两个网的状态空间之间存在一种等价关系，(N_1, m_1) 与 (N_2, m_2) 在一个线性 E 方程组意义下的等价。

定义 4.1 (E-等价)。我们称 (N_1, m_1) 与 (N_2, m_2) 是 E-等价的，记作 $(N_1, m_1) \rhd_E (N_2, m_2)$，当且仅当：

(A1) 对于所有标识 m，E、$\lfloor m \rfloor$ 在 $R(N_1, m_1)$ 或 $R(N_2, m_2)$ 中相容；*(A2)* 初始标识彼此兼容，即 E、$\lfloor m_1 \rfloor$、$\lfloor m_2 \rfloor$ 相容；*(A3)* 假设 m_1'、m_2' 分别是 N_1、N_2 的标识，且 E、$\lfloor m_1' \rfloor$、$\lfloor m_2' \rfloor$ 相容，则 m_1' 可达，当且仅当 m_2' 可达：$m_1' \in R(N_1, m_1) \iff m_2' \in R(N_2, m_2)$。

根据定义，关系 $\rhd_E$ 是对称的。我们特意使用一个从左向右的符号，以强调 N_2 应当是 N_1 的化简版本。特别地，我们期望 N_2 中的库所少于 N_1 中的库所。

给定关系 $(N_1, m_1) \rhd_E (N_2, m_2)$，$N_2$ 中的每个 m_2' 可达标识都可以与 N_1 中可达标识的一个唯一子集相关联；该子集由 E，$\lfloor m_2' \rfloor$ 的解（根据条件 (A1) 和 (A3)）定义。我们可以证明，[1, 2] 这会将 (N_1, m_1) 的可达标识划分为“凸集”——因此得名多面体抽象——每个凸集都与 N_2 中的一个可达标识相关联。当 N_2 的状态空间相比 N_1 的状态空间非常小时，我们的方法尤其有用。在极端情况下，我们甚至可以找到 N_2 为“空”网的例子（即没有库所、因而只有唯一标识的网），但这一条件并非我们方法的必要条件。

我们可以用图 1 中的两个带标识网 M_1, M_2 来说明这一结果，并可证明 $M_1 \rhd_{E_M} M_2$（详见 [1, 2]）。我们知道，$m_2' \triangleq (p_0 = 0, p_6 = 1, a_2 = 1)$ 在 M_2 中可达，这意味着方程组 $p_0 = 0, p_6 = 1, p_4 = p_5, p_1 + p_2 = 1, p_3 + p_4 = 1$ 的每个解都会给出 M_1 的一个可达标识。此外，每个满足 $p_i \geqslant 1$ 和 $p_j \geqslant 1$ 的解都能作为 $p_i \parallel p_j$ 的见证。例如，p_1, p_4, p_5 和 p_6 肯定彼此并发。我们应该

利用这样一个事实：在对 E 作出某些假设的情况下，我们可以找到所有这类“变量对”，而无需显式求解形如 $E, \lfloor m \rfloor$ 的方程组；只需考察 E 的结构即可。

在当前工作中，为了描述我们的方法，无须解释如何推导或检查某个等价陈述是否正确。实际操作中，我们从一个初始网开始，(N_1, m_1)，并通过结合若干结构化简规则推导出 (N_2, m_2) 和 E。我们的规则集的精确定义见 [7]，而这些约简所得结果总能导出有效 E-等价的证明见 [1, 2]。通过这一过程得到的线性方程组呈现出类图结构。在下一节中，我们将介绍一组约束，以形式化地描述这一观察。这是本文的贡献之一，因为我们在以往的工作中从未定义过等价的内容。基准测试（第 9 节）表明，这些约束具有足够的通用性，能够在大量模型上取得良好结果。

5 令牌流图

我们针对等价陈述 $(N_1, m_1) \rhd_E (N_2, m_2)$ 中出现的方程，提出一组结构约束。我们的目标是定义一种算法，利用 E 中方程的结构，根据 N_2 的并发关系，轻松计算有关 N_1 的并发关系的信息。

我们将方程组 E 的令牌流图（TFG）定义为一个有向无环图（DAG），其中每个在 E 中出现的变量对应一个顶点。TFG 中的弧用于表示由 E 中的方程所导出的关系。我们考虑两类弧。冗余方程的弧 $p = q$ $p = q + r + \ldots$ p 用于表示形如（或）的方程，表明库所的标识可以根据 q、r、..... 的标识重构。在这种情况下，我们说库所 p 被弧 $q \to\!\bullet\, p$ 移除，因为库所 q 的标识可能影响库所 p 的标识，但反过来则未必成立。图 2 展示了给定网 M_1 的一个子部分上的此类化简规则，该网见图 1。在这种情况下，库所 p_4 与 p_5 具有相同的前置与后置关系，因此两个库所都是冗余的。于是，移除库所 p_5 后，我们得到右侧的 TFG，它对应于方程 $p_5 = p_4$（用“黑点”弧表示）。

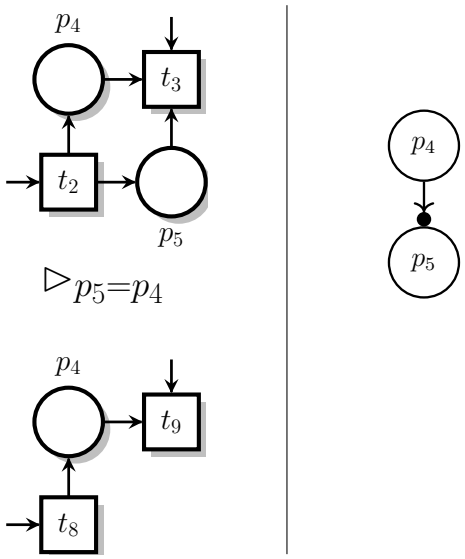


图 2：对网的一个子部分应用冗余约简 M_1（见图 1 左图）及其对应的 TFG（右图）。

第二类弧 $a \circ\!\to p$ 用于聚合方程。它表示形如 $a = p + q$ 的方程，这些方程是在将若干库所聚合为一个新库所时生成的。在这种情况下，我们预期：如果能够到达一个在 a 中有 k 个令牌的标识，那么当 $k = k_1 + k_2$ 时，必定能够到达一个在 p 中有 k_1 个令牌且在 q 中有 k_2 个令牌的标识（参见引理 6.2 中的“聚合”性质）。因此，可以根据 a 的标识重建 p 和 q 的标识。于是，我们称库所 p 和 q 被移除。我们还称节点 a 被插入；它在 N_1 中并不存在，但可能会作为一个新库所出现在 N_2 中，除非它被后续的缩减移除。一次聚合可以涉及两个以上的库所。图 3 展示了一个此类缩减的例子：在图 1 的网 M_1 中，将库所 p_1、p_2 聚合到一个新库所 a_1。因此，图 3 右侧的 TFG 描绘了所得方程 $a_1 = p_1 + p_2$（用“白点”弧建模）。

TFG 也可以包含表示常量的节点，用于表达关于形如 $p + q = k$ 的标识的不变式陈述。为此，我们假设存在一族互不相交的集合 $K(n)$（它们也与库所名和变量名互不相交），对每个 n 属于 $\mathbb{N}$ 的情形均如此，使得节点 $v \in K(n)$ 的“赋值”始终为 n。我们用 K 表示所有常量组成的集合。我们可以

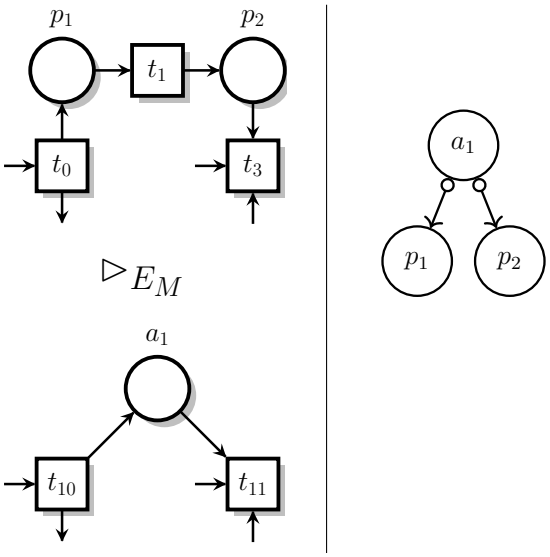


图 3：将聚合化简应用于网的一部分 M_1，来自图 1（左）及其对应的 TFG（右）。

将值为 n的常量节点写作 v^n（而不只是 v）。

定义 5.1（令牌流图）。一个具有库所集合 P 的 *TFG* 是一个有向（多重）图 (V, R, A)，满足：

- $V = P \cup S$ 是一个顶点（或节点）集合，$S \subset K$ 是一个有限常量集合，
- $R \in V \times V$是一个冗余弧集合，$v \to\!\!\bullet\, v'$，
- $A \in V \times V$是一个聚合弧集合，$v \circ\!\!\to v'$，且与 R 不相交。

我们的方法中复杂度的主要来源，是需要管理 A 与 R 弧之间的相互依赖关系，也就是冗余与聚合相结合的情况。仅通过查看 E 中的方程并不能轻易实现这一点，因此需要一种专用的数据结构。

我们定义一些在下文中会用到的记号。当在 R 中有 $(v \to\!\!\bullet\, v')$ 或在 A 中有 $(v \circ\!\!\to v')$ 时，我们使用记号 $v \to v'$。如果节点 v 不是任何弧的目标节点，我们就称其为 根节点。如果它没有形如 $(v \circ\!\!\to v')$ 的输出弧，则称其为 $\circ$-叶节点。V^n 中的节点序列 $(v_1, \ldots, v_n)$ 是一条 路径，当且仅当对所有 $1 \leqslant i < n$，都有 $v_i \to v_{i+1}$。如果图中存在一条从 v 到 v' 的路径，或者 $v = v'$，我们就使用记号 $v \to^\star v'$。当 X 是 V 的最大子集 $\{v_1, \ldots, v_k\}$，且满足 $X \neq \emptyset$，并且对所有 $i \in 1..k$ 都有 $v \circ\!\!\to v_i \in A$ 时，我们记为 $v \circ\!\!\to X$。类似地，当 X 是最大的非空节点集合 $\{v_1, \ldots, v_k\}$，且对所有 $i \in 1..k$ 都有 $v_i \to\!\!\bullet\, v \in R$ 时，我们记为 $X \to\!\!\bullet\, v$。最后，记号 $\downarrow v$ 表示 v 的后继节点集合，即：$\downarrow v \triangleq \{v' \in V \mid v \to^\star v'\}$。

我们在图 4 中展示了一个令牌流图的示例，其中“黑点”弧表示 R 中的边，“白点”弧表示 A中的边。其思路是，每个关系 $X \to\!\!\bullet\, v$ 或$v \circ\!\!\to X$ 都对应 E中的一个方程 $v = \sum_{v_i \in X} v_i$，并且 E 中的所有方程都应反映在 TFG 中。我们希望避免同一个库所被删除多次，或某个库所出现在 TFG 中，却从未在 N_1、N_2 或 E中提及。我们还有与 N_2中的库所相匹配的根节点（不含最终的常量节点），以及与 N_1中的库所相匹配的 $\circ\!\!\to$-叶节点。所有这些约束都可以用适当的良构图概念来表达。

定义 5.2 (良构 TFG**)**。等价陈述 $(N_1, m_1) \triangleright_E (N_2, m_2)$ 的 *TFG* $G = (V, R, A)$是良构的，当且仅当满足以下所有约束，其中 P_1和 P_2 分别表示 N_1 和 N_2中的库所集合：

(T1) 没有未使用名称：$V \setminus K = P_1 \cup P_2 \cup fv(E)$；**(T2)** K 中的节点是根节点：若 $v \in V \cap K$，则 v 是 G的根节点；**(T3)** 节点只能被移除一次

：不可能出现 $p \circ\!\!\to q$ 和 $p' \to q$ 同时满足 $p \neq p'$，也不可能同时出现 $p \to\!\!\bullet\, q$和 $p \circ\!\!\to q$；**(T4)** 包含且仅包含 E中的所有方程：当且仅当方程 $v = \sum_{v_i \in X} v_i$ 属于E时，$v \circ\!\!\to X$或 $X \to\!\!\bullet\, v$ 成立；**(T5)** G 是无环的；**(T6)** G 中的节点与网相匹配：G的根节点集合（不含常量 K）等于集合 P_2。G的 $\circ$-叶节点集合（不含常量 K）等于集合 P_1。

给定一个关系 $(N_1, m_1) \triangleright_E (N_2, m_2)$，良态性条件足以确保 TFG 的唯一性（除常量节点的选择外），当我们将每个方程设为属于 A 或属于 R时。在此情况下，我们将该 TFG 记为 $[\![E]\!]$。实际中，我们使用名为 Reduce 的工具，从初始网 E 生成 N_1, m_1)-等价关系。该工具输出一系列适合构建 TFG 的方程，并为每个方程添加一个标签，指示其属于冗余还是聚合。我们在图 4中展示了由 Reduce 为图 M_1 所示的网生成的方程。1。

```
# R |- p5 = p4
# A |- a1 = p2 + p1
# A |- a2 = p4 + p3
# R |- a1 = a2
```

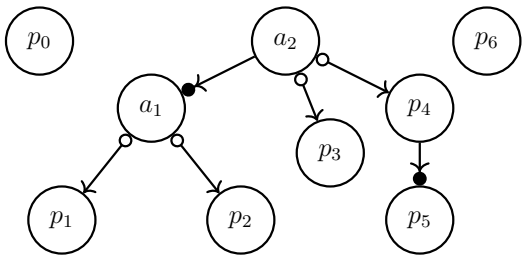


图 4：从图 M_1中的网生成的方程及其对应的 TFG。图 1。

约束 (T1)–(T6) 并非人为设定或任意规定的。实际中，我们通过多步结构约简来计算 E-等价关系，而 TFG 准确记录了这些约简过程中生成的约束和信息。从某种意义上说，方程 E 抽象了两个网的语义之间的关系，而 TFG 则记录了库所之间约简的结构。

6 语义

根据构造，“归约方程组” E与其关联图$[\![E]\!]$之间存在紧密联系。我们证明，E 的解与图的“赋值”（下文称为配置）之间也存在类似关系。

TFG（V，R，A）的配置 c是一个从 V到 $\mathbb{N}$的偏函数。当 c在 v上未定义时，我们使用记号 $c(v) = \bot$；并且，当 v 是 $K(n)$中的常量节点时，我们总是假设 $c(v) = n$。

当对于 V 中的所有节点 v，$c(v)$ 都有定义时，配置 c 是全的；否则称其为部分的。我们使用记号 $c_{\restriction N}$ 表示将 c 的定义域限制为网 N中的库所集合后得到的配置。注意，当 c在 N 的所有库所上都有定义时，$c_{\restriction N}$ 可以视为一个

标识。与标识类似，我们称两个配置 c和 c' 是相容的，记作 $c \equiv c'$，如果它们在二者都有定义的节点上取值相同：当 $c(v) \neq \perp$ 且 $c'(v) \neq \perp$时，$c(p) = c'(p)$。（将配置与标识进行比较时同样如此。）我们还使用 $\lfloor c \rfloor$ 表示方程组 $v_1 = c(v_1)$, $\cdots$, $v_k = c(v_k)$，其中 $(v_i)_{i\in 1..k}$ 是满足 $c(v_i) \neq \perp$的节点。若一个配置 c的节点赋值与 $[\![E]\!]$的方程一致，则称该配置为良定义。

定义 6.1 (良定义配置)。 当配置 c对所有节点 p 均满足以下两个条件时，称其为良定义配置：

(CBot) c **</style}若，则** $(v) = \perp$ 当且仅当 $c(w) \neq \perp$；**(CEq)** 若 $\sum_{v_i \in X} c\ v_i$ $() \neq \perp$ 且 $v \circ\!\!\rightarrow X$ 或 $X \rightarrow\!\!\bullet\ v$，则 $()$ $()$。

我们证明，TFG $[\![E]\!]$ 的良定义配置是 E的部分解，反之亦然。因此，由于 E 中的所有变量都是 TFG 的节点（条件 (T1)），我们可以得出 E 的解与 $[\![E]\!]$的全配置、良定义配置之间存在等价关系。

引理 6.1 (良定义配置是解)。设 $[\![E]\!]$ 是等价关系 $(N_1, m_1) \rhd_E (N_2, m_2)$的良构 *TFG*。若 c 是 $[\![E]\!]$ 的良定义配置，则 E，$\lfloor c \rfloor$ 相容。反过来，若 c 是 $[\![E]\!]$ 的全配置，且 E，$\lfloor c \rfloor$ 相容，则 c 也是良定义的。

证明： 我们分别证明每个性质。

假设 c 是 $[\![E]\!]$的良定义配置。由于E是一个归约方程组，它是由一系列等式 ϕ_1, ..., ϕ_k 组成，其中每个方程 ϕ_i 的形式为$x_i = y_1 + \cdots + y_n$。此外，由于 $[\![E]\!]$ 是良构的，我们有 $X_i \rightarrow\!\!\bullet\ v_i$ 或 $x_i \circ\!\!\rightarrow X_i$（只能有一种情况成立），且对于所有指标 $i \in 1..k$, $X_i = \{y_1, \ldots, y_n\}$。我们将 $1..k$ 中满足 c（x_i）有定义的指标子集定义为 I。根据条件（CBot），对于所有 $v \in X_i$, c（x_i）$\neq \perp$ 当且仅当 c（v）$\neq \perp$。因此，若 c（x_i）$\neq \perp$，则根据条件（CEq），ϕ_i，$\lfloor c \rfloor$ 是相容的。此外，ϕ_i 中的所有变量的值均由 $\lfloor c \rfloor$ 确定（这些变量在每个解中的取值都相同）。因此，将 $\lfloor c \rfloor$ 与（ϕ_i）$_{i\in I}$ 结合得到的方程组有唯一解。反之，若 c（x_i）$= \perp$，则 ϕ_i 中没有变量由$\lfloor c \rfloor$定义。尽管如此，我们知道方程组 E 是相容的。事实上，根据 E-等价的性质，我们知道 E, $\lfloor m_1 \rfloor$ 有解，因此 E也有解。因此，（ϕ_i）$_{i\notin I}$ 中的方程所组成的方程组是相容的。由于该方程组与（ϕ_i）$_{i\in I}$中的方程没有共同变量，因此 E，$\lfloor c \rfloor$ 是相容的。

对于第二种情况，我们假设 c 是全的，且 E, $\lfloor c \rfloor$ 相容。由于 c 是全的，条件 (CBot) 成立（$c(v) \neq \perp$ 对 $[\![E]\!]$中的所有节点均成立）假设我们有 $(N_1, m_1) \rhd_E (N_2, m_2)$。对于条件 (CEq)，我们依赖于 $[\![E]\!]$ 良定义这一事实 (T4)。事实上，对于 E 中的所有方程，都有一个对应的关系 $X \rightarrow\!\!\bullet\ v$ 或 $v \circ\!\!\rightarrow X$。因此，E，$\lfloor c \rfloor$ 相容意味着 $c(v) = \sum_{w\in X} c(w)$。

我们可以证明若干性质，说明 TFG 的结构如何约束良定义配置中的可能取值。这些结果可以视为一种等价的“令牌游戏”，它解释了令牌如何沿着 TFG 的弧传播。在我们的情境中，这很有用，因为当两个节点可以在同一配置中被标记时，我们便可以判定它们是并发的。（对于寻找非并发节点对，也有类似的结果。）

我们的第一个结果表明，我们总能将令牌从一个节点传播到其子节点；也就是说，如果一个节点有令牌，就能在其后继节点中找到令牌（可能处于不同的良定义配置中）。性质（后向）陈述了一个对偶结果：如果一个子节点被标记，那么其某个父节点必定也被标记。

引理 6.2 (令牌传播)。 假设 $[\![E]\!]$ 是等价关系$(N_1, m_1) \rhd_E (N_2, m_2)$ 下的良定义 *TFG*，且 c 是 $[\![E]\!]$的一个良定义配置。

（聚合）若 $p \circ\!\!\rightarrow \{q_1, \ldots, q_k\}$ 且 $c(p) \neq \perp$，则对于满足 $c(p) = \sum_{i\in 1..k} l_i$ 的每个 $\mathbb{N}^k$ 序列 $(l_i)_{i\in 1..k}$，我们都能找到一个良定义配置 c'，使得 $c'(p) = c(p)$，并且对于所有 $i \in 1..k$, $c'(q_i) = l_i$，且对于每个不在 $\downarrow p$ 中的节点 v，$c'(v) = c(v)$。

（前向）若 p、q 是满足 $c(p) \neq \perp$ 且 $p \rightarrow^\star q$ 的节点，则我们可以找到一个良定义配置 c'，使得$c'(q) \geqslant c'(p) = c(p)$ 且 $c'(v) = c(v)$，对于每个不在 $\downarrow p$ 中的节点 v 均如此。

（后向）若 c（p）> 0，则存在一个根节点 v，使得 $v \rightarrow^\star p$ 且 c（v）> 0。

证明：我们分别证明每个性质。不失一般性，我们假设节点上存在一个（任意的）全序。

聚合：令 p $p \circ\!\!\rightarrow X$ $X = \{q_1, \ldots, q_k\}$ $l_1 \ldots l_k \in \mathbb{N}^k$ 为一个节点，使得，其中，并令 $(,,)$ 为一个序列，使得 $c(p) = \sum_{i\in 1..k} l_i$。我们将配置 c' 定义为一个递归函数。基本情形定义如下：$c'(p) = c(p)$，对所有 $i \in 1..k$, $c'(q_i) = l_i$，以及 $c'(v) = c(v)$，对所有满足 $v \notin \downarrow p$ 的节点 v。递归情形仅涉及 X 中节点的后继节点。令 w 为这样一个节点。它不可能是根节点（因为 w 是 X 中某个节点的后继节点），因此它至少有一个父节点 x。我们考虑两种情况：

- 或者 $x \rightarrow\!\!\bullet\ w$ 成立：那么，令 Y为 w 的父节点集合（正如预期，$x \in Y$）。定义 5.2 中的性质（T3）意味着 $Y \rightarrow\!\!\bullet\ w$ 成立，因此我们定义 $c'(w) = \sum_{y\in Y} c'(y)$。
- 或者 $x \circ\!\!\rightarrow w$成立：那么，根据性质（T3），x是 w唯一的父节点。令 Y为x的聚合子节点集合：我们有 $x \circ\!\!\rightarrow Y$，且 $w \in Y$。如果 w 是 Y中最小的节点（依据节点上的全序），则定义 $c'(w) = c'(x)$；否则定义 $c'(w) = 0$。由此可得$c'(x) = \sum_{y\in Y} c'(y)$（其中除一个定义为 $c'(x)$ 的项外，所有项均定义为零）。

注意，递归总是蕴含给定节点的父节点，因此是良基的，因为 TFG 是有向无环图。显然可以验证 c' 是良定义的：根据构造，在 c' 有定义的所有节点上，(CEq) 均成立。

前向：取 $[\![E]\!]$ 的一个良定义配置 c，并假设有两个节点 p, q，满足 $c(p) \neq \perp$ 和 $p \rightarrow^\star q$。证明通过对从 p 到q的路径长度进行归纳来完成。初始情形为 $p = q$，这显然成立。否则，假设 $p \rightarrow r \rightarrow^\star q$。只需找到一个良定义配置 c'，使得 $c'(r) \geqslant c'(p) = c(p)$。由于不在 $\downarrow p$ 中的节点不在从 p 到 q 的路径上，因此对于任意不在 $\downarrow p$中的节点 v，我们都可以确保$c'(v) = c(v)$。证明通过对 $p \rightarrow r$进行分类讨论来完成：

- 要么 $X \rightarrow\!\!\bullet\ r$ 与 $p \in X$。那么根据 (CEq)，我们有$c(r) = c(p) + \sum_{v\in X,\ v\neq p} c(v) \geqslant c(p)$，并且可以选择 $c' = c$。
- 或者，我们有 $p \circ\!\!\rightarrow X$ 与 $r \in X$。根据 (Agglomeration)，我们可以找到一个良定义配置 c'，使得 $c'(c' = c$ $c'\ v = 0$ $)$ (p) (p)（并且对于所有 $v \in X \setminus \{r\}$，也有 $()$）。

后向：令 c 为 $[\![E]\!]$ 的一个良定义配置，并假设我们有 $c(p) > 0$。证明通过对有向无环图进行逆向结构归纳得出。如果 p 是一个根节点，则结论显然成立；否则，p 至少有一个父节点 q，使得 $q \rightarrow p$。如上所述，我们进行分类讨论。

- 或者 $X \rightarrow\!\!\bullet\ p$ 满足 $q \in X$。根据 (CEq)，我们有$c\ p = \sum_{v\in X} c\ v > 0$ $()$ $()$。因此，在 X 中必定至少存在一个节点q'，使得 $c(q') > 0$，由此根据关于 q' 的归纳假设即可得出结论；q' 是 p 的父节点。

· 或者我们有 $q \circ\!\!\rightarrow X$ 满足 $p \in X$。根据 (CEq)，我们有 $c(q) = \sum_{v \in X} c(v) \geqslant c(p) > 0$，由此根据关于 q 的归纳假设即可得出结论；q 同样是 p 的父节点。

到目前为止，我们的所有结果都不依赖于 E-等价的性质。现在我们证明 N_1 与 N_2 的可达标识和 $[\![E]\!]$ 的配置之间存在等价关系。更准确地说，我们证明（定理 6.3）： N_1 或 N_2 中的每个可达标识，都可以扩展为 $[\![E]\!]$ 的一个全定义配置。这意味着，我们可以通过考察由 N_2 的可达标识得到的良定义配置，重建 N_1 的所有可达标识。我们用于计算并发关系的算法（见第 8 节）将更为高效，因为我们不必穷举 N_2 的所有标识。相反，我们只需知道哪些根节点可以同时被标记。

定理 6.3 (配置可达性)。 假设 $[\![E]\!]$ 是关于等价关系 $(N_1, m_1) \rhd_E (N_2, m_2)$ 的良构 *TFG*。若 m 是 $R(N_1, m_1)$ 或 $R(N_2, m_2)$ 中的一个标识，则存在 $[\![E]\!]$ 的一个全定义配置 c，使得 $c \equiv m$。反过来，给定 $[\![E]\!]$的一个全定义配置 c，当且仅当 $c_{\lfloor N_2}$ 在 (N_2, m_2)中可达时，标识 $c_{\lfloor N_1}$ 在 (N_1, m_1) 中可达。

证明： 我们分别证明各点。

首先，取一个标识 m，使其属于 $R(N_1, m_1)$（ m 属于 $R(N_2, m_2)$ 的情形类似）。根据定义 4.1 的性质 (A1), E $\lfloor m \rfloor$ 相容。因此，它存在一个非负整数解 c，即对 $fv(E)$ N_1 和 N_2 中所有变量和库所的赋值，使得 $E, \lfloor c \rfloor$ 相容，并且当 $p \in N_1$ 时， $c\ p = m\ c$ ( ) (p)。我们可以自由地扩展赋值，使其包含 K 中的常量（其取值固定）；因此，根据定义 5.2 的条件 (T1)，该解 c 定义在 $[\![E]\!]$的所有节点上。由引理 6.1 可知，它是良定义的。

对于逆向性质，我们假设 c 是 $[\![E]\!]$ 的一个全定义且良定义的配置，并且 $c_{\lfloor N_1}$ 属于 $R(N_1, m_1)$（$c_{\lfloor N_2}$ 属于 $R(N_2, m_2)$ 的情形也类似）。由于 c 是一个良定义的配置，根据引理 6.1 可知， $E, \lfloor c \rfloor$ 相容。因此， $E, \lfloor c_{\lfloor N_1} \rfloor, \lfloor c_{\lfloor N_2} \rfloor$ 相容。根据定义 4.1 的条件 (A3)，可得 $c_{\lfloor N_2}$ 属于 $R(N_2, m_2)$，证毕。

该定理的第二个结果为以下定义提供了依据。

定义 6.2 (可达配置)。 对于等价陈述 $(N_1, m_1) \rhd_E (N_2, m_2)$，配置 c 是可达的，当且仅当 c $c_{\lfloor N_1} \in R\ N_1\ m_1$ $c_{\lfloor N_2} \in$ 是全定义的、良定义的， $R\ N_2\ m_2$ 并且满足(,) (分别满足 (,)) 。

前述基本结果表明，可以利用 TFG 根据约简网的状态空间推理初始网的状态空间，反之亦然。

7 标识可达性

我们通过描述一种简单的模型检测算法，展示令牌流图的优势。该算法旨在通过检查较小的网 (N_2, m_2) 上的可达性性质，判断初始网 (N_1, m_1) 中某个标识（例如 m_1'）是否可达。我们先证明一些辅助结果。

引理 7.1 (标识约简的唯一性)。 假设 $[\![E]\!]$ 是等价关系 $(N_1, m_1) \rhd_E (N_2, m_2)$的良构 *TFG*。给定 N_1 的一个标识 m_1'，至多存在一个全定义配置 c，使得 $c \equiv m_1'$。

证明： 设 c_1 和 c_2 是两个全定义配置，满足 $c_1 \equiv m_1'$ 和 $c_2 \equiv m_1'$。设 X 为节点x 的集合，使得 $c_1(x) \neq c_2(x)$。反证，假设 X 非空（也就是说，假设 $c_1 \neq c_2$）。对于 X中的每个节点 x，我们知道 x 不属于 P_1，因为 c_1 和 c_2 在 m_1'上相同。因此，根据定义 5.2 的性质（T6）， X 中的每个 x 都有一个输出 $x \circ\!\!\rightarrow y$。更一般地，对于 X中的每个 x，都有 $x \circ\!\!\rightarrow Y$，其中 Y是某个非空集合。我们考虑 X 中的一个元素 x_0，使得 $x_0 \circ\!\!\rightarrow Y_0$成立，且 Y_0 与 X 不相交（如果 X 非空，则必然存在这样的元素 x_0；否则 X 将包含一个由 $\circ$弧构成的环，这与良构 TFG 的无环性质（T5）相矛盾）。接着，由于 c_1 是良定义的，根据性质（CEq），我们知道 $c_1(x_0) = \sum_{y \in Y_0} c_1(y)$。然而，对于所有 $y \in Y_0$，都有 $c_1(y) = c_2(y)$，因为 Y_0 与 X不相交。因此， $c_1(x_0) = \sum_{y \in Y_0} c_2(y) = c_2(x_0)$，这与 $x_0 \in X$矛盾。由此可得， X 必为空，也就是说 $c_1 = c_2$。良定义配置至多有一个。

因此，由该引理和定理 6.3可得以下推论：

定理 7.2 (可达性判定)。 假设 $[\![E]\!]$ 是等价关系 $(N_1, m_1) \rhd_E (N_2, m_2)$的良构 *TFG*。判定标识 m_1' 在 $R(N_1, m_1)$ 中是否可达，相当于构造一个全定义配置 c，使得 $c \equiv m_1'$，然后检查 $c_{\lfloor N_2}$ 在 $R(N_2, m_2)$中是否可达。

因此，给定一个等价关系 $(N_1, m_1) \rhd_E (N_2, m_2)$ 及其关联的 TFG $[\![E]\!]$，我们首先按照下文算法 1 的做法，将我们关注的标识m_1' 扩展为一个全定义配置 c。（引理 7.1 保证，若这样的配置存在，则它是唯一的。）如定理 7.2 所述，如果 c 限制到 N_2 后是 (N_2, m_2) 中可达的标识，那么 m_1' 在 (N_1, m_1) 中可达。否则， $\neg \lfloor m_1' \rfloor$ 是 (N_1, m_1) 上的一个不变量R。

我们通过带标识网 M_1 上的两个具体示例来说明该算法，该网见图 1。假设我们要判断标识 $m_1' \triangleq (p_0 = 0, p_1 = 2, p_2 = 0, p_3 = 1, p_4 = 1, p_5 = 1, p_6 = 0)$ 是否在 (N_1, m_1) 中可达，其中 m_1 如图 1 所示。该标识可以扩展为全定义配置 c，其中 $c(a_1) = c(a_2) = 2$。因此，判断标识 m_1' 在 (N_1, m_1) 中是否可达，等价于判断标识$m_2' \triangleq (p_0 = 0, a_2 = 2, p_6 = 0)$ 是否在 (N_2, m_2') 中可达（实际上不可达）。请注意，如果初始标识 m_1为 $(p_0 = 2, p_6 = 1)$ 且其他库所为空，则 m_1'可达。反过来，假设我们关注的标识为 m_1''，使得$m_1''(p_4) = 2$且 $m_1''(p_1) = m_1''(p_2) = 0$。由于$c(a_1) = m_1''(p_1) + m_1''(p_2) = 0$且 $c(a_1) = c(a_2) > m_1''(p_4)$，因此无法将该标识扩展为良定义配置 c。在这种情况下，我们可以直接得出结论：对于任意初始标识 m_1， m_1'' 在 (N_1, m_1) 中不可达。

可达函数（见算法1）是定理7.2的直接实现：它构造一个全配置c，然后检查其是否良定义（我们省略函数良定义，这显而易见），最后判断 $c_{\lfloor N_2}$ 在 (N_2, m_2)中是否可达。该算法依赖递归过程自底向上（见算法2），若存在，则将感兴趣的标识 m_1' 扩展为全定义配置。

我们注意到，第二个算法是递归的，它总会终止，因为它只是沿着 TFG 结构执行。我们仍需证明算法1 总能返回正确答案。

证明： 我们分两种情形讨论：

情形*C1*：算法返回 false，因为 c 不是良定义的（第7行）。在这种情况下，我们接下来证明，不存在扩展 c 的良定义配置 m_1'，因此根据定理 m_1' 不可达。6.3。

情形 *C* 2 $c_{\lfloor N_2} \in$: $R\ N_2\ m_2$ 7 2, 算法返回 (,) 的值。在此情形下，得益于定理 。，只需证明 c 是全定义的、良定义的，并且扩展了 m_1'。

我们从情形 C2 开始，它基本上说明了算法的可靠性。下面证明 c 是全定义的：对于每个节点 v，如果

算法 1 可达$(m_1', [\![E]\!], (N_2, m_2))$

1: **输入：** m_1' ： N_1 的一个标识
(N_2, m_2)：一个约简网，满足
$(N_1, m_1) \rhd_E (N_2, m_2)$ 成立
$[\![E]\!]$ ：{v1} 的良构 TFG
上述 E-等价
输出： 一个布尔值，指示 $m_1' \in R(N_1, m_1)$
2: $c \leftarrow \vec{\perp}$;; c 是 $[\![E]\!]$ 的一个配置
3: **对所有** $p \in P_1$，**执行** $c[p] \leftarrow m_1[p]$
4: **对所有** $v^n \in K$，**执行** $c[v] \leftarrow n$
5: ; ; $[\![E]\!]$ 是 (V, R, A)，如定义 5.1
6: **对所有** $v \in V$，**执行** 自底向上$(c, v, [\![E]\!])$
7: **返回** 良定义$(c) \wedge c_{\lfloor N_2} \in R(N_2, m_2)$

算法 2 自底向上$(c, v, [\![E]\!])$

1: **In:** $[\![E]\!]$ ： TFG 结构
v， $[\![E]\!]$中的一个节点
输入： c， $[\![E]\!]$的部分配置
后置条件： c 对 $\downarrow v$的所有节点均有定义
2: **对于所有满足** $v \to v'$ **的** v'，**执行**
3: 自底向上$(c, v', [\![E]\!])$
4: **结束循环**
5: **如果** $v \circ\!\!\to X$ **那么** $c[v] \leftarrow \sum_{v' \in X} c[v']$

v 是一个常数，或者属于 P_1，则由算法 1 的第 3 和第 4 行赋值。否则， v 不是 $\circ$-叶子（根据定义 5.2的性质（T6）），因此由算法 2 的第 5 行赋值；算法 2 会在 TFG 的每个节点上调用（算法 1的第 c 行）。此外， c 是良定义的，因为它通过了第 7 行的测试。根据第 m_1' 行，它还扩展了 3（这些值之后不会在算法 2中被覆盖，这是性质（T6）所保证的）。

情形C1说明了该算法的完备性。通过逆否命题，我们假设存在一个良定义的全配置 c' ，它扩展了 m_1'，并通过对 BottomUp 的递归调用进行归纳，证明算法构造出的配置 c 等于 c'。更准确地说，我们证明调用 $BottomUp(c, v, [\![E]\!])$会返回一个配置 c，它在 $\downarrow v$的所有节点上都与 c' 一致。算法构造的初始配置c 扩展了 m_1' ，并设置了常量（算法 1 的第 3 行和第 4 行）。接下来，对于任意一次调用$BottomUp(c, v, [\![E]\!])$，对 $\downarrow v$ 中的每个节点 w，若 P_1中的某个节点是 w，则 $c(v) = c'(v)$立即成立。否则， w不是 $\circ$-叶节点。若 $w \neq v$，则对于 v的某个子节点 u， w属于 $\downarrow u$，并且根据对第 3 行递归调用 $BottomUp(c, u, [\![E]\!])$的归纳假设， $c(v) = c'(v)$成立。若$w = v$，则第 5 行设置 $c(v)$，也就是说， $c(v) = \sum_{v' \in X} c(v')$。根据归纳假设，对于 v 的所有 v' 个子节点，均有 $c(v') = c'(v')$（该递归调用同样发生在第 3 行）。因此，由于 c' 是良定义的，根据性质（CEq），$c(v) = \sum_{v' \in X} c'(v') = c'(v)$。由此，对于 $\downarrow v$ 中的所有 w ，均有 $c(w) = c'(w)$。因此， $c = c'$，因为 BottomUp 会在 TFG 的所有节点上调用。

状态空间划分 在关联的 TFG 良定义的情况下，我们可以利用前面的结果，推导出一个关于等价佩特里网状态空间的有趣结论。事实上，我们可以证明，在这种情况下，可以构造一个 (N_1, m_1) 的可达标识划分，该划分与 (N_2, m_2) 的可达标识一一对应。

给定约简网 N_2 的一个标识 m_2' ，我们将 $Inv(m_2')$ 定义为初始网 N_1 中与 m_2' 相关联的标识集合。

$$Inv(m_2') \triangleq \{c_{\lfloor N_1} \mid c \text{ 全定义配置 } \wedge c \equiv m_2'\}$$

定理 7.3 (状态空间划分)。 假设 $[\![E]\!]$ 是等价$(N_1, m_1) \rhd_E (N_2, m_2)$的一个良构 *TFG*。集合族 $P \triangleq \{Inv(m_2') \mid m_2' \in R(N_2, m_2)\}$ 是 $R(N_1, m_1)$ 的一个划分。

证明： 以下几点表明， P 构成一个划分：

P 中没有空集：对于任意 m_2'属于 $R(N_2, m_2)$，根据定理 6.3，存在一个全定义且良定义的配置 c使得 $c \equiv m$。因此， $Inv(m_2')$ 非空。这意味着 $\emptyset \notin P$。

并集 $\cup_{A \in P} A$ **覆盖** $R(N_1, m_1)$：取 m_1'属于$R(N_1, m_1)$。根据定理6.3，存在一个全定义且良定义的配置 c $c \equiv m$ $c_{\lfloor N_2} \in R\ N_2\ m_2$ P $R\ N_1\ m_1$ 使得且(,)。因此，这些集合覆盖(,)。

两两不相交：取两个不同的标识 m_2' 和 m_2''，它们属于$R(N_2, m_2)$。根据引理7.1，我们有 $Inv(m_2') \cap Inv(m_2'') = \emptyset$，因为$(N_1, m_1)$ 中的每个标识至多能扩展为一个可能的配置 c。

作为推论，当一个带标识网 (N_1, m_1) 可以部分归约时，我们知道如何将其状态空间划分为互不相交的凸集之并；也就是说，这些集合是由线性方程组的解所定义的标识集合。

8 并发关系

本节介绍一种利用 TFG 计算佩特里网并发关系的加速算法。其思路与前面用于可达性判定的算法截然不同。给定等价陈述 $(N_1, m_1) \rhd_E (N_2, m_2)$，这里的思路是直接在约简网 (N_2, m_2) 上计算并发关系，并利用相应的 TFG $[\![E]\!]$将信息追溯回初始网 (N_1, m_1)。

接下来，我们将重点讨论安全网。幸运的是，我们的化简规则保持安全性（参见推论 8.1.1）。因此，如果 (N_2, m_2) 是安全的，我们就无需检查 (N_1, m_1) 是否安全。网的安全性会对配置产生影响。

引理 8.1 （安全配置）。 设 $[\![E]\!]$ 是一个良定义的 *TFG*，对于一个安全佩特里网 $(N_1, m_1) \rhd_E (N_2, m_2)$。则对于 $[\![E]\!]$ 的每个全定义配置 c，若其在 (N_1, m_1) 中可达，并且对于每个不属于 K的节点 v，都有 $c(v) \in \{0, 1\}$。

证明：我们用反证法证明该结论。取一个全定义配置 c，使得 $c_{\lfloor N_1}$ 在 (N_1, m_1) 中可达，并取一个节点 v，使得 $c(v) > 1$。由于 (N_1, m_1) 是安全的， v不属于 P_1。由定义 5.2 的性质（T6），可知 v不是 $\circ$-叶节点，因此 v有一些输出弧，使得 $v \circ\!\!\to X$。此外，性质（T6）还意味着存在 N_1 中的一个库所 p，使得 $v \to^\star p$。根据引理 6.2（正向），我们可以找到 $[\![E]\!]$ 的一个良定义配置 c'，使得对于每个不属于 $\downarrow v$的节点 w，都有 $c'(p) \geqslant c'(v) = c(v) > 1$，且 $c'(w) = c(w)$。后者蕴含 $c'_{\lfloor N_2} = c_{\lfloor N_2}$。于是，根据定理 6.3， $c'_{\lfloor N_1}$ 在 (N_1, m_1) 中可达。然而， $c'(p) > 1$，这与 (N_1, m_1) 的安全性矛盾。

推论 8.1.1 (安全性保持)。 假设 $[\![E]\!]$ 是针对$(N_1, m_1) \rhd_E (N_2, m_2)$ 的良构 *TFG*。如果(N_1, m_1) 是安全的，那么(N_2, m_2) 也是安全的。

我们的方法基于这样一个事实：我们可以将（带标识网中的）并发库所这一概念扩展为 TFG 中的并发节点，即在可达配置中可以同时被标记的节点（如定义6.2所述）。

根据定理 6.3，如果我们取 N_2 中的可达标识——也就是说，我们固定 $[\![E]\!]$ 中根节点的值——就可以通过将令牌从根节点传播到叶节点，找到 N_1 中同时被标记的库所（引理 6.2）。接下来，在我们的算法中，我们通过考虑两种情况来计算 N_1 的并发关系：(1) 我们从单个根节点 p 中的一个令牌开始，其中 p 非死，并将该令牌向前传播，直到找到 N_1 中有两个库所同时被标记的配置（这基本上是由某些冗余库所导致的）；或者 (2) 我们以相同方式进行，但将令牌分别放置在两个独立的根节点 p_1 和 p_2 中，使得 $p_1 \parallel p_2$。

8.1 算法描述

我们假设 $[\![E]\!]$ 是关于关系 $(N_1, m_1) \triangleright_E (N_2, m_2)$ 的良构 TFG。我们用符号 $\|_2$ 表示 (N_2, m_2) 上的并发关系，用 $\|_1$ 表示 (N_1, m_1) 上的并发关系。$[\![E]\!]$ 的节点集合为 P。

我们定义一种算法，其输入为一个良构 TFG $[\![E]\!]$，以及网 (N_2, m_2) 上的并发关系 $\|_2$，输出为 (N_1, m_1) 上的并发关系 $\|_1$。实际上，我们的算法会计算一个并发矩阵 C。C 是一个对称矩阵：当节点 v、w 可以在某个可达配置中同时被标记时，$\mathrm{C}[v, w] = 1$；否则为 0。我们证明（定理 8.7），由 C 导出的关系与 N_1 上的 $\|_1$ 相符。我们的算法可以根据实际需要在达到给定时间限制后中断，此时会返回一个部分关系 $\|_2$。矩阵 C 中未定义的情形记为 $\mathrm{C}[v, w] = \bullet$，此时该矩阵称为不完整矩阵。

计算并发关系的复杂度高度依赖网中库所的数量。因此，我们称算法执行了某种“降维”，因为它使我们能够通过先在较低维空间中求解问题（因为 N_2可能少得多的库所），再将结果映射回原始网，从而求解高维空间（N_1中的库所数量）中的问题。实践中，我们使用 CADP 工具箱中的 cæsar.bdd 工具计算（N_2，m_2）上的并发关系；但也可以依靠任意类型的“预言机”来为我们计算该关系。若初始网完全可约，则无需执行此步骤；此时，N_2 的并发关系是平凡的，且 $[\![E]\!]$ 中的所有根节点都是常量。

为简化记号，我们假设当 v 是 K中的常量节点且 w 非死时，$v \parallel_2 w$。相反，当 $v \in K$为(0)或 w 为死时，$v \#_2 w$。

我们的算法分为两个主函数 3 和 4。此外，它还隐式依赖一个辅助函数，该函数会返回给定节点 x 的后继节点 $\downarrow x$（细节从略）。在主函数 Matrix 中，我们遍历 $[\![E]\!]$ 的非死根节点，并递归传播节点 v 为非死的信息：第 5 行对 Propagate 的调用通过找出由唯一根节点 v 导致的所有并发节点，更新并发矩阵 C。我们可以证明，所有此类情况都源于起点位于 $\downarrow v$ 的冗余弧。更确切地说，我们在引理 8.5 中证明，若 $v \rightarrow\!\!\bullet\, w$ 成立，则集合 $\downarrow v \setminus \downarrow w$ 中的节点与 $\downarrow w$ 中的所有节点并发。算法 4 第 11 行的 for 循环明确实现了这一点。接下来，在 Matrix 的第二个 for 循环中，我们计算由两个不同的非死根节点（v、w）导致的并发节点。在这种情况下，我们可以证明，v 的所有后继节点都与 w 的后继节点并发：$\downarrow v \times \downarrow w$ 中的所有节点对都是并发的。

我们可以对算法的复杂度进行粗略分析。从根节点出发，沿着 $[\![E]\!]$的边递归调用 Propagate，以更新矩阵。（当然，一个直接的优化方法是标记已访问的节点，这样就不会对同一节点重复调用函数 Propagate。由于这项优化不会影响可靠性、完备性或理论复杂度，因此我们不再详述。）更准确地说，我们只对 $[\![E]\!]$中的非死节点调用 Propagate。

算法 3 Matrix($[\![E]\!]$, $\|_2$)

```
1:  In: [[E]]：TFG 结构
        ||_2：(N_2, m_2) 上的并发关系
    输出：并发矩阵 C。
2:  C ← 0⃗          ;; C 是一个以 P × P 为索引的矩阵
3:  ;; v (∈ P_2) 非死当且仅当 v ||_2 v 成立
4:  对所有 v 非死根节点 [[E]] 中的节点执行
5:      传播([[E]], C, v)
6:  结束循环
7:  ;; v 与 w (∈ P_2) 并发当且仅当 v ||_2 w 成立
8:  对所有 (v, w) 互异的并发根节点 [[E]] 中执行
9:      对所有(v', w') ∈ ↓v × ↓w 执行
10:         C[v', w'] ← 1
11:         C[w', v'] ← 1
12:     结束循环
13: 结束循环
14: 返回C
```

算法 4 传播($[\![E]\!]$, C, v)

```
1:  In:        [[E]]：TFG 结构
               v：节点
    输入：     并发矩阵 C。
    后置条件： C 包含所有并发关系
               由已知以下关系
               v 是非死的。
2:  ;; 此循环包含 C[v, v] ← 1
3:  对所有 w ∈ ↓v 执行
4:      C[v, w] ← 1
5:      C[w, v] ← 1
6:  结束循环
7:  对于所有满足 w 的 v → w 执行
8:      传播([[E]], C, w)
9:  结束循环
10: 对于所有满足 w 的 v →• w 执行
11:     对于 (v', w') ∈ ((↓v \ ↓w) × ↓w) 执行
12:         C[v', w'] ← 1
13:         C[w', v'] ← 1
14:     结束循环
15: 结束循环
```

因此，我们的算法执行的函数调用次数与非死节点的数量呈线性关系。每次调用Propagate时，最多可以更新 $O(N^2)$ 个 C 中的值，其中 N 是 $[\![E]\!]$ 中的节点数（参见第 行 的 11 循环）。因此，给定并发关系 $O(N^3)$，我们的算法复杂度为 $\|_2$。这需要与构建并检查该网的状态空间的复杂度进行比较，后者属于 PSPACE。因此，计算维度较低的 $\|_2$ (N_2, m_2) 的并发关系，并将其追溯到 (N_1, m_1) 非常有益。

实际上，我们的算法效率很高；与计算并发关系所涉及的其他任务相比，其执行时间通常可以忽略不计。第9节中给出了一些性能结果。

8.2 正确性证明

该算法的可靠性和完备性证明依赖于以下定义：

定义8.1（并发节点）。$[\![E]\!]$上的并发关系（记为 $\mathcal{C}$）是 $[\![E]\!]$ 中节点对之间的关系，满足：当且仅当存在一个全定义配置 c，使得 $v\,\mathcal{C}\,w$ 成立，其中：*(1)* c 可达，即 $c_{|N_2} \in R(N_2, m_2)$；且*(2)* $c(v) > 0$ 且 $c(w) > 0$。

$[\![E]\!]$ 上的并发关系 $\mathcal{C}$ 是 N_1 上的并发关系 $\|_1$ 和 N_2上的并发关系 $\|_2$ 的推广：对于任意一对库所 $(p, q) \in P_1^2$，根据定理 6.3，有 $p \,\|_1\, q$ 当且仅当 $p\,\mathcal{C}\,q$。对于 $(p, q) \in P_2^2$，情况类似：$p \,\|_2\, q$ 当且仅当 $p\,\mathcal{C}\,q$。在后一种情况下，我们称 p、q 为并发根。因此，$\mathcal{C}$ 是对称的，而 $v\,\mathcal{C}\,v$ 表示 v 是非死的（即存在一个赋值 c，使得 $c(v) > 0$）。我们可以将这一概念扩展到常量：当 $v_1\,\mathcal{C}\,v_2$ 成立时，我们称两个根v_1、v_2 并发；当 $v_1\,\mathcal{C}\,v_1$成立时，我们称根 v_1 非死。这包括 v_1或 v_2 位于 K中的情况（它们是值为1的常量）。

我们证明关于关系 $\mathcal{C}$ 的一些性质，这些性质是我们令牌传播性质的直接推论。对于以下所有结果，我们默认假设，对于关系 $(N_1, m_1) \rhd_E (N_2, m_2)$，$[\![E]\!]$ 是一个良构的 TFG，两个带标识网都是安全的，并且 $\mathcal{C}$ 是 $[\![E]\!]$的并发关系。

8.2.1 检查非死节点

我们首先介绍一个性质（引理 8.2），指出非死节点的后继也是非死节点。引理8.3 给出了一个对偶结果，可用于证明我们方法的完备性；该结果指出，只需探索非死根节点，就能找到所有非死节点。

引理 8.2。如果 $v\,\mathcal{C}\,v$ 且 $v \to^\star w$，那么 $w\,\mathcal{C}\,w$ 且 $v\,\mathcal{C}\,w$。

证明：假设 $v\,\mathcal{C}\,v$。这意味着存在一个全定义、良定义的配置 c，使得 $c(v) > 0$ 且 $c_{|N_2} \in R(N_2, m_2)$。现在取 v 的一个后继节点，比如 v'。根据引理 6.2，我们可以找到另一个可达配置 c'，使得对于所有不在 $\downarrow v$ 中的节点 x，$c'(x) \geqslant c(x) = c'(v)$，且 $c'(x) = c(x)$。因此，我们同时有 $w\,\mathcal{C}\,w$ 和 $v\,\mathcal{C}\,w$。

引理 8.3. 如果 $v\,\mathcal{C}\,v$，那么存在一个根节点 v_0，使得 $v_0\,\mathcal{C}\,v_0$ 和 $v_0 \to^\star v$。

证明： 假设 $v\,\mathcal{C}\,v$。那么存在一个全定义配置 c使得 $c_{|N_2} \in R(N_2, m_2)$ 且 $c(v) > 0$。根据引理 6.2 的反向传播性质，我们知道存在一个根节点，记作 v_0，使得 $c(v_0) \geqslant c(v)$ 且 $v_0 \to^\star v$。因此 v_0 在 $[\![E]\!]$中是非死的。

8.2.2 检查并发节点

对于并发节点而非非死节点，我们也可以证明类似的结果。我们考虑函数 Matrix所考虑的两种情况：并发节点由两个并发根得到（引理 8.4）；或由于冗余弧而由单个非死根节点得到（引理 8.5）。最后，引理 8.6 给出了相应的完备性结果。

引理 8.4. 假设 v、w 是 $[\![E]\!]$ 中的两个节点，且 $v \notin \downarrow w$ 和 $w \notin \downarrow v$。如果 $v\,\mathcal{C}\,w$，那么对于所有节点对 $(v', w') \in \downarrow v \times \downarrow w$，都有 $v'\,\mathcal{C}\,w'$。

证明： 假设 $v\,\mathcal{C}\,w$, $v \notin \downarrow w$ 和 $w \notin \downarrow v$。根据定义，存在一个全定义配置 c，使得 $c(v)$ $c(w) > 0$ 且 $c_{|N_2} \in R(N_2, m_2)$。

取 v' 在 $\downarrow v$中的一个后继，应用引理 6.2 中的令牌传播，可以构造一个全定义配置 c' 关于 $[\![E]\!]$，使得 $c'(v') \geqslant c'(v) = c(v)$ 且 $c'\ x = c\ x \quad x \quad \downarrow v \quad c'\ w = c\ w > ()() 0$ 对于任何不在 中的节点均成立。因此，$()()$。

我们可以再次对 c' 使用令牌传播性质。这将得到一个全定义配置 c''，使得$c''(w') \geqslant c''(w) = c'(w) = c(w)$ 且 $c''(x) = c'(x)$ 对于任何不在 $\downarrow w$中的节点 x 均成立。

我们仍需证明 $v' \notin \downarrow w$（在树结构中这显而易见，但在我们的有向无环图结构中则需要额外证明）。然后，只需注意到它蕴含 $c''(v') = c'(v') \geqslant c(v)$，因此可得所需的 $v'\,\mathcal{C}\,w'$。

我们用反证法证明 $v' \notin \downarrow w$ $v' \in$。事实 $\downarrow w$ $\downarrow v \cap \downarrow w \neq \emptyset$ E 上，假设。因此，。此外，由于 是良构 TFG，必定存在（条件 (T3)）三个节点 p、q、r，使得 $X \to\!\bullet\, r$，$p \in \downarrow v \cap X$ 且 $q \in \downarrow w \cap X$。与引理 6.2的证明类似，我们可以将 v 中包含的令牌 w 传播到p，q，并由 (CEq) 得到 $c''(r) > 1$，这与我们假设网是安全的相矛盾。

引理 8.5。若 $v\,\mathcal{C}\,v$ 且 $v \to\!\bullet\, w$，则对于满足 $v' \in (\downarrow(v) \setminus \downarrow w)$ 及 $w' \in \downarrow w$ 的每一对节点 (v', w')，都有 $v'\,\mathcal{C}\,w'$。

证明： 假设 $v\,\mathcal{C}\,v$ 且 $v \to\!\bullet\, w$。根据定义，存在一个全定义配置 c，使得 $c(v) > 0$ 且 $c_{|N_2} \in R(N_2, m_2)$。此外，根据 $v \to\!\bullet\, w$ 和条件（CEq），我们有 $c(w) > 0$。

取 w' 于 $\downarrow w$ 中。由引理 6.2，我们可以找到一个全定义配置 c'，使得对于任意不在 $\downarrow w$ 中的节点 x，都有 $c'(w') \geqslant c'(w) = c(w) > 0$ 且 $c'(x) = c(x)$。由于 v 不在$\downarrow w$ 中，我们有 $c'(v) = c(v)$。同样，N_2 中的库所是根节点，因此不能属于 $\downarrow w$。所以我们有 $c'_{|N_2} \equiv c_{|N_2}$，这意味着 $c'_{|N_2}$ 在 (N_2, m_2) 中可达。此时我们有 $v\,\mathcal{C}\,w'$。

现在，考虑 $v' \in \downarrow v \setminus \downarrow w$，其中 $v' \neq v$（若 $v' = v$，则预期结果已经成立）。必然存在 $v_0 \neq w$，使得 $v \to v_0$ 且 $v_0 \to^\star v'$。我们可以在 c' 上应用引理 6.2 中的前向传播，找到一个全定义且良定义的配置 c''，使得 $c''(v_0) \geqslant c''(v) = c'(v)$，并且对于所有不在 $\downarrow v$ 中的节点 x，都有 $c''(x) = c(x)$。因此，$c''_{|N_2}$ 在 (N_2, m_2) 中可达。由于配置 c'' 是良定义的，我们有（条件 (CEq)）$c''(w) \geqslant c''(v)$。我们分三种情况讨论：

- 或者 $v_0 \notin \downarrow w$ 并且 $w \notin \downarrow v_0$，根据引理8.4，我们得出 $v'\,\mathcal{C}\,w'$ 对每个节点 $v' \in \downarrow v_0$均成立。
- 或者 $v_0 \in \downarrow w$：根据假设，由于 $v' \notin \downarrow w$ 和 $v_0 \to^\star v'$，这种情况不可能发生。
- 或者 $w \in \downarrow v_0$：通过应用引理8.4末尾的相同证明，我们可以证明这种情况会导致非安全标识，因此应予排除。

因此，对于所有 $v' \in \downarrow v \setminus \downarrow w$ 以及所有 $w' \in \downarrow w$，我们都有 $v'\,\mathcal{C}\,w'$。

引理 8.6。如果 $v\,\mathcal{C}\,w$ 满足 $v \notin \downarrow w$ 和 $w \notin \downarrow v$，那么以下两个条件之一成立。

(冗余) 存在一个非死节点 w_0 $v\ w$ $w\ v$ v_0 $v_0 \to\bullet$ $\downarrow v_0 \setminus \downarrow w_0$ $\times \downarrow w_0$，使得 和 (,) 或 (,) 之一属于 ()。

(互异) 存在一对互异的根节点(v_0, w_0) 使得 $v_0 \mathcal{C} w_0$ 与 $v \in \downarrow v_0$ 和 $w \in \downarrow w_0$。

证明：假设 $v \mathcal{C} w$。那么存在一个全定义配置 c，使得 $c_{|N_2} \in R(N_2, m_2)$，且 $c(v)\ c(w) = 1$（这些网是安全的）。根据引理 6.2 中的反向传播性质，存在两个根节点 v_0 和 w_0，使得$c(v_0) = c(w_0) = 1$ 与 $v \in \downarrow v_0$ 和 $w \in \downarrow w_0$。我们需要考虑两种情况：

· 要么 $v_0 \neq w_0$，即条件 (互异)。

· 要么有 $v_0 = w_0$。我们证明，必定存在一个节点v_1，使得 $v_0 \to^\star v_1$ 且 $v_1 \to\bullet w_1$，并且 (v, w) 或 (w, v) 位于 $(\downarrow v_1 \setminus \downarrow w_1) \times \downarrow w_1$ 中。我们用反证法证明这一结论。事实上，如果不存在这样的节点，那么从 v_0 出发，只沿着 A 中的边（聚合弧），就能到达 v 和 w。考虑 $v_0 \circ\!\!\to Y$，存在两个节点 $v', w' \in Y$，使得 $v \in \downarrow v'$ 和 $w \in \downarrow w'$。由于 c c $=$ 0 $c\ v'\ = 0$ v' 良定义，根据 (CEq)，要么 $(v') = 0$，要么 (w')。取 ( ) 以及从 到 v 的聚合路径，其中 $v' \circ\!\!\to a_0 \circ\!\!\to \cdots \circ\!\!\to a_n = v$ 与 $n \in \mathbb{N}$。根据这条路径上的归纳法，对所有 $i \in 0..n$，必有 $c(a_i) = 0$，因为 c 是良定义的，而且一个节点只能有一个父节点（条件 (T3)）。因此， $c(v') = 0$，这与 $v \mathcal{C} w$ 矛盾。

8.2.3 算法的可靠性与完备性

定理 8.7。 如果C 是调用$Matrix([\![E]\!], \|)$返回的矩阵，其中 $\|$ 表示 $[\![E]\!]$ 的根节点之间的并发关系（意为 N_2和常量），那么对于所有节点 v, w，有 $v \mathcal{C} w$ 当且仅当 $\mathrm{C}[v, w] = 1$。

证明： 首先需要指出，调用 $Matrix([\![E]\!], \|)$ 总会终止，因为唯一的递归（传播）遵循有向无环图结构。我们将证明分为两种情况：首先证明非死节点的计算（C 的对角线以及 $\mathcal{C}$的非死节点）是可靠且完备的。接着，证明不同节点对的可靠性与完备性。

非死库所，C 的对角线：

（完备性）如果某个节点 v满足 $v \mathcal{C} v$，那么根据引理 8.3，存在一个非死根节点 v_0，满足 $v \in \downarrow v_0$。因此，算法 3 在第 5 行以 v_0为参数调用传播。随后，算法 4 第 8 行递归访问 $\downarrow v_0$ 中的所有节点，包括 v。因此，$\mathrm{C}[v, v]$ 在第 4 行被设为 1（并且一直保持为 1，直到算法结束，因为初始化的第 2 行之后，算法中没有任何一行会将 C 的值设为 0）。至此，非死库所的完备性部分得证。

（可靠性）反过来，假设对于某个节点 v，有 $\mathrm{C}[v, v] = 1$。我们考虑三种子情况：

· $\mathrm{C}[v, v]$ 在第10 行或第11 行的算法3中被设置：这意味着存在两个互异的并发根节点 v_0 和 w_0，使得 $v \in \downarrow v_0 \cap \downarrow w_0$。因此， v_0 是非死的（w_0也如此）。由此可知，根据引理，是非死的8.2。

· $\mathrm{C}[v, v]$ 在算法4 的第4行被设置：算法3第4 行和算法4第7 行的循环确保只对非死根节点的后继调用Propagate $[\![E]\!]$。因此， v 属于 $\downarrow v_0$ 中的某个非死根节点 v_0，故根据引理 $v \mathcal{C} v$，结论8.2成立。

· $\mathrm{C}[v, v]$ 在算法4 的第12 行或第13行被设置：该子情形不可能发生，因为这些行只考虑由互异节点构成的二元组(v', w')。

总结第一种情况，关于非死库所和 C 的对角线，该算法是可靠且完备的。**并发库所：**

（完备性）我们假设 $v \mathcal{C} w$ 对于两个互异节点 v 和 w成立。这意味着 v 和 w 均非死，即 $v \mathcal{C} v$ 和 $w \mathcal{C} w$。如果 $v \in \downarrow w$，则$\mathrm{C}[v, w]$在算法4 的第4 行或第5 行被设置为1；同样地，若$w \in \downarrow v$也是如此，这正是预期结果。因此，我们现在假设 $v \notin \downarrow w$ 和 $w \notin \downarrow w$，从而引理8.6 适用。我们考虑该引理中的两种情形：

· （冗余）：此时，$\mathrm{C}[v, w]$ 被设为1，位于算法4的第12 行或第13行。

· （互异）：此时$\mathrm{C}[v, w]$ 被设为1，位于算法3的第10 行或第11行。

至此，关于并发库所的算法完备性证明结束。

（可靠性）我们假设，对于某些互异节点 v、 w，有 $\mathrm{C}[v, w] = 1$。我们考虑三种子情况：

· $\mathrm{C}[v, w]$ 是在算法3的第10 行或第11行设置的：根据引理8.4，我们得出 $v \mathcal{C} w$ 成立。

· $\mathrm{C}[v, w]$ 是在算法4 的第4 行或第5行设置的：根据引理8.2。

· $\mathrm{C}[v, w]$ 设置在第 12 或 13 行的算法 4：由引理 8.5可得结论。

至此，关于并发库所的算法可靠性证明完毕。

因此，该算法对于非死库所和并发库所是可靠且完备的。

8.3 对不完整并发关系的扩展

使用我们的方法时，我们只会向并发矩阵 C 中写入 1。这已经足够，因为我们确切知道关系 $\|_2$，而在这种情况下，关系 $\|_1$ 也必定是完备的（C 中只能有 0 或 1）。这一点从 C 初始化时各处均为 0 这一事实中显而易见。我们可以扩展算法，使其支持我们只掌握 $\|_2$ 的部分知识的情况。具体做法是用特殊值 • （未定义）初始化 C，并添加一些规则，使我们能够在 TFG 上“传播 0”，这与完整算法只传播 1 的方式相同。例如，我们知道，如果 $\mathrm{C}[v, w] = 0$ (v，w 非并发）且 $v \circ\!\!\to w'$（我们知道，在可达配置上始终有 $c(v) \geqslant c(w')$），那么 $\mathrm{C}[w', w] = 0$ 必然成立。同样，我们可以证明以下传播“死节点”的规则是可靠的：如果 $X \to\bullet v$，并且对于所有 $w \in X$，$\mathrm{C}[w, w] = 0$（节点 w 是死的），那么 $\mathrm{C}[v, v] = 0$。

关于并发关系的部分知识可能很有用。事实上，许多应用场景能够处理部分知识，或者只依赖非并发关系（即并发矩阵中的 0）。例如，在计算 NUPN 划分时，总可以安全地将一个 • 替换为 1。这也意味着，知道两个库所不并发，往往比知道它们并发更有价值；0 比 1 更有用。

我们基于这一思路，为不完整矩阵的情形扩展了算法，并报告了一些使用该算法获得的结果。遗憾的是，篇幅所限，我们无法在此描述完整算法。它比完整情形的算法稍复杂一些，并基于另外六条公理。虽然我们可以证明该算法是可靠的，但完备性有着不同的含义：我们证明，当节点 p和 q是根节点 v_1和 v_2的后继节点，且对于所有 $i \in 1..2$都有 $\mathrm{C}[v_i, v_i] \neq \bullet$ 时，必然有 $\mathrm{C}[p, q] \neq \bullet$。

¯下文中，我们使用记号 $v \mathcal{C} w$ 表示 $\neg$（vCw）；这意味着 v、 w 根据 C 是非并发的。按照我们的¯记号， $v \mathcal{C} v$ 表示 v 是死节点：不存在良定义的可达配置 c，使得 c（v） > 0。

8.3.1 死节点的传播

我们证明，死节点 v 必然与所有其他节点非并发。此外，如果某个节点的所有“直接后继”都是死节点，那么该节点也是死节点。

引理 8.8. 设 v 是 $[\![E]\!]$ 中的一个节点。若 $v\bar{\mathcal{C}}v$，则对于 $[\![E]\!]$ 中的所有节点 w，都有 $v\bar{\mathcal{C}}w$。

证明： 设 $v\bar{\mathcal{C}}v$。那么，对于任意满足 $c_{|N_2}$ 的全定义良定义配置 c，若 $c_{|N_2}$ 在 (N_2, m_2) 中可达，则有 $c(v) = 0$。根据并发关系 $\mathcal{C}$ 的定义，v 不可能与任何节点并发。

引理 8.9. 设 v 是 $[\![E]\!]$ 中的一个节点，且满足 $v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$。那么，$v\bar{\mathcal{C}}v$ 当且仅当对于 X 中的所有节点 w，都有 $w\bar{\mathcal{C}}w$。

证明： 我们用矛盾法证明两个方向。

设那么存在$\bar{\mathcal{C}}v$并取满足配置$\mathcal{C}w$的使得$X(w) > 0$.必然，由于 $v\mathcal{C}v$，我们有 $c(v) = 0$，这与 (CEq) 矛盾。

接下来，假设 $v\mathcal{C}v$ 且对于每个节点 $w \in X$，都有 $w\bar{\mathcal{C}}w$。那么存在一个全定义配置 c，使得 $c(v) > 0$。必然，对于所有节点 $w \in X$，我们有 $c(w) = 0$，这也与 (CEq) 矛盾。

这些性质蕴含以下三个公理的可靠性：

1. 如果 $\mathsf{C}[v, v] = 0$，那么对于 $[\![E]\!]$ 中的所有节点 w，都有 $\mathsf{C}[v, w] = 0$。 2. 如果 $v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$，且对于所有节点 $w \in X$ 都有 $\mathsf{C}[w, w] = 0$，那么 $\mathsf{C}[v, v] = 0$。 3. 如果 $v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$，且 $\mathsf{C}[v, v] = 0$，那么对于所有节点 $w \in X$，都有 $\mathsf{C}[w, w] = 0$。

8.3.2 兄弟节点之间的非并发性

我们证明，同一节点的直接后继彼此非并发（对于安全网而言）。这基本上是以下事实的推论：$c(v) = c(w) + c(w') + \ldots$ 和 $c(v) \leqslant 1$ 意味着，在配置固定时，$c(w)$ 和 $c(w')$ 中至多有一个等于 1。

引理 8.10. 假设 v 是 $[\![E]\!]$ 中的一个节点，且满足 $v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$。对于 X 中的任意一对节点 w, w'，我们有：$w \neq w'$ 意味着 $w\bar{C}w'$。

证明：采用反证法。取 X 中的一对互异节点 w, w'，并假设 $w\mathcal{C}w'$。那么存在一个全定义配置 c，使得 $c(w) = 1$ 且 $c(w') = 1$，其中 $c_{|N_2}$ 在 (N_2, m_2) 中可达。由于 c 必须满足 (CEq)，我们有 $c(v) \geqslant 2$，这与我们的网是安全网这一事实矛盾，参见引理 8.1。

这一性质意味着以下公理是可靠的：

4. 若 $v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$，则对于所有满足 $w \neq w'$ 的节点对 $w, w' \in X$，$\mathsf{C}[w, w'] = 0$。

8.3.3 遗传性与非并发

我们证明，若 v 和 v' 非并发，则 v' 必定与 v 的所有直接后继节点非并发（反之亦然）。这基本上是以下事实的推论：$c(v) = c(w) + \ldots$ 且 $c(v) + c(v') \leqslant 1$ 意味着 $c(w) + c(v') \leqslant 1$。

引理 8.11. 设 $[\![E]\!]$ 中的节点 v 满足 $v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$。那么对于每个节点 v' 使得 $v\bar{\mathcal{C}}v'$ 我们还有 $w\bar{\mathcal{C}}v'$ 对于 X 中的每个节点 w。反之，如果 $w\bar{\mathcal{C}}v'$ 对于 X 中的每个节点 w，那么 $v\bar{\mathcal{C}}v'$。

证明： 我们分别通过反证法证明每个性质。

假设 $v\bar{\mathcal{C}}v'$ 并取 $w \in X$，使得 $w\mathcal{C}v'$。那么存在一个全定义良定义的配置 0 $v\bar{\mathcal{C}}v'$ $c\ v\ = 0$ $c\ v'\ = 0$，使得 $(\ )$ (v')。必然地，由于我们必须有 $(\ )$ 或 $(\ )$。

我们已经知道 $c(v') > 0$，因此 $c(v) = 0$，这与 (CEq) 矛盾，因为 $w \in X$。

接下来，假设对于所有节点 $w\bar{\mathcal{C}}v'$ $w \in X$，并且我们有 $v\mathcal{C}v'$。于是存在一个全定义配置 c，使得 $c(v)$ $c(v') > 0$。必然地，对于所有节点 $w \in X$，我们都有 $c(w) = 0$ 或 $c(v') = 0$。我们已经知道 $c(v') > 0$，因此对于所有节点 $w \in X$，$c(w) = 0$，这同样与（CEq）矛盾。

这些性质蕴含以下两个公理的可靠性：

5. 如果对于所有节点 $w \in X$，$v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$ 且 $\mathsf{C}[w, v'] = 0$，那么 $\mathsf{C}[v, v'] = 0$。6. 如果在 X 中，对于所有节点 w，$v \circ\!\!\to X$ 或 $X \to\!\!\bullet\, v$ 且 $\mathsf{C}[v, v'] = 0$，那么 $\mathsf{C}[w, v'] = 0$。

9 实验结果

我们实现了一款名为 Kong 的新工具，用于 Koncurrent places Grinder，负责执行我们在第 7 节和 8节中介绍的“逆变换”。该工具采用 GPLv3 许可证开源，可在 GitHub 上免费获取 [1]。

我们使用模型检验竞赛（MCC）提供的大型模型数据库 [4, 15] 来测试我们的方法。Kong 的输入是使用 Petri 网标记语言（PNML） [16], 或嵌套单元 Petri 网（NUPN）格式 [11] 定义的佩特里网。

我们不直接使用 Kong 计算网约简，而是依赖另一款名为 Reduce 的工具，该工具由 Tina 工具箱 [22]开发。我们使用的是 Kong 2.0 版和 Tina 工具箱 3.7 版。

9.1 工具链说明

我们介绍用于标识可达性判定过程和并发矩阵计算的工具链。

图 5 展示了用于检查给定标识 m_1'是否在输入网 (N_1, m_1) 中可达的工具链。在此情况下，标识 m_1' 在输入文件中以简单的文本格式定义。工具 Kong 会获取使用 E计算得到的缩减系统，并将其用于把 Reduce 投影到标识 m_1' 中，得到标识 m_2'（如果可行）。如果投影返回错误，我们就知道 m_1' 不可能可达。否则，我们会调用一个辅助工具，此处使用的是 Sift，以探索(N_2, m_2) 的状态空间，并尝试找到标识 m_2'。

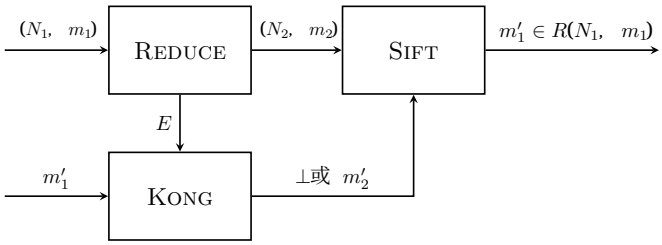


图 5：可达性判定过程的工具链。

我们在图6中介绍了第二套工具链。使用Reduce计算出多面体约简后，我们使用（N_2, m_2）通过cæsar.bdd计算约简网的并发矩阵。cæsar.bdd 是 CADP 工具箱 [9, 18]的一部分。我们的实验结果使用cæsar.bddv3.6 版计算得出；该工具是 CADP 2022-b “Kista”版的一部分，该版本于2022年2月发布。工具Kong 接收该并发关系（记作 $\|_2$）以及约简系统 E，随后重建初始网上的并发关系。

[1] https://github.com/nicolasAmat/Kong

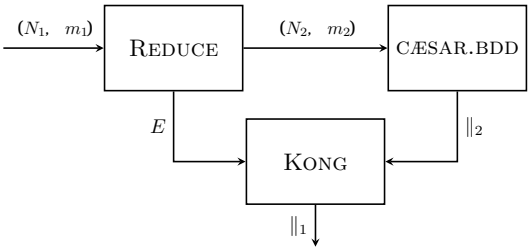


图 6：并发加速算法的工具链。

9.2 基准测试与约简率分布

我们的基准测试基于 MCC 2020 竞赛中使用的 102 个模型集合。大多数模型都带参数，因此同一模型可能有多个实例。这些佩特里网约有 1000 个实例（其中 588 个是安全的），规模差异很大，库所数量从 9 个到 50 000 个不等，变迁数量则从 7 个到 200000 个不等。总体而言，该集合提供了大量示例，具有各种结构和行为特征，涵盖了范围广泛的用例。

9.2.1 约简率分布

由于我们关注网中可实现的缩减程度，因此我们计算了所有实例的约简率（r），该比率由 REDUCE 得出（见图 7a）。该比率的计算方法是：可移除库所的数量除以初始网中的库所数量。比率为 100%（$r = 1$）表示该网已完全缩减；剩余网中没有库所，其 TFG 中的所有根节点都是常数。

我们可以看到，采用我们的方法后，有数量出人意料之多的模型，其规模缩减至不足原来的一半（约 25% 的实例的比率为 $r \geqslant 0.5$），其中大约一半的实例可以达到 30% 或更高的约简率。我们考虑约简率的两种取值：一种对应于产生良构 TFG 的约简（浅橙色），另一种对应于使用 REDUCE 所能实现的最佳约简（深蓝色），例如 SMPT模型检查器 [1, 2]中所采用的约简。对于安全网，也可以观察到相同的趋势（图 7b）。

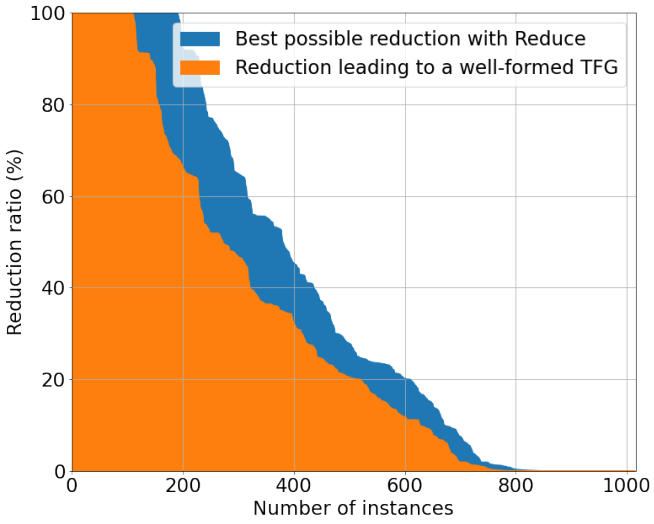
我们还注意到，由于良构约束，我们仅失去了少数约简网的机会。实际上，我们主要失去的是简化某些“部分”标识图的能力；利用抑制弧或弧上的权重可以对这些标识图进行约简（这两项功能均不受 cæsar.bdd 支持）。

9.2.2 标识可达性基准测试

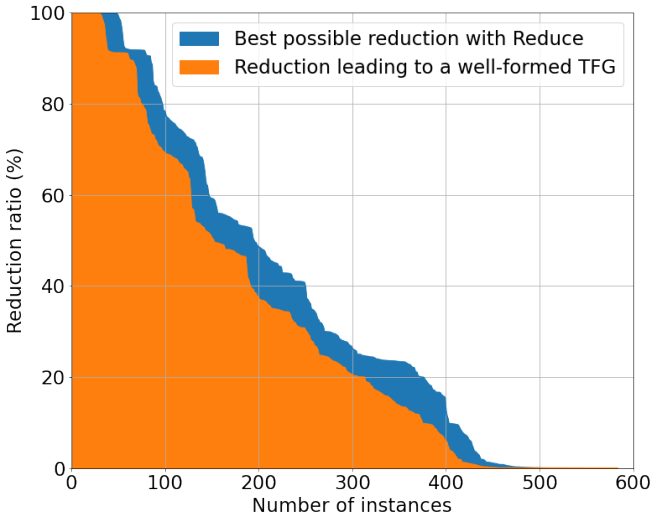
我们从约简率大于 1% 的实例中选取了 426 个佩特里网，用于评估 Kong 在标识可达性问题上的性能。为避免实例数量较多的模型造成偏差，我们从每个模型中选取了至多 5 个约简率相近的实例。对于每个实例，我们生成了 5 个查询，这些查询是通过在该网的状态空间上进行“随机游走”找到的标识（为此，我们使用了 Tina 发行版中的工具 Walk）。我们对每个查询分别运行 Kong 和 Sift，时间限制为 5 分钟。

9.2.3 并发关系基准测试 ion

我们在约简率大于 1% 的 424 个安全 Petri 网实例上评估了 Kong 的性能。我们对每个实例都运行了 Kong 和 cæsar.bdd，采用两种主要模式：首先，设置 15 分钟的时限，比较完全求解的实例数量（即工具计算出完整并发矩阵的实例）；其次，设置



(a) 所有实例



(b) 安全实例

图 7：约简率的分布：(a) MCC 中的所有实例；(b) MCC 中的所有安全实例。

60 秒的超时时间，比较部分矩阵中计算出的数值（填充率）。使用 cæsar.bdd 计算部分并发矩阵分为两个阶段：首先是用户可以中止的“BDD 探索”阶段；接着是无法中止的后处理阶段。实际上，这意味着在初始网上的执行时间通常比在约简后的网更长：cæsar.bdd 的平均计算时间约为 48 秒，而 Kong 不到 12 秒。在每次测试中，我们都将 Kong 的输出与使用 cæsar.bdd 在初始网上得到的数值进行了比较，结果的可靠性达到 100%。

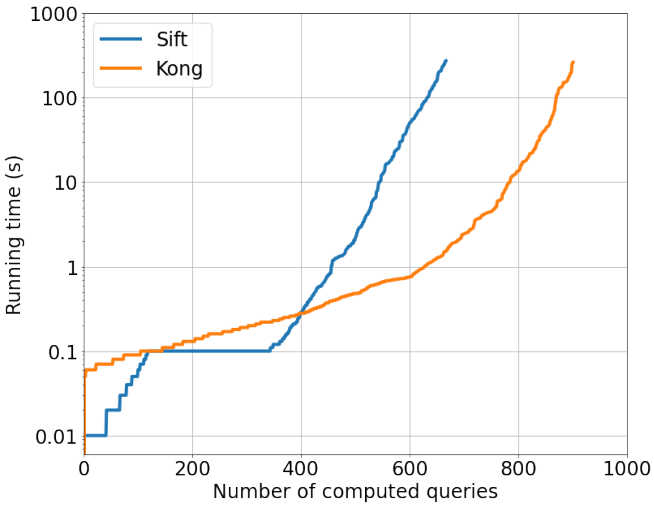
接下来，我们将详细介绍实验所得结果，并分析使用约简的影响。

9.3 标识可达性结果

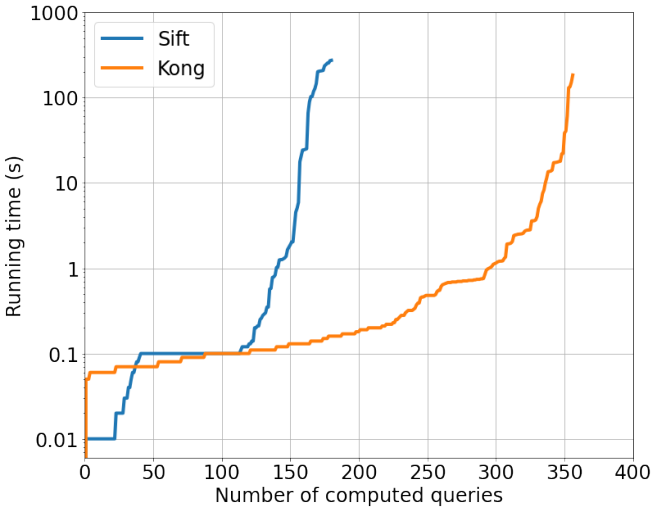
我们在图8的图表中展示结果，比较在使用和不使用约简的情况下，计算给定数量查询所需的时间。（请注意，时间采用对数刻度。）我们考察了两组不同的实例样本。首先，仅考察约简率较高的实例（区间为 $[0.5, 1]$），然后考察全部实例。

我们观察到，使用约简时具有明显优势。例如，对于约简率处于区间 $[0.5, 1]$ 内的实例，在 5 分钟的时间限制下，我们几乎

将已计算查询的数量翻了一番（仅使用 Sift 时为 181 个，而使用 Kong 时为 357 个）。相反，当运行时间低于 0.1 秒时，仅使用 Sift 的小幅优势可以解释为：我们将 Reduce 的运行时间计入了 Kong 的运行时间。



(a) $r \in]0,\ 1]$



(b) $r \in [0.5, 1]$

图8：计算给定数量可达性查询所需的最短运行时间：(a) 所有实例；(b) 满足 $r \in [0.5, 1]$ 的实例。

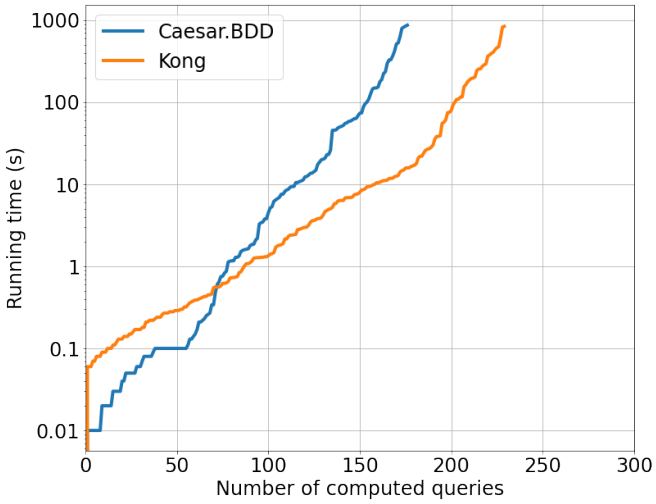
9.4 完整计算的并发矩阵

我们的下一组结果针对完整矩阵的计算，超时时间为 15 分钟。下表列出了已计算的实例数量。我们将结果按实例分为三个类别：低/一般/高，分别对应不同的比率范围。我们发现，采用约简比不采用约简能计算出更多结果（+30%）。正如预期，高类别的收益更大（+86%），但一般 实例的收益也相当显著（+28%）。

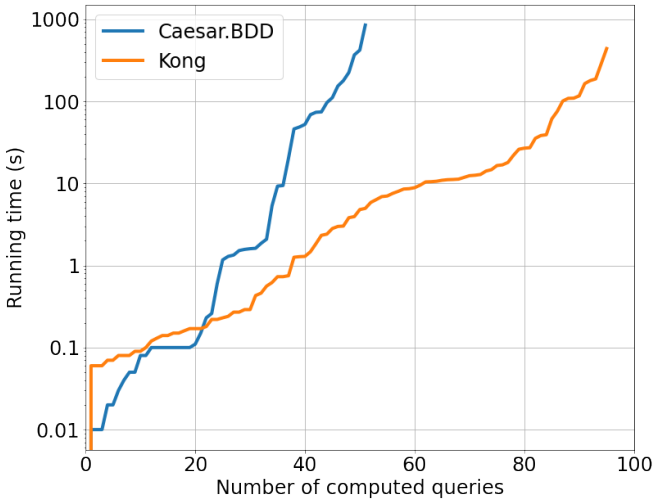
与前一种情况一样，我们研究使用 Kong 所获得的加速比，方法是绘制图表比较计算给定数量实例所需的时间；参见图 9。

9.5 部分矩阵的结果

即使结果不完整，我们也可以比较所用方法的“准确性”。为此，我们在 cæsar.bdd 上设置 60 秒的超时时间来计算并发关系。



(a) $r \in]0,\ 1]$



(b) $r \in [0.5, 1]$

图 9：计算给定数量并发矩阵所需的最短运行时间：(a) 所有实例，(b) 包含 $r \in [0.5, 1]$ 的实例。

我们比较采用和未采用规约时得到的 填充率 。对于一个有 n 个库所的网，该比率由公式 $2\,\mathrm{C}/(n^2 + n)$ 给出，其中 C 是矩阵中 0 和 1 的数量。

我们使用线性尺度的散点图展示结果，见图 10。我们观察到，几乎所有数据点都位于对角线的一侧，这意味着在这种情况下，约简会增加计算值的数量；有许多示例（图中的顶行）仅使用约简就能在 60 秒内计算出完整关系。该图无法区分 1 和 0，但当我们仅考虑并发库所（1）或仅考虑非并发库所（0）的填充率时，也能获得类似的良好结果。

10 结论与后续工作

我们提出了一种新方法，将可达性问题的计算从初始“高维”域（初始网中的库所集合）转移到更小的域（约简网中的库所集合）。我们的方法结合了结构约简与线性方程，这一思路最早由 [6, 7] 提出。

本研究的主要贡献是定义了一种新的数据结构，能够精确捕捉这些线性方程的结构，我们称之为令牌流图（TFG）。我们展示了如何利用 TFG 加速

缩减比率 (r)		# 测试用例	# 计算矩阵		
			KONG	cæsar.bdd	
Low	$r \in]0, 0.25[$	160	85	87	$\times 0.97$
Fair	$r \in [0.25, 0.5[$	112	49	38	$\times 1.28$
High	$r \in [0.5, 1]$	152	95	51	$\times 1.86$
总计	$r \in]0, 1]$	424	229	176	$\times 1.3$

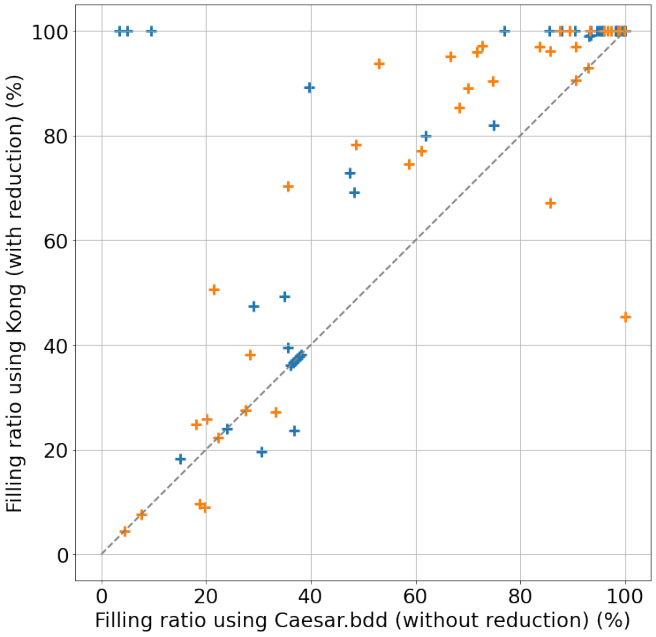


图10：比较部分矩阵的填充率，Kong（y 轴）与 cæsar.bdd（x 轴），针对具有 $r \in [0.25, 0.5[$（浅橙色）和 $r \in [0.5, 1]$（深蓝色）的实例。（计算时 BDD 探索的超时时间设为 60 秒。）

标识可达性问题的求解，并用于计算完整和部分情况下的并发关系。

我们有一些想法，探讨如何将 TFG 应用于其他问题以及如何扩展它们。一个自然的应用是模型计数（这是我们最初的目标，见 [6]），TFG 或可为我们所处理的线性方程组中（整数）解的数量计数带来新的算法。

另一种可能的应用是最大标识 问题，即在所有可达标识中求表达式 $\sum_{p \in P} m(p)$ 的最大值。对于安全网，这相当于求能够同时被标识的库所的最大数量。我们可以轻松调整算法来计算这个值，甚至可以调整算法，以便在网不安全时也能计算结果。

我们甚至可以处理一个更一般的问题，它与库所集合的最大并发性质有关。若存在一个可达的 m，使得对 S 中的所有库所 p，$m(p) > 0$，则称集合 S 为并发的。（这涵盖了库所对和单个库所的情况。）若不存在并发的超集 $S' \supsetneq S$，则称集合 S 为最大并发的。

计算一个网的最大并发集合有几个方面的意义。首先，它为并发关系提供了一种替代表示，有时可以更节省空间：(1) 最大并发集合构成了网的库所集合的唯一覆盖；(2) 当且仅当存在一个满足 $\{p, q\} \subset S$ 的 S 最大并发集合时，$p \parallel q$ 成立。显然，对于安全网，最大并发集合中规模最大的集合的大小就是最大标识问题的答案。

未来的工作中，我们希望回答更具挑战性的问题，例如证明广义互斥约束 [14]，，这需要检查涉及库所标识加权和的不变量，其形式为 $\sum_{p \in P} w_p.m(p)$。另一个可能的扩展是支持非普通网（这需要为 TFG 的弧添加权重）以及不安全网（使用我们当前的方法已经可以处理，但需要更改算法中使用的一些“公理”）。

致谢。 我们谨向 Pierre Bouvier 和 Hubert Garavel 致谢，感谢他们提出的宝贵建议，这些建议帮助提升了本文的质量。

参考文献

[1] Nicolas Amat、Bernard Berthomieu 和 Silvano Dal Zilio。关于将多面体抽象与基于 SMT 的模型检测相结合以分析 Petri网。载于*Petri* 网与并发的应用与理论（*Petri Nets*），第 12734 卷*LNCS*。Springer，2021年。

[2] Nicolas Amat、Bernard Berthomieu 和 Silvano Dal Zilio。佩特里网的多面体抽象及其在基于 SMT 的模型检测中的应用。*Fundamenta Informaticae*，187(2-4)，2022年。

[3] Nicolas Amat、Silvano Dal Zilio 和 Didier Le Botlan。使用约简加速死库所和并发库所的计算。载于软件模型检测（*SPIN*），第 12864 卷*LNCS*。Springer，2021年。

[4] Elvio Amparore、Bernard Berthomieu、Gianfranco Ciardo、Silvano Dal Zilio、Francesco Gallà、Lom Messan Hillah、Francis Hulin-Hubard、Peter Gjøl Jensen、Loïg Jezequel、Fabrice Kordon、Didier Le Botlan、Torsten Liebke、Jeroen Meijer、Andrew Miner、Emmanuel Paviot- Adet、Jiří Srba、Yann Thierry-Mieg、Tom van Dijk 和 Karsten Wolf。介绍第九届模型检验竞赛。载于系统构造与分析工具和算法（*TACAS*），第 11429 卷*LNCS*。Springer，2019年。

[5] G. Berthelot。《网的变换与分解》。载于*Petri* 网：核心模型及其性质，第 254 卷*LNCS*。Springer，1987年。

[6] Bernard Berthomieu、Didier Le Botlan 和 Silvano Dal Zilio。用于计数标识的佩特里网约简。载于软件模型检测（*SPIN*），第 10869 卷*LNCS*。Springer，2018年。

[7] Bernard Berthomieu、Didier Le Botlan 和 Silvano Dal Zilio。从归约方程计算佩特里网标识。国际软件技术转移工具期刊，2019年。

[8] Frederik M Bønneland、Jakob Dyhr、Peter G Jensen、Mads Johannsen 和 Jiří Srba。佩特里网的顽固约简与结构约简。《程序设计中的逻辑与代数方法》，102，2019年。

[9] Pierre Bouvier 和 Hubert Garavel。在安全 Petri 网中解决三个可达性问题的高效算法。载于*Petri* 网与并发的应用与理论（*Petri Nets*），LNCS 第 12734 卷。Springer，2021年。

[10] Pierre Bouvier、Hubert Garavel 和 Hernán Ponce-de León。将 Petri 网自动分解为自动机网络——综述。载于 Petri 网与并发的应用与理论（Petri Nets），第 12152 卷。Springer，2020年。[11] Hubert Garavel。嵌套单元 Petri 网。《逻辑与代数编程方法杂志》，104：60–85，2019年4月。[12] Hubert Garavel。关于为 Petri 网模型检验器添加实用功能的提案。研究报告 03087421，Inria Grenoble - Rhône-Alpes，2020年12月。[13] Hubert Garavel 和 Wendelin Serwe。进程代数规约的状态空间缩减。载于代数方法论与软件技术，LNCS 第 3116 卷。Springer，2004年。[14]Alessandro Giua、Frank DiCesare 和 Manuel Silva。具有不可控变迁的网上的广义互斥约束。载于 IEEE 国际系统、人类与控制论会议。IEEE，1992年。[15]Lom-Messan Hillah 和 Fabrice Kordon。Petri网资源库：用于基准测试和调试 Petri 网工具的工具。载于 Petri 网与并发的应用与理论（Petri Nets），LNCS 第 10258 卷。Springer，2017年。[16] Lom-Messan Hillah、Fabrice Kordon、Laure Petrucci 和 Nicolas Treves。PNML框架：Petri 网标记语言的可扩展参考实现。载于 Petri 网应用与理论国际会议，LNCS 第 6128 卷。Springer，2010年。[17] Thomas Hujsa、Bernard Berthomieu、Silvano Dal Zilio 和 Didier Le Botlan。使用状态方程检查 Petri 网子类中的标识可达性。技术报告 20278，LAAS-CNRS，2020年11月。[18] INRIA。CADP。https://cadp.inria.fr/，2020年。[19] Ryszard Janicki。网、顺序组件与并发关系。《理论计算机科学》，29(1-2)，1984年。[20] A. V. Kovalyov. 佩特里网的并发关系与安全性问题。载于《Petri 网与并发的应用与理论》，LNCS 第616卷，柏林、海德堡，1992年。Springer。[21] Andrei Kovalyov. 计算正规 STG并发关系的多项式算法。载于《硬件设计与佩特里网》。Springer，波士顿，马萨诸塞州，2000年。[22] LAAS-CNRS。Tina 工具箱。http://projects.laas.fr/ tina，2020年。[23] Richard J. Lipton. 缩减：证明并行程序性质的一种方法。《ACM 通讯》，18(12)，1975年。[24] Tadao Murata. 佩特里网：性质、分析与应用。《IEEE 会议录》，77(4)，1989年。[25] A. Semenov 和 A. Yakovlev. 结合偏序与符号遍历，高效验证异步电路。载于 ASP-DAC’95/CHDL’95/VLSI’95 与 EDA Technofair 会议论文集，1995年。[26] Manuel Silva、Enrique Terue 和 José Manuel Colom. 库所/变迁网系统分析的线性代数与线性规划技术。载于《佩特里网讲义 I：基本模型：佩特里网进展》，LNCS 第1491卷。Springer，1996年。[27] Yann Thierry-Mieg. 重新审视结构约简。载于《Petri 网与并发的应用与理论（Petri Nets）》，LNCS 第12152卷。Springer，2020年。[28] Remigiusz Wisniewski、Andrei Karatkevich、Marian Adamski、Aniko Costa 和 Luis Gomes. 应用佩特里网与可比图对并发控制系统进行原型设计。《IEEE控制系统技术汇刊》，26(2)，2018年。[29] Remigiusz Wiśniewski、Monika Wiśniewska 和 Marcin Jarnut. C-精确超图在由安全 Petri 网描述的信息物理系统的并发性与顺序性分析中的应用。《IEEE Access》，7，2019年。