arXiv:2512.13538v1 [cs.GT] 15 Dec 2025

分布式库所与安全网约简

Victor Khomenko

瑞萨电子，英国斯温顿 *Welton Rd，Delta 200 Office Park*，邮编 *SN5 7XB victor.khomenko.xh@renesas.com*

Maciej Koutny

计算机学院，纽卡斯尔大学，英国泰恩河畔纽卡斯尔 *Science Square 1*号，邮编 *NE45TG maciej.koutny@ncl.ac.uk*

Alex Yakovlev

纽卡斯尔大学工程学院，*Merz Court*，泰恩河畔纽卡斯尔，*NE1 7RU*，英国 *alex.yakovlev@ncl.ac.uk*

摘要。 能够找到与并发系统形式化规约具有相同行为的小型Petri网，有助于有效验证此类系统并将其付诸实际实现。本文研究以组合式定义的安全网形式给出的规约。

本文探讨了一个新颖的“分布式库所”概念，用以实现单个网库所的行为。研究表明，如果分布式库所覆盖一个安全Petri网，那么可以删除某些库所而不改变其行为。关键在于，这种化简既是静态的，也是局部的，因此在实践中具有计算可行性。

随后，将所得化简技术应用于安全Petri网（箱）的代数；这些箱由进程（盒）表达式以组合方式导出。尽管原始推导可能产生规模呈指数级增长的箱，但先前研究表明，若盒表达式不涉及循环行为，指数级数量的库所可化简至多项式级别（二次）。在

通信地址：maciej.koutny@ncl.ac.uk

本文中，我们使用分布式库所，证明即使对于包含迭代的进程表达式，也能实现类似的优化。

关键词： 安全网、分布式库所、静态约简、局部约简、盒表达式、控制流、组合、连接图、余图、边团覆盖

1. 引言

Petri网是一种形式化建模技术，旨在处理并发和分布式计算系统。它支持简单而富有表现力的语义、直观的图形表示法，并能够简洁地捕获行为，而不会使后续的形式化验证或综合变得难以处理。Petri网有多种软件工具，并已广泛用于建模形式化语言，也被用作中间表示，将最初以其他形式化方法表达的设计转换为该表示。事实上，在过去四十年中，人们一直广泛致力于将进程代数、并发编程语言及其他形式化方法转换为Petri网，例如参见 [1, 2, 3, 4]。

盒代数 [1, 5] 为Petri网提供了一个通用的进程代数框架，称为（*Petri*）盒。通过组合定义的盒子可以由盒表达式 导出，这些盒表达式在通常的语言理论意义下是一种进程代数。例如，图1（a）展示了一个由盒表达式 $[a \,\square b * c;(d \parallel e) * f \parallel g]$导出的盒。该表达式规定了一个并发系统：首先执行$a$或$b$中的一个，随后执行一个（可能执行零次的）循环“$c$ 后跟d 和e并发执行”，最后通过并发执行f 和g 终止整个表达式。盒代数有多种具体实现，包括CCS [6] 和 TCSP [7]。一般而言，能够找到与并发系统形式化规约具有相同行为的小型Petri网，将有助于此类系统的高效验证和实际实现。本文研究以安全Petri网和盒表达式形式给出的规约。

能够创建简洁的系统模型，通常是 Petri网相较于有限状态机（FSMs）等更简单的形式化方法的主要优势；在有限状态机中，状态空间往往在建模阶段就已发生指数级爆炸 [8]。遗憾的是，正如我们在 [9], 中观察到的，即使将简单的控制流转换为 Petri网，也可能导致 Petri网规模指数级爆炸。当两个或更多并发且同时启用的动作组成的群组（“突发”）被置于互斥选择中时，往往会出现这种情况。举例来说，图1(b)中的网刻画了以下并发执行之间的选择：a_1 和 b_1, 、a_2和 b_2,，以及 a_3和 b_3（这对应于类似 $(a_1 \parallel b_1) \square (a_2 \parallel b_2) \square (a_3 \parallel b_3)$ 的盒表达式）。图1(b)中的标准构造使用了 $2^3 = 8$ 个库所，但如图1(c)所示，只用4个库所即可实现相同的行为。一般而言，要表示 n 对并发动作 a_i 和 b_i之间的选择，标准构造需要使用 2^n 个库所。然而，经过优化的构造只需使用其中一小部分库所。例如，图1(d)展示了 $n = 10$ 时的一种简化方案，它使用6个库所，而非 $2^{10} = 1024$ 个。

前人工作。 作为一个动机示例， [9]研究了突发自动机 [10] 在异步电路设计领域中的应用。它们类似于有限状态机，其弧标记为并发执行的集合

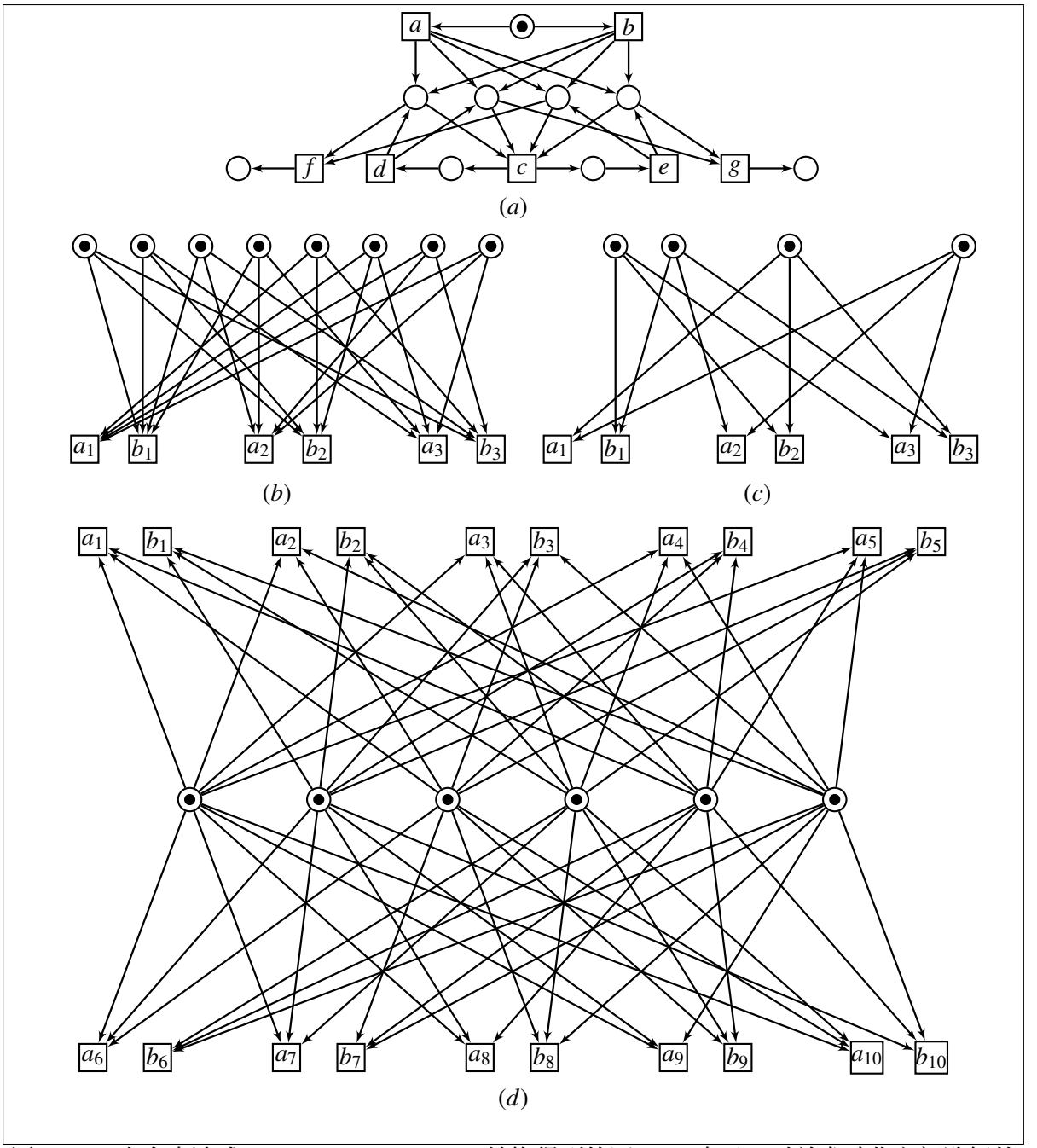

图1： (*a*) 由盒表达式 $[a \,\square\, b * c;(d \,\|e) * f \,\|g]$ 转换得到的网。(*b*) 表示三对并发动作之间选择的网：a_i 和 b_i，以及 (*c*) 展现相同行为的简化版本。(*d*) 表示在 10 对并发动作之间进行选择的标记网的简化版本：a_i 和b_i。

动作（突发）。在 [10], 中，有人提出了一种保持语言的线性规模转换：在每个突发前添加一个静默的“分叉”变迁，并在突发后添加另一个静默的“联结”变迁，以检测其是否完成。不幸的是，在某些情况下，静默变迁是不可接受的 [11]。首先，语言等价可能过于弱（例如，它无法保持分支时间时态属性，甚至无法保持死锁），而在突发前添加静默变迁会破坏强双模拟和弱双模拟 [6]。

为了保持强双模拟，常采用笛卡尔积，例如参见 [12, 1, 2, 13]。为了表示多个突发（即并发变迁的集合）之间的选择 B_1,$B2$,...,B_n，这种构造会创建一个库所集合，对应于笛卡尔积 $B_1 \times B_2 \times \cdots \times B_n$中的元组，导致Petri网的规模随突发数量呈指数级增长。例如， [10] 提出了基于笛卡尔积的从突发自动机到Petri网的转换，并保持弱双模拟或强双模拟。

论文 [9] 提出了一种替代笛卡尔积的方法，该方法使用的库所数量至多为二次级（相对于所有突发的总大小），即可表示突发之间的选择，从而将突发自动机到Petri网的转换规模从指数级 [10] 降至多项式级。此外，在某些情况下，只需对数数量的库所，与笛卡尔积方法相比可实现双指数级缩减。图1(*b*，*c*，*d*)展示了后一种情况：对于n 个二元突发之间的选择，我们可以（渐近地）将规模从 2^n 降至$O(logn)$，实现双指数级缩减。该技术基于这样一种等价关系：表示并发事件突发之间选择的建模问题，等价于寻找完全多部图的边团覆盖问题。

针对以进程表达式形式描述的并发系统规范， [14]将上述技术推广到了包含选择、并发和顺序算子（但不包含迭代）的盒表达式。我们提出了一种针对这类盒表达式的多项式转换，该转换保持强双模拟（事实上，它保证可达图同构，这是一种更强的等价关系）。所开发的转换是组合式的——为此，我们在盒代数 [1]中引入了接口图的概念，使得许多来自盒代数的结果都可以应用于此。

基于寻找某个边团覆盖 的补可约图（余图 *[15]*）而构造的方法（其中一些边可能已被视为“已覆盖”），所得到的转换中，创建的库所数量与覆盖中的团数相对应。由此很容易看出，至多多项式（平方）数量的团总是足够的（因为相关余图的顶点数量是线性的），因此得到的是一个多项式 Petri网。因此， [9, 14] 的结果表明，如果盒表达式不涉及循环行为，那么笛卡尔积构造所创建的指数级库所数量，即使在最坏情况下，也可以从指数级降至多项式级（平方）；在最佳（未退化）情况下，则可降至对数级。

在本文中，我们将 [14] 的结果扩展到包含迭代的进程表达式情形。为此，我们引入一种新颖的分布式库所 概念，它为 Petri网中的各个库所提供了一种建模表示（因此，每个分布式库所都可以抽象为单个网库所）。

本文所采用的方法。一般而言，本文探讨的问题可以表述如下：

给定以某种形式化语言表达的并发系统形式化规约，寻找

可能具有相同行为的最小 *Petri*网。

这类广泛问题中的两个具体实例，涉及使用 Petri网和进程代数制定的规约：

给定一个安全*Petri*网 $\mathcal{N}$，寻找可能最小的安全*Petri*网 $\mathcal{N}'$，使其具有与 $\mathcal{N}$ 相同的行为（例如，具有同构的可达图）。 **问题I**

给定一个进程代数表达式，寻找具有相同行为的可能最小安全*Petri*网。 **问题II**

为了求解问题I，寻找 $\mathcal{N}'$ 的一种简单方法可能是推导出 $\mathcal{N}$ 的顺序可达图$RG_{\mathcal{N}}$，然后应用专门针对$RG_{\mathcal{N}}$的Petri网综合技术，以生成类似于 [16, 17, 18]中那些的最小网解。尽管这种方法在理论上是可靠的，但在实践中几乎不可行，因为$RG_{\mathcal{N}}$的规模相对于 $\mathcal{N}$的规模而言往往呈指数增长。

鉴于完全一般地求解问题I的前景相当黯淡，人们可能会寻求有限但仍具有实际意义的解。具体而言，可以按照以下思路，致力于开发一种组合式解：

给定由组件 $\mathcal{C}_1,\ldots,\mathcal{C}_k$ 构造的安全*Petri*网 $\mathcal{N}$，求由（部分）组件的优化版本构造的、行为等价且可能最小的安全*Petri*网 $\mathcal{N}'$。此外，所有决策都应尽可能在“局部”作出。

作为这种方法的一个例子，考虑由顺序Petri网 $\mathcal{S}_1,\ldots,\mathcal{S}_k$构成的 $\mathcal{N}$，这些组件通过公共变迁上的粘合（同步）进行组合。在这种情况下，从每个 $\mathcal{S}_i$ 中删除一些库所，使其仍生成相同的发射序列，和/或删除一些 $\mathcal{S}_i$，便可得到 $\mathcal{N}'$，其行为与 $\mathcal{N}$相同。

例如，考虑图2(*a*)所示的Petri网 $\mathcal{N}_0$。它描述了一个系统：先（并发地）执行变迁a 和 b，然后（并发地）执行c和d，或（并发地）执行e 和f。 $\mathcal{N}_0$可唯一分解为八个顺序子网 $\mathcal{S}_3,\mathcal{S}_4,\ldots,\mathcal{S}_{10}$，如图2(*c*)所示。这些子网均无法按上述方式缩减。然而，可以移除其中四个顺序网，在组合中保留 $\mathcal{S}_4,\mathcal{S}_5,\mathcal{S}_8,\mathcal{S}_9$。结果得到 $\mathcal{N}_0'$，其可达图与图2(*b*)所示的可达图同构。还要注意，无法在局部移除 $\mathcal{S}_3,\mathcal{S}_6,\mathcal{S}_7,\mathcal{S}_9$，因为这需要考虑其他顺序子网。

本文将展示，对于当前问题，存在另一种组合式解。为此，我们将考虑用分布式库所$\mathcal{P}_1,\ldots,\mathcal{P}_k$ 覆盖安全 Petri 网的库所 P。每个分布式库所 $\mathcal{P}_i$ 都是一个库所集合，满足的性质仅取决于 $\mathcal{P}_i$与 ${}^\bullet\mathcal{P}_i \cup \mathcal{P}_i^\bullet$中的变迁之间的弧。重要的是，与这些变迁相连的其他弧无关，因此分布式库所的定义完全是局部的。随后我们证明，可以从 $\mathcal{P}_i$中删除一些库所，使 $\mathcal{P}_1',\ldots,\mathcal{P}_k'$成为一个行为等价的安全Petri网的库所。而且，至关重要的是，从 $\mathcal{P}_i$ 到$\mathcal{P}_i'$的化简既是静态的，也是局部的。

例如，再次考虑图2（a）中所示的 $\mathcal{N}_0$。其库所可划分为两个分布式库所： $\mathcal{P}_1=\{p_1,p_2\}$和 $\mathcal{P}_2=\{p3,\ldots,p_{10}\}$。在这种情况下， $\mathcal{P}_1$ 无法化简，因此

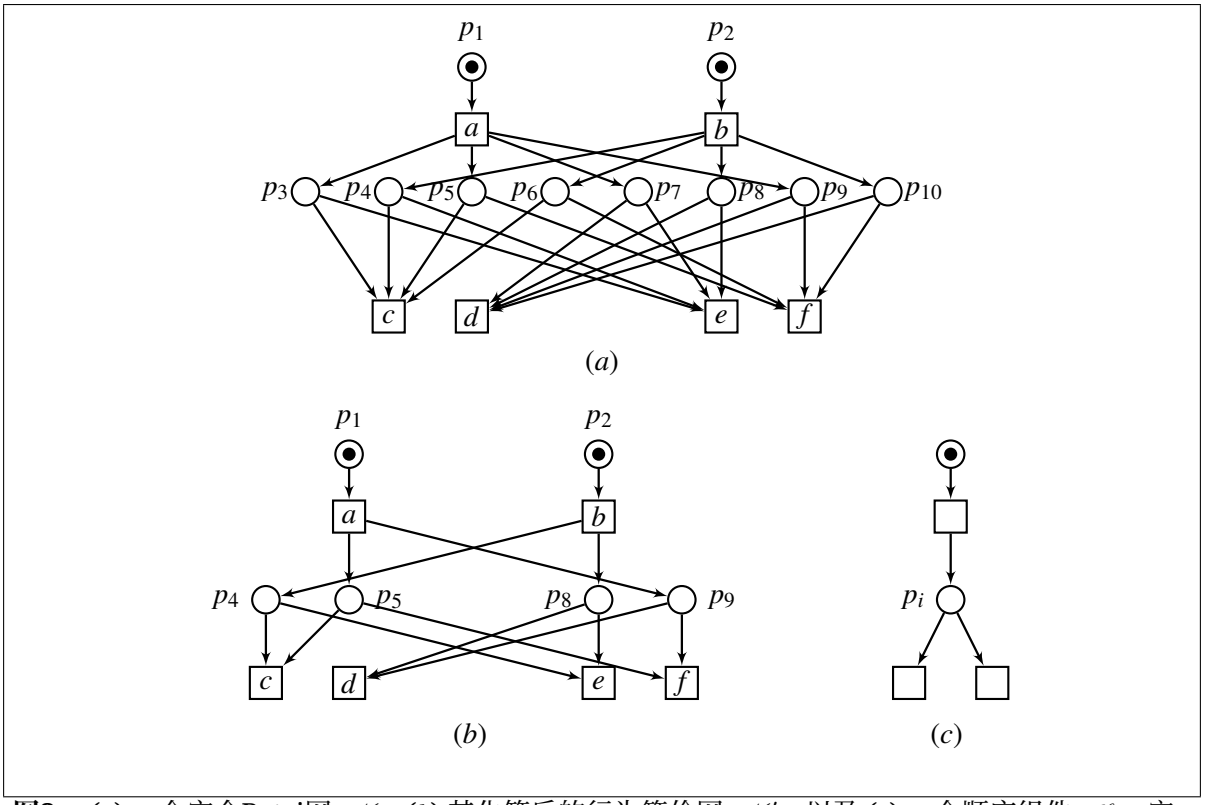

图2： (*a*) 一个安全Petri网 $\mathcal{N}_0$，(*b*) 其化简后的行为等价网 $\mathcal{N}_0'$，以及 (*c*) 一个顺序组件 $\mathcal{S}_i$，它属于 $\mathcal{N}_0 (i = 3, 4, \ldots\ldots, 10)$。

$\mathcal{P}_1' = \mathcal{P}_1$，但 $\mathcal{P}_2$ 可以化简为 $\mathcal{P}_2' = \{p_4, p_5, p_8, p_9\}$。结果仍是图2（*b*）中所示的网，其行为等价于 $\mathcal{N}_0'$，不过这次化简是在局部完成的。

本文提出了“分布式库所”这一新颖概念，直观而言，它为 Petri网的各个库所提供了一种并发实现（因此，Petri网为分布式库所提供了抽象）。我们还基于分布式库所提出了化简过程，并证明了其正确性。

接下来讨论问题I，我们将考虑由进程表达式导出的箱（安全Petri网），这些表达式支持顺序组合（$E;F$）、选择组合（$E \,\square F$）、并行组合（$E \parallel F$）和迭代组合（$[E * F * G]$）。对于每个这样的表达式E，都有一种标准转换，可生成一个安全Petri网箱（E），称为箱。对于由组合方式导出的箱，分布式库所是“通过构造”产生的。更重要的是，要化简箱（E），不必显式导出箱（E），因为其规模可能随E的大小呈指数增长。相反，我们构造一个无向“连接图”CG_E ——一种特殊的余图——其表示规模至多与E的大小呈线性关系，并直接从CG_E的边团覆盖中导出$\mathcal{P}_i'$。请注意，余图上的边团覆盖问题属于 NP，且被认为是 NP 完全问题；不过，存在多项式启发式算法，甚至将每条边分别用一个团覆盖的简单方法，所得解的规模也至多为二次，这远优于笛卡尔积构造所产生的指数规模解。

本文结构。 本文分为三部分。在第2节的预备内容之后，第一部分（第3节）介绍分布式库所。特别是，定理 3.8 对分布式库所作为一种保持行为的安全Petri网化简手段的适用性给出了形式化论证。第二部分（第4节和第5节）首先确定了两种组合分布式库所的运算，并证明了分布式库所的一些基本性质。随后，本文展示如何在将盒表达式标准转换为箱时使用这两种运算。第三部分（第6节和第7节）介绍连接图的基本事实，并说明如何利用它们，通过优化翻译将盒表达式转换为行为等价的箱。

本文是会议论文 [19] 的修订扩展版。

2. 基本概念和记号

网。 网 是一个元组 (P, T, Fl)，其中 P 是有限的库所集合，T 是与之不相交的有限变迁集合，而 $Fl \subseteq (T \times P) \cup (P \times T)$ 是流关系。

对于每个 $x \in P \cup T$，$^\bullet x = \{y \mid (y, x) \in F\}$，$x^\bullet = \{y \mid (x,y) \in F\}$，且 $^\bullet x^\bullet = {}^\bullet x \cup x^\bullet$。这些记号可扩展至集合 $X \subseteq P \cup T$，如下所示：$^\bullet X = \bigcup\{^\bullet x \mid x \in X\}$，$X^\bullet = \bigcup\{x^\bullet \mid x \in X\}$，以及 $^\bullet X^\bullet = {}^\bullet X \cup X^\bullet$。

我们假设每个库所 p 至少有一个输入变迁，$(^\bullet p \neq \varnothing)$，或至少有一个输出变迁，$(p^\bullet \neq \varnothing)$。此外，每个变迁 t 至少有一个前置库所，$(^\bullet t \neq \varnothing)$。

对于每个 $t \in T$，$^\blacktriangleright t = {}^\bullet t \setminus t^\bullet$ 且 $t^\blacktriangleright = t^\bullet \setminus {}^\bullet t$。直观地说，变迁 t *effectively removes* 令牌从 $^\blacktriangleright t$ 中的库所移除，并*effectively inserts* 令牌到$t^\blacktriangleright$中的库所。

变迁序列。 设 σ 和 σ' 为有限变迁序列，且 $V \subseteq T$。那么：(i) $\mathrm{pref}(\sigma)$ 是 σ 的所有前缀构成的集合；(ii) $\sigma|_V$ 是从 σ 中删除所有不属于V的变迁后得到的；(iii) $\sigma \circ \sigma'$ 是 σ 和 σ' 的连接。空序列记作 λ，并假设对于有限序列集合 Σ，$\varnothing^* = \{\lambda\}$ 和 $\Sigma \circ \varnothing = \varnothing$。

标记网。 一个标记网 是一个元组 $\mathcal{N} = (P, T, Fl, M_{init})$，其中 (P, T, Fl) 是一个网，且$M_{init} \subseteq P$ 是初始标识。一般而言，任意库所集合都是一个标识。我们采用将网表示为有向图的标准方式，例如，变迁画成小方块，库所画成圆圈。

若变迁$t \in T$ 在标识M 处满足 $^\bullet t \subseteq M$，则该变迁可发射 。这样的变迁可以发射 ，从而到达标识$M' = (M \setminus {}^\bullet t) \cup t^\bullet = (M \setminus {}^\blacktriangleright t) \cup t^\blacktriangleright$。我们分别将其记为$t \in$可发射$_{\mathcal{N}}(M)$和$M[t\rangle_{\mathcal{N}} M'$。

由 $\mathcal{N}$构成的发射序列 是一个变迁序列 $\sigma = t_1 \ldots t_k (k \geq 0)$，且存在标识$M0, \ldots, M_k$满足$M_{init} = M_0$以及$M_{i-1}[t_i\rangle_{\mathcal{N}} M_i$，对每个 $1 \leq i \leq k$均成立。于是，每个标识M_i在 $\mathcal{N}$中都是可达的。所有发射序列和所有可达标识集合分别记为fseq($\mathcal{N}$)和reach($\mathcal{N}$)。

若存在某些标识 $M1$，……，M_{k-1} 以及变迁 $t1$，……，t_k，使得 $M[t_1\rangle_{\mathcal{N}} M_1[t_2\rangle_{\mathcal{N}} M_2 \ldots M_{k-1}[t_k\rangle_{\mathcal{N}} M'$，则标识 M'可由标识 M经由可达 得到。我们将此记作 $M[t_1 \ldots t_k\rangle_{\mathcal{N}} M'$。

可达图。 </style}三元组 $\mathbf{RG}_{\mathcal{N}} = (\text{reach}(\mathcal{N}),\ A,\ M_{init})$，其中 $\text{reach}(\mathcal{N})$ 是节点，$A = \{(M,\ t,\ M') \mid M \in \text{reach}(\mathcal{N}) \wedge M[t\rangle_{\mathcal{N}} M'\}$ 是有向弧，M_{init} 是初始节点，它就是 $\mathcal{N}$ 的可达图 </style}。

如果存在一个双射 $\imath : \text{reach}(\mathcal{N}) \to \text{reach}(\mathcal{N}')$，使得 $\imath(M_{init}) = M'_{init}$，且对于所有 M、$M' \in \text{reach}(\mathcal{N})$，均有 $(M,t,M') \in A \iff (\imath(M),t,\imath(M')) \in A'$，则两个可达图 $RG_{\mathcal{N}} = (\text{reach}(\mathcal{N}),A,M_{init})$ 和 $RG_{\mathcal{N}'} = (\text{reach}(\mathcal{N}'),A',M'_{init})$ 同构。我们将其记为 $RG_{\mathcal{N}} \cong RG_{\mathcal{N}'}$。

安全网。 若一个标记网 $\mathcal{N}$ 满足：对于每个标识 $M \in \text{reach}(\mathcal{N})$ 以及每个 $t \in$ 可从 $M_{\mathcal{N}}$ 发射的变迁，安全 当且仅当 $t^\bullet \cap (M \setminus {}^\bullet t = \varnothing$（或者，另一种说法是，$t^\blacktriangleright \cap M = \varnothing$）。

其他记号。 设 Q 是一个标记网的非空库所集合。定义三个互不相交的变迁集合：$ins_Q = {}^\bullet Q \setminus Q^\bullet$、$rem_Q = Q^\bullet \setminus {}^\bullet Q$，以及 $read_Q = \{t \in {}^\bullet Q^\bullet \mid {}^\bullet t \cap Q = t^\bullet \cap Q\}$。因此，$ins_Q$ 包含所有向 Q 中添加令牌且不从中移除令牌的变迁，rem_Q 包含所有从 Q 中移除令牌且不向其中添加令牌的变迁，而 $read_Q$ 包含仅“检查”Q 中某些库所是否存在令牌的变迁。此外，我们记作：

$$\text{使能}_Q = \{(t,u) \in T \times T \mid Q \cap t^\blacktriangleright \cap {}^\bullet u \neq \varnothing\} \text{ 且禁用}_Q = \{(t,u) \in T \times T \mid Q \cap {}^\blacktriangleright t \cap {}^\bullet u \neq \varnothing\}.$$

示例 2.1。 对于图2(*a*)中的 $\mathcal{N}_0$，我们有：

$$\begin{aligned}
ins_{\{p_3,p_6\}} &= \{a,b\} \\
rem_{\{p_3,p_6\}} &= \{c,e,f\} \\
read_{\{p_3,p_6\}} &= \varnothing \\
enables_{\{p_3,p_4,\ldots,p_{10}\}} &= \{a,b\} \times \{c,d,e,f\} \\
disables_{\{p_3,p_4,\ldots,p_{10}\}} &= (\{c,d\} \times \{e,f\}) \cup (\{e,f\} \times \{c,d\}) \\
enables_{\{p_6,p_7,p_{10}\}} &= \{(b,c),(b,f),(b,d),(a,d),(a,e)\} \\
disables_{\{p_6,p_7,p_{10}\}} &= \{(d,f),(f,d),(d,e),(e,d),(c,f),(f,c)\}.
\end{aligned}$$

此外，在图2(*a*)和图2(*b*)中，我们都有使能$_{\{p_4,p_5,p_8,p_9\}}$ =使能$_{\{p_3,p_4,\ldots,p_{10}\}}$ 且禁用$_{\{p_4,p_5,p_8,p_9\}}$ =禁用$_{\{p_3,p_4,\ldots,p_{10}\}}$。 ◇

标记网规模。 初始标识中的令牌总数不能超过库所数，因此可以将 $\mathcal{N}$ 的规模定义为库所、变迁和弧的总数。实际上，$\mathcal{N}$ 的规模主要由其弧决定。将进程表达式转换为网时，变迁集合是给定的，目标是使用尽可能少的库所和弧。

3. 分布式库所

我们现在引入一个概念，旨在捕捉“分布式库所”（位置集合）这一简单的结构性概念；随着执行的推进，其状态（标识）会在空与满之间单调变化

。例如， $\mathscr{P}=\{p_3,p4,\ldots\ldots,p_{10}\}$是图2($a$)中网的一个分布式库所。初始时， $\mathscr{P}$ 为空，触发a 和 b 会向其中的库所插入令牌。请注意，在填满过程中，没有变迁能从 $\mathscr{P}$移除令牌。随后，例如，变迁c 和d 可以触发并移除 $\mathscr{P}$ 中的所有令牌；在清空过程中，没有变迁能向 $\mathscr{P}$插入令牌。

我们首先明确，对库所集合Q的标识进行单调改变是什么意思。请注意，一般而言，单次发射一个变迁不足以用令牌填满空的Q。相反，我们可能需要通过一系列变迁来达到预期效果。同样，一般而言，单次发射一个变迁也不足以清空装满令牌的Q。

定义 3.1（输入序列和输出序列）
设Q 为标记网的一个非空库所集合。

- An Q的 输入序列是 $\sigma=t_1\ldots t_k\in(ins_Q\cup read_Q)^+$满足 $t_1\in ins_Q$以及

$$Q\cap{}^\bullet t_i\subseteq\{t_1,\ldots,t_{i-1}\}^\bullet \quad\text{and}\quad Q\cap t_i^\blacktriangleright\cap\{t_1,\ldots,t_{i-1}\}^\bullet=\varnothing \ \ (\text{for every } 1<i\le k)\,.$$

  我们将其记为 $\sigma\in inseq_Q$。此外， σ若完整，则$Q\subseteq\{t1,\ldots,t_k\}^\bullet$，此时记为$\sigma\in cinseq_Q$。

- 输出序列 Q是 $\sigma=t1\ldots tk\in(rem_Q\cup read_Q)^+$满足$t_1\in rem_Q$以及

$$Q\cap{}^\bullet t_i\subseteq Q\setminus({}^\blacktriangleright t_1\cup\cdots\cup{}^\blacktriangleright t_{i-1})\ \ (\text{for every } 1<i\le k)\,.$$

  我们将其记为 $\sigma\in$输出序列$_Q$。此外，若 σ是完整的，则 $Q\subseteq{}^\blacktriangleright t_1\cup\cdots\cup{}^\blacktriangleright t_k$，此时我们记为 $\sigma\in$完整输出序列$_Q$。◇

直观地说，输入序列会以单调且安全的方式将令牌插入空的 Q中（这是因为假设$Q\cap t_i^\blacktriangleright\cap\{t1,\ldots,t_{i-1}\}^\bullet=\varnothing$），而输出序列则以单调的方式从完全标记的 Q中移除令牌。

注意，如果 输入序列$_Q=\varnothing$ 则 输入序列$_Q=$ 完整输入序列$_Q=\varnothing$，且如果 移除$_Q=\varnothing$ 则 输出序列$_Q=$ 完整输出序列$_Q=\varnothing$.

定义 3.2。（分布式库所）
标记网中的一个分布式库所是一个非空库所集合 $\mathscr{P}$，满足：

1. ${}^\bullet\mathscr{P}^\bullet=$输入$_\mathscr{P}\cup$移除$_\mathscr{P}\cup$读取$_\mathscr{P}$。
2. 输入$_\mathscr{P}\times$移除$_\mathscr{P}\subseteq$使能$_\mathscr{P}$。
3. 输入序列$_\mathscr{P}\subseteq$前缀(输入序列$_\mathscr{P}$) 以及输出序列$_\mathscr{P}\subseteq$前缀(输出序列$_\mathscr{P}$)。

此外， $\mathscr{P}$是纯的，当且仅当 ${}^\bullet\mathscr{P}\cap\mathscr{P}^\bullet=\varnothing$（或者，等价地，读取$_\mathscr{P}=\varnothing$）。 ◇

请注意，上述定义是局部的，因为它仅取决于与构成该分布式库所的库所相邻的变迁和弧。

根据定义 3.2(1)，与分布式库所 $\mathscr{P}$ 相邻的变迁有三种：(i) $ins_\mathscr{P}$ 实际上向 $\mathscr{P}$中插入令牌的变迁；(ii) $rem_\mathscr{P}$ 实际上从 $\mathscr{P}$ 中移除令牌的变迁；以及 (iii) $read_\mathscr{P}$ 仅检查 $\mathscr{P}$ 中令牌是否存在的变迁。

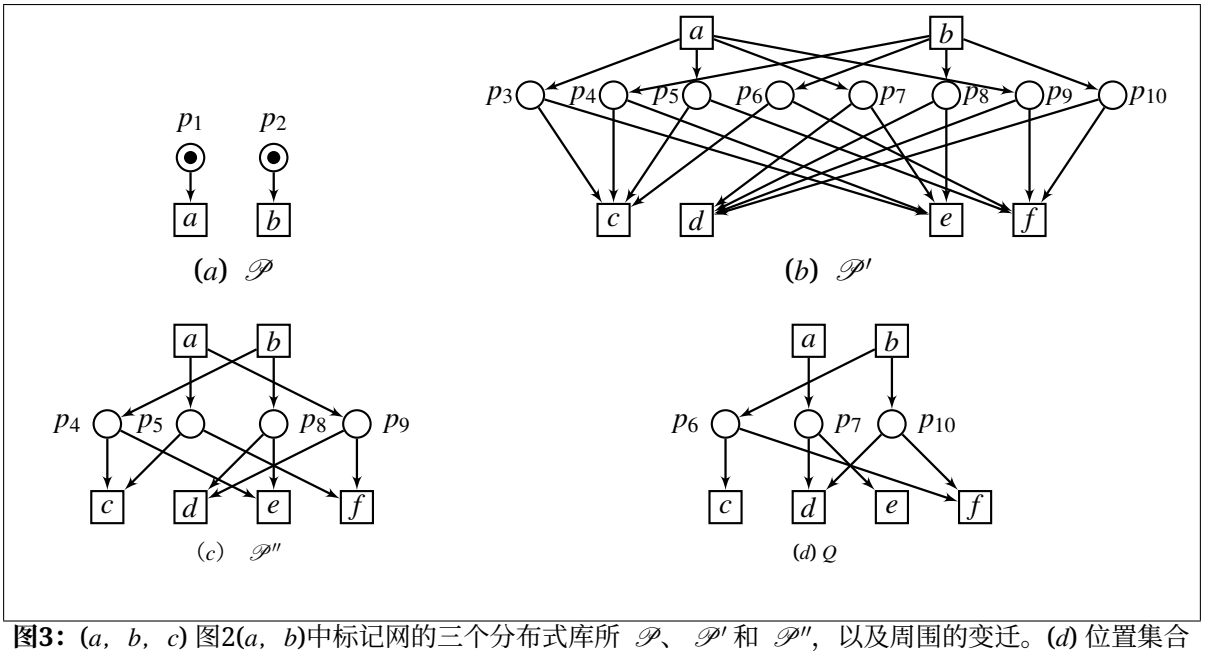

图3： (*a*，*b*，*c*) 图2(*a*，*b*)中标记网的三个分布式库所 $\mathscr{P}$、$\mathscr{P}'$ 和 $\mathscr{P}''$，以及周围的变迁。(*d*) 位置集合 $Q = \{p_6, p_7, p_{10}\}$ 及其周围的变迁；对于图 2(*a*) 中的标记网而言，它不是一个分布式库所 $\mathscr{N}_0$。

当分布式库所 Q 是安全标记网 $\mathscr{N}$ 的一部分时，以下两个重要性质便会显现：(i) 如果 Q 初始为空，那么在不完整的（投影）输入序列 σ之后，无法“发射”变迁$u \in rem_{\mathscr{P}}$；(ii) 如果 Q 初始为满，那么在不完整的（投影）输出序列 σ之后，无法“发射”变迁 $u \in ins_{\mathscr{P}}$。更准确地说，对于 (ii)，假设 $\omega \circ u$是 $\mathscr{N}$ 的一个发射序列，它导向标识 M，$u \in ins_{\mathscr{P}}$，并且 $\sigma = \omega|_{\bullet Q^\bullet} = t_1 \ldots t_m$ 是 Q 的一个不完整（投影）输出序列。显然，$M \cap Q = Q \setminus ({}^\blacktriangleright t_1 \cup \cdots \cup {}^\blacktriangleright t_m)$。然后，根据定义 3.2(3)，存在 $t \in rem_{\mathscr{P}}$使得 $\sigma \circ t$ 是一个输出序列，因此，根据定义 3.1，有 $Q \cap {}^\bullet t \subseteq Q \cap M$ (*)。另一方面，根据 $(u,t) \in ins_{\mathscr{P}} \times rem_{\mathscr{P}}$和定义 3.2(2)，我们有 $(u,t) \in enables Q$，因此存在 $q \in Q \cap u^\blacktriangleright \cap {}^\bullet t$。于是，根据 (*)，有 $q \in M$，因此 u 在 ωas $\mathscr{N}$之后不能被发射，因为其是安全的。

例3.3。图 3(a, b) 描绘了安全标记网 $\mathscr{N}_0$ 图 2(a) 中所示的两个不相交的分布式库所及其周围的变迁：$\mathscr{P}' = \{p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$ 以及$\mathscr{P} = \{p_1, p_2\}$。此外，它们的完整输入序列和输出序列如下：$inseq_{\mathscr{P}} = \varnothing$、$inseq_{\mathscr{P}'} = \{ab、ba\}$、$outseq_{\mathscr{P}} = \{ab、ba\}$，以及 $outseq_{\mathscr{P}'} = \{cd、dc、e\,f、f\,e\}$。

然后我们观察到，$\mathscr{P}'' = \{p_4, p_5, p_8, p_9\}$ 在图2（c）中是 $\mathscr{N}_0$的一个分布式库所，其输入序列和输出序列与 $\mathscr{P}'$相同。我们还观察到，位置集合$Q = \{p_6, p_7, p_{10}\}$（如图2（d）所示）不是分布式库所，因为（a,c）$\notin$ 使能$_Q$。$\diamond$

对于纯分布式库所，输入序列（输出序列）的所有排列都是输入序列（输出序列）。这一观察为分布式库所提供了静态刻画。

这类结果有助于在带标记的网中识别分布式库所，然后再应用支撑所提出网约简的定理 3.8。

命题 3.4. 网的非空库所集合 $\mathscr{P}$ 是分布式库所，当且仅当满足：

1. $^\bullet\mathscr{P}^\bullet =$ 输入$_{\mathscr{P}} \cup$ 移除$_{\mathscr{P}} \cup$ 读取$_{\mathscr{P}}$。 2. 输入$_{\mathscr{P}} \times$ 移除$_{\mathscr{P}} \subseteq$ 使能$_{\mathscr{P}}$。 3. 若输入$_{\mathscr{P}} \neq \varnothing$ 且 $U \subseteq$ 输入$_{\mathscr{P}}$ 是一个极大非空集合，使得对于所有 $t \neq u \in U$，$t^\bullet \cap u^\bullet \cap \mathscr{P} = \varnothing$，则 $\mathscr{P} \subseteq U^\bullet$。 4. 若移除$_{\mathscr{P}} \neq \varnothing$ 且 $U \subseteq$ 移除$_{\mathscr{P}}$ 是一个极大非空集合，使得对于所有 $t \neq u \in U$，$^\bullet t \cap {}^\bullet u \cap \mathscr{P} = \varnothing$，则 $\mathscr{P} \subseteq {}^\bullet U$。

证明：
首先注意到，部分 (1) 和 (2) 与定义 3.2(1,2) 相同。

($\Longrightarrow$) 为证明第 (3) 部分，令 $U = \{t1, \ldots, t_k\}$（证明第 (4) 部分时，采用类似的方法）。于是，$\sigma = t_1 \ldots t_k$ 是 $\mathscr{P}$ 的一个输入序列，且无法通过属于 $ins_{\mathscr{Q}}$ 的变迁将其扩展为更长的输入序列。若 $\mathscr{P} \not\subseteq U^\bullet$，则根据定义 3.2(3)，存在 $t \in ins_{\mathscr{Q}}$，使得$\sigma \circ t \in \mathrm{pref}(cinseq_{\mathscr{Q}})$，矛盾。

($\Longleftarrow$) 为证明定义 3.2(3)，假设 $\sigma \in \mathrm{inseq}_{\mathscr{P}}$（若 $\sigma \in \mathrm{outseq}_{\mathscr{P}}$，则采用类似的方法）。于是，根据定义 3.1，$\sigma|_{ins_{\mathscr{P}}} = t_1 \ldots t_k \in \mathrm{inseq}_{\mathscr{P}}$。因此，$U = \{t1, \ldots, t_k\} \subseteq \mathrm{ins}_{\mathscr{P}}$ 满足：对于所有 $1 \leq i < j \leq k$，均有 $t_i \neq t\,j$ 且 $t_i^\bullet \cap t_j^\bullet = \varnothing$。若 U 是极大的，则有 $U^\bullet = \mathscr{P}$，因此 $\sigma \in \mathrm{cinseq}_{\mathscr{P}}$。否则，存在 $t \in \mathrm{ins}_{\mathscr{P}}$，使得 $t^\bullet \cap U^\bullet \cap \mathscr{P} = \varnothing$。于是 $\sigma|_{ins_{\mathscr{P}}} \circ t \in \mathrm{inseq}_{\mathscr{P}}$。接着可以重复同样的论证（可能重复多次，但次数有限），直到得到一个完整输入序列 $\sigma|_{ins_{\mathscr{P}}} \circ \omega \in \mathrm{cinseq}_{\mathscr{P}}$。然后，根据定义 3.1，$\sigma \circ \omega \in \mathrm{cinseq}_{\mathscr{P}}$。因此，定义 3.2(3) 成立。□

我们还有以下直接推论，它揭示了分布式库所与普通库所之间的关系。

命题 3.5. 对于标记网的每个库所 p，$\mathscr{S}_p = \{p\}$ 都是一个分布式库所，使得 $ins_{\mathscr{S}_p} = {}^\blacktriangleright p$、$rem_{\mathscr{S}_p} = p$ 和 $read^{\mathscr{S}_p} = p^\bullet \cap {}^\bullet p$ 成立，并且：

$$inseq_{\mathscr{S}_p} \;=\; cinseq_{\mathscr{S}_p} \;=\; \begin{cases} ins_{\mathscr{S}_p} \circ read^*_{\mathscr{S}_p} & if \quad ins_{\mathscr{S}_p} \neq \varnothing \\ \varnothing & if \quad ins_{\mathscr{S}_p} = \varnothing \end{cases} \tag{1}$$

$$outseq_{\mathscr{S}_p} \;=\; coutseq_{\mathscr{S}_p} \;=\; \begin{cases} rem_{\mathscr{S}_p} & if \quad rem_{\mathscr{S}_p} \neq \varnothing \\ \varnothing & if \quad rem_{\mathscr{S}_p} = \varnothing\,. \end{cases}$$

此外，$\mathscr{S}_p$ 是纯的，当且仅当 $p^\bullet \cap {}^\bullet p = \varnothing$。

证明：
这直接由定义得出。 □

因此，标记网 $\mathcal{N}$的库所可由平凡的单元素分布式库所 $\mathcal{S}_p$覆盖。由此立即推论，分布式库所覆盖并不能保证 $\mathcal{N}$的安全性。相反，若存在顺序组件覆盖，则意味着 $\mathcal{N}$是安全的。

直观地说，分布式库所是局部状态的一种实现，该状态以循环且单调的方式被填充和清空。接下来的结果对此进行了明确说明：安全标记网中每个分布式库所的行为都可以理解为输入序列和输出序列交替执行的过程，前提是将初始标识设置为“空”或“满”（这种行为体现在下一个结果表述中方程式 (2) 的前两行）。

定理 3.6。设 $\mathscr{P}$ 是安全标记网 $\mathcal{N}=(P,\ T,\ Fl,\ M_{init})$中的一个分布式库所。那么，对于每个 $\sigma\in\mathrm{fseq}(\mathcal{N})$，都有：

$$\sigma|_{\bullet\mathscr{P}\bullet}\in\begin{cases}\mathrm{pref}((cinseq_{\mathscr{P}}\circ coutseq_{\mathscr{P}})^{+}) & if\quad ins_{\mathscr{P}}\neq\varnothing\neq rem_{\mathscr{P}}\wedge\mathscr{P}\cap M_{init}=\varnothing\\ \mathrm{pref}(Read_{\mathscr{P}}^{*}\circ(coutseq_{\mathscr{P}}\circ cinseq_{\mathscr{P}})^{+}) & if\quad ins_{\mathscr{P}}\neq\varnothing\neq rem_{\mathscr{P}}\wedge\mathscr{P}\subseteq M_{init}\\ \mathrm{pref}(cinseq_{\mathscr{P}}) & if\quad ins_{\mathscr{P}}\neq\varnothing=rem_{\mathscr{P}}\wedge\mathscr{P}\cap M_{init}=\varnothing\\ \mathrm{pref}(Read_{\mathscr{P}}^{*}\circ coutseq_{\mathscr{P}}) & if\quad ins_{\mathscr{P}}=\varnothing\neq rem_{\mathscr{P}}\wedge\mathscr{P}\subseteq M_{init}\\ \{\lambda\} & if\quad ins_{\mathscr{P}}=\varnothing\qquad\wedge\mathscr{P}\cap M_{init}=\varnothing\\ Read_{\mathscr{P}}^{*} & if\quad rem_{\mathscr{P}}=\varnothing\qquad\wedge\mathscr{P}\subseteq M_{init}\ .\end{cases}\tag{2}$$

证明:
令 $M_{init}[\sigma'\rangle_{\mathcal{N}}M$。证明对 σ 的长度进行归纳法。

在基础步骤中，$\sigma=\lambda$ 满足方程式 (2)。在归纳步骤中，取 $\sigma\circ t\in\mathrm{fseq}(\mathcal{N})$，其中 σ 满足方程式 (2)，并在假设 $\mathscr{P}\cap M_{init}=\varnothing$ 的情况下考虑几种情形。（当 $\mathscr{P}\subseteq M_{init}$ 时，证明类似。）

情形 *1*：输入$_{\mathscr{P}}\neq\varnothing\neq$剩余$_{\mathscr{P}}$和 $\sigma|_{\bullet\mathscr{P}\bullet}\in(cinseq_{\mathscr{P}}\circ coutseq\mathscr{P})^{*}\circ$输入序列$_{\mathscr{P}}$。那么 σ可以表示为 $\sigma=\sigma'\circ\sigma''$，其中：

$$\sigma'|_{\bullet\mathscr{P}\bullet}\in(cinseq_{\mathscr{P}}\circ coutseq_{\mathscr{P}})^{*}\quad\text{and}\quad\sigma''|_{\bullet\mathscr{P}\bullet}\in inseq_{\mathscr{P}}\ .$$

设 $M_{init}[\sigma'\rangle_{\mathcal{N}}M'[\sigma''\rangle_{\mathcal{N}}M$。注意，$\mathscr{P}\cap M'=\varnothing$。如果 $t\notin{}^{\bullet}\mathscr{P}^{\bullet}$，那么 $(\sigma\circ t)|_{\bullet\mathscr{P}\bullet}=\sigma|_{\bullet\mathscr{P}\bullet}$，证明完毕。否则，$(\sigma''\circ t)|_{\bullet\mathscr{P}\bullet}=\sigma''|_{\bullet\mathscr{P}\bullet}\circ t$，根据定义 3.2(1)，我们可以考虑三种情形。

情形 *1.a*：$t\in$读取$_{\mathscr{P}}$。那么，显然，$\sigma''|_{\bullet\mathscr{P}\bullet}\circ t\in$输入序列$_{\mathscr{P}}$。

情形 *1.b*：$t\in$输入$_{\mathscr{P}}$。于是，由于 $\mathcal{N}$ 是安全的且 $\sigma''|_{rem_{\mathscr{P}}}=\lambda$，我们有 $\sigma''|_{\bullet\mathscr{P}\bullet}\circ t\in$输入序列$_{\mathscr{P}}$。

情形 1.c：$t\in\mathrm{rem}_{\mathscr{P}}$。于是，由于 $\mathscr{P}\cap M'=\varnothing$，$\sigma''|_{ins_{\mathscr{P}}}\neq\lambda$。如果 $\mathscr{P}\subseteq M$，那么我们有 $\sigma''|_{\bullet\mathscr{P}\bullet}\in\mathrm{cinseq}_{\mathscr{P}}$，证明完毕。如果 $\mathscr{P}\not\subseteq M$，那么 $\sigma''|_{\bullet\mathscr{P}\bullet}\notin\mathrm{cinseq}_{\mathscr{P}}$，因此根据定义 3.2(3)，存在 $u\in$输入$_{\mathscr{P}}$，使得 $\sigma''|_{\bullet\mathscr{P}\bullet}\circ u\in$输入序列$_{\mathscr{P}}$。因此，根据 $\mathscr{P}\cap M'=\varnothing$以及输入序列的定义，$\mathscr{P}\cap u^{\blacktriangleright}\cap M=\varnothing$。另一方面，$\mathscr{P}\cap{}^{\bullet}t\subseteq M$。因此 $u^{\blacktriangleright}\cap{}^{\bullet}t\cap\mathscr{P}=\varnothing$，这与定义 3.2(2) 以及 $(u,t)\in\mathrm{enables}_{\mathscr{P}}$矛盾。

情形 *2*：输入$_{\mathscr{P}}\neq\varnothing\neq$剩余$_{\mathscr{P}}$和 $\sigma|_{\bullet\mathscr{P}\bullet}\in$(输入序列$_{\mathscr{P}}\circ$输出序列$_{\mathscr{P}}$)$^{*}\circ$输入序列$_{\mathscr{P}}\circ$输出序列$_{\mathscr{P}}$。那么 σ可以表示为（不一定是唯一的）$\sigma=\sigma'\circ\sigma''$，其中：

$$\sigma'|_{\bullet\mathscr{P}\bullet}\in(cinseq_{\mathscr{P}}\circ coutseq_{\mathscr{P}})^{*}\circ cinseq_{\mathscr{P}}\quad\text{and}\quad\sigma''|_{\bullet\mathscr{P}\bullet}\in outseq_{\mathscr{P}}\ .$$

设 $M_{init}[\sigma'\rangle_{\mathcal{N}} M'[\sigma''\rangle_{\mathcal{N}} M$。注意，$\mathcal{P} \subseteq M'$。如果 $t \notin {}^\bullet\mathcal{P}^\bullet$，那么 $(\sigma'' \circ t)|_{\bullet\mathcal{P}\bullet} = \sigma''|_{\bullet\mathcal{P}\bullet}$，问题得证。否则，$(\sigma'' \circ t)|_{\bullet\mathcal{P}\bullet} = \sigma''|_{\bullet\mathcal{P}\bullet} \circ t$，并且根据定义 3.2(1)，我们可以考虑三种情形。

情形 *2.a*：$t \in$读取$_{\mathcal{P}}$。那么，显然，$\sigma''|_{\bullet\mathcal{P}\bullet} \circ t \in$输出序列$_{\mathcal{P}}$。

情形 2.b：$t \in$插入$_{\mathcal{P}}$。那么，由于 $\mathcal{P} \subseteq M'$且 $\mathcal{N}$是安全的，$\sigma''|_{rem_{\mathcal{P}}} \neq \lambda$。如果 $\mathcal{P} \cap M = \varnothing$，那么我们有$\sigma''|_{\bullet\mathcal{P}\bullet} \in$输出序列$_{\mathcal{P}}$，从而得证。如果 $\mathcal{P} \not\subseteq M$，那么 $\sigma''|_{\bullet\mathcal{P}\bullet} \notin$c输出序列$_{\mathcal{P}}$，因此，根据定义 3.2(3)，存在 $u \in$移除$_{\mathcal{P}}$，使得 $\sigma''|_{\bullet\mathcal{P}\bullet} \circ u \in$输出序列$_{\mathcal{P}}$。因此，根据 $\mathcal{P} \subseteq M'$以及输出序列的定义，$\mathcal{P} \cap {}^\bullet u \subseteq M'$。另一方面，由于 $\mathcal{N}$是安全的，$\mathcal{P} \cap t^\bullet \cap M = \varnothing$。于是 $t^\bullet \cap {}^\bullet u \cap \mathcal{P} = \varnothing$，这与定义 3.2(2) 以及 $(t, u) \in$启用$_{\mathcal{P}}$相矛盾。

情形 *2.c*：$t \in rem_{\mathcal{P}}$。然后，作为 $\sigma''|_{ins_{\mathcal{P}}} = \lambda$，$\sigma''|_{\bullet\mathcal{P}\bullet} \circ t \in$输出序列$_{\mathcal{P}}$。

情形 *3*：$ins_{\mathcal{P}} \neq \varnothing \neq rem_{\mathcal{P}}$且 $\sigma|_{\bullet\mathcal{P}\bullet} \in (cinseq_{\mathcal{P}} \circ coutseq_{\mathcal{P}})^*$。然后 $\mathcal{P} \cap M = \varnothing$。接着二者必居其一：$t \in ins_{\mathcal{P}}$或$t \notin {}^\bullet Q^\bullet$，因此 $\sigma \circ t|_{\bullet\mathcal{P}\bullet} \in (cinseq_{\mathcal{P}} \circ coutseq_{\mathcal{P}})^* \circ inseq_{\mathcal{P}}$。

情形 *4*：$ins_{\mathcal{P}} \neq \varnothing = rem_{\mathcal{P}}$且 $\sigma|_{\bullet\mathcal{P}\bullet} \in inseq_{\mathcal{P}}$。然后我们按照情形 1 中的方式继续，但情形 1.c 无法成立。

*C*情形 *5*：$ins_{\mathcal{P}} = \varnothing$。然后对于每个$M \cap Q = \varnothing$，$\in \mathrm{reach}(\mathcal{N})$。因此 $\sigma|_{\bullet\mathcal{P}\bullet} = \lambda$. □

推论3.7。设 $\mathcal{P}$ 是安全标记网 $\mathcal{N} = (P, T, Fl, M_{init})$ 的一个纯分布式库所。则对于每个 $\sigma \in \mathrm{fseq}(\mathcal{N})$，都有：

$$\sigma|_{\bullet\mathcal{P}\bullet} \in \begin{cases} \mathrm{pref}((cinseq_{\mathcal{P}} \circ coutseq_{\mathcal{P}})^+) & if \quad ins_{\mathcal{P}} \neq \varnothing \neq rem_{\mathcal{P}} \wedge \mathcal{P} \cap M_{init} = \varnothing \\ \mathrm{pref}((coutseq_{\mathcal{P}} \circ cinseq_{\mathcal{P}})^+) & if \quad ins_{\mathcal{P}} \neq \varnothing \neq rem_{\mathcal{P}} \wedge \mathcal{P} \subseteq M_{init} \\ \mathrm{pref}(cinseq_{\mathcal{P}}) & if \quad ins_{\mathcal{P}} \neq \varnothing = rem_{\mathcal{P}} \wedge \mathcal{P} \cap M_{init} = \varnothing \\ \mathrm{pref}(coutseq_{\mathcal{P}}) & if \quad ins_{\mathcal{P}} = \varnothing \neq rem_{\mathcal{P}} \wedge \mathcal{P} \subseteq M_{init} \\ \{\lambda\} & if \quad ins_{\mathcal{P}} = \varnothing \quad \wedge \mathcal{P} \cap M_{init} = \varnothing \\ \{\lambda\} & if \quad rem_{\mathcal{P}} = \varnothing \quad \wedge \mathcal{P} \subseteq M_{init}\,. \end{cases} \tag{3}$$

下一个结果从本文所研究的应用角度刻画了分布式库所的一个基本性质，即在由分布式库所覆盖的安全标记网中，可以在局部层面（即每个分布式库所内部）应用化简，而不改变其语义。

定理 3.8。设 $\mathcal{N} = (P, T, Fl, M_{init})$ 是一个安全标记网，并且对于每个库所 $p \in P$，设 $\mathcal{P}_p$ 是 $\mathcal{N}$ 中的一个分布式库所，且包含 p，使得 $\mathcal{P}_p \cap M_{init} = \varnothing$ 或 $\mathcal{P}_p \subseteq M_{init}$。此外，设 $P' \subseteq P$ 是一个库所集合，使得

$$\mathcal{N}' = (P', T, Fl', M'_{init}) = (P', T, Fl|_{T \times P' \cup P' \times T}, M_{init} \cap P')$$

是一个标记网，并且对每个 $p \in P$ 及 $\mathcal{P}'_p = \mathcal{P}_p \cap P'$ 均满足：

$${}^\bullet(\mathcal{P}'_p)^\bullet = {}^\bullet(\mathcal{P}_p)^\bullet \tag{4a}$$

$$enables_{\mathcal{P}'_p} = enables_{\mathcal{P}_p} \tag{4b}$$

$$disables_{\mathcal{P}'_p} = disables_{\mathcal{P}_p} \tag{4c}$$

那么，以下结论成立：

1. fseq($\mathcal{N}'$) = fseq($\mathcal{N}$)。

2. $\mathcal{N}'$ 是一个安全标记网。

3. $RG_{\mathcal{N}} \cong RG_{\mathcal{N}'}$ 满足以下条件：对于每个$p \in P$ 以及 $\mathcal{N}$的每个可达标识M，都存在一个标识M'，它从M 可达，并且满足 $\mathscr{P}_p \cap M' = \varnothing$ 或 $\mathscr{P}_p \subseteq M'$。

证明：

(1) 我们首先注意到，P' 中的每个库所在 $\mathcal{N}'$ 中的输入变迁与在 $\mathcal{N}$中的相同，输出变迁也与在 $\mathcal{N}$中的相同。因此，由于 $\mathcal{N}$是安全的，对于每个 $\sigma \in$fseq($\mathcal{N}$)，我们有如下结论：

$$M_{init}[\sigma\rangle_{\mathcal{N}} M \quad \Longrightarrow \quad M'_{init}[\sigma\rangle_{\mathcal{N}'} M \cap P' \,. \tag{5}$$

由方程式(5)，我们得到 fseq($\mathcal{N}$) ⊆fseq($\mathcal{N}'$)。为了证明反向包含关系也成立，我们对发射序列 $\sigma \in$fseq($\mathcal{N}'$)的长度进行归纳。

在基础步骤中 $\sigma = \lambda$，显然有 $\sigma \in$fseq($\mathcal{N}$)。在归纳步骤中，我们取 $\sigma \circ t \in$fseq($\mathcal{N}'$)，使得 $\sigma \in$ fseq($\mathcal{N}$)，并假设 $\sigma \circ t \notin$ fseq($\mathcal{N}$)。

设 $M_{init}[\sigma\rangle_{\mathcal{N}} M$。于是，根据公式(5)，有 $M'_{init}[\sigma\rangle_{\mathcal{N}'} M \cap P'$。此外，由于 $t \notin$可发射$_{\mathcal{N}}$(M)，存在 $p \in {}^\bullet t$（即 (p,t) $\in Fl$），使得 $p \notin M$。下文中，我们记

$$\mathscr{P} = \mathscr{P}_p \quad \text{and} \quad \mathscr{P}' = \mathscr{P}'_p \quad \text{and} \quad S = \mathscr{P} \setminus M \quad \text{and} \quad R = \mathscr{P} \cap M \,.$$

显然，$p \in S \setminus P'$。此外，根据 $t \in$可发射$_{\mathcal{N}'}$($M \cap P'$) 和公式(4a)，有 ${}^\bullet t \cap R \cap P' \neq \varnothing$。因此，我们可以取 $r \in {}^\bullet t \cap R \cap P'$。

假设 $\mathscr{P} \cap M_{init} = \varnothing$ （当 $\mathscr{P} \subseteq M_{init}$ 时，证明类似）。由定理 3.6，对于 $\mathscr{P}$，我们需要考虑以下情形。

情形 *1*：$ins_{\mathscr{P}} \neq \varnothing \neq rem_{\mathscr{P}}$且 $\sigma|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \in (cinseq_{\mathscr{P}} \circ coutseq_{\mathscr{P}})^* \circ inseq_{\mathscr{P}}$。

那么 σ 可以表示为 $\sigma = \sigma' \circ \sigma''$，其中：

$$\sigma'|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \in (cinseq_{\mathscr{P}} \circ coutseq_{\mathscr{P}})^* \quad \text{and} \quad \sigma''|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \in inseq_{\mathscr{P}} \,.$$

令 $M_{init}[\sigma'\rangle_{\mathcal{N}} M'[\sigma''\rangle_{\mathcal{N}} M$。那么 $\mathscr{P} \cap M' = \varnothing$ 且 $\mathscr{P} \not\subseteq M$（由 $p \notin M$）。因此，$\sigma''|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \notin$cinseq$_{\mathscr{P}}$。于是，根据定义 3.2(3)，存在 $\sigma''' \circ u$，使得 $p \in u^\blacktriangleright$ 且 $\sigma''|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \circ \sigma''' \circ u \in$inseq$_{\mathscr{P}}$。显然，$\mathscr{P} \cap u^\blacktriangleright \subseteq S$。此外，$P' \cap {}^\bullet t \subseteq R$。因此，(u,t) $\notin$使能$_{\mathscr{P}_p}$，于是根据公式（4b），我们得到 (u,t) $\notin$使能$_{\mathscr{P}_p}$，这与 $p \in u^\blacktriangleright \cap {}^\bullet t$ 矛盾。

情形 *2*：$ins_{\mathscr{P}} \neq \varnothing \neq rem_{\mathscr{P}}$以及 $\sigma|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \in (cinseq_{\mathscr{P}} \circ coutseq_{\mathscr{P}})^* \circ cinseq_{\mathscr{P}} \circ outseq_{\mathscr{P}}$。

然后，σ可以表示为 $\sigma = \sigma' \circ \sigma''$，其中：

$$\sigma'|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \in (cinseq_{\mathscr{P}} \circ coutseq_{\mathscr{P}})^* \circ cinseq_{\mathscr{P}} \quad \text{and} \quad \sigma''|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \in outseq_{\mathscr{P}} \,.$$

设 $M_{init}[\sigma'\rangle_{\mathcal{N}} M'[\sigma''\rangle_{\mathcal{N}} M$。则 $\mathscr{P} \subseteq M'$。因此，根据 $p \notin M$，存在 $u \in rem_{\mathscr{P}}$发生于 σ''中，使得 $p \in {}^\blacktriangleright u$。所以（$u$、$t$）$\in$禁用$_{\mathscr{P}}$，从而根据方程式（4c），（$u$、$t$）$\in$禁用$_{\mathscr{P}'}$。但后者意味着$t \notin$可发射$_{\mathcal{N}'}$（$M \cap P'$），从而产生矛盾。

情形 *3*：$ins_{\mathscr{P}} \neq \varnothing \neq rem_{\mathscr{P}}$以及 $\sigma|_{{}^\bullet\mathscr{P}\mathscr{P}^\bullet} \in (cinseq_{\mathscr{P}} \circ coutseq_{\mathscr{P}})^*$。

情形 *4*：输入$_{\mathscr{P}} \neq \varnothing = rem_{\mathscr{P}}$。

情形 *5*：输入$_{\mathscr{P}}=\varnothing\neq rem_{\mathscr{P}}$或 输入$_{\mathscr{P}}=\varnothing=rem_{\mathscr{P}}$。

(2) 由第（1）部分结合方程式（5）以及 $\mathscr{N}$ 的安全性可得。

(3) 由于 fseq($\mathscr{N}$) =fseq($\mathscr{N}'$)，且标记网的可达图是确定性的，因此只需证明：对于所有 $\sigma,\sigma'\in$fseq($\mathscr{N}$) =fseq($\mathscr{N}'$)，发射序列 σ和 σ' 在 $\mathscr{N}$当且仅当 σ和 σ' 在 $\mathscr{N}'$到达相同标识。我们首先注意到，根据方程式（5），从左到右的蕴含确实成立。

现在假设 $\sigma,\sigma'\in$fseq($\mathscr{N}$) =fseq($\mathscr{N}'$) 满足 $M_{init}[\sigma\rangle_{\mathscr{N}}M$且 $M_{init}[\sigma'\rangle_{\mathscr{N}}M'$，并且：

$$M\neq M' \quad \text{and} \quad M\cap P'=M'\cap P' . \tag{6}$$

那么，存在$p\in P\setminus P'$，使得不失一般性，$p\in M\setminus M'$。由 fseq($\mathscr{N}$) =fseq($\mathscr{N}'$) 以及 $M\cap P'=M'\cap P'$，可得

$$\{\sigma''\mid\sigma\circ\sigma''\in\text{fseq}(\mathscr{N})\}=\{\sigma'''\mid\sigma'\circ\sigma'''\in\text{fseq}(\mathscr{N})\}(=\Sigma) . \tag{7}$$

根据方程式(7)以及 $\mathscr{N}$是安全的，Σ 中的序列不包含 $^\bullet p^\bullet$(*)中的任何变迁。由于 $\mathscr{P}'_p\neq\varnothing$，我们可以取$r\in\mathscr{P}'_p$。根据方程式(6)，$r\in M$当且仅当$r\in M'$（因此$r\neq p$）。接下来我们考虑两种情形。

情形 *1*：$r\in M\cap M'$。

那么，根据$p\notin M'$ 和(*)，以及第(3)部分中的假设，存在一个从M'可达的M''，使得$r\notin M''$。因此，根据方程式(7)，存在一个从M 可达的M'''，使得$r\notin M'''$。于是，根据针对 $\mathscr{P}_p$ 的定理3.6以及第(3)部分中的假设，存在一个从M''' 可达的M''''，使得$p\notin M''''$，这与方程式(7)和(*)矛盾。

情形 *2*：$r\notin M\cup M'$。

于是，根据 $p\in M$、(*) 以及第 (3) 部分中的假设，存在从 M 可达的 M''，使得 $r\in M''$。因此，根据方程式 (7)，存在从 M 可达的 M'''，使得 $r\in M'''$。于是，根据定理 3.6 以及第 (3) 部分中的假设，存在从 M''' 可达的 M''''，使得 $p\in M''''$，这与方程式 (7) 和 (*) 矛盾。□

上述结果为在安全标记网中约简分布式库所提供了一个静态条件。此外，如下所示，定理3.8 中的关键假设均不可省略。

在定理 3.8 的表述中，不能去掉“ $\mathscr{N}$是安全的”这一假设。事实上，否则我们可以取任意一个非安全的 $\mathscr{N}$，再加上平凡单例分布式库所，从而错误地得出“ $\mathscr{N}'=\mathscr{N}$是安全的”这一结论。

我们也不能省略公式（4b）。事实上，考虑图 4(a) 中的 $\mathscr{N}_1$ 输入，其中有三个分布式库所：$\mathscr{P}_1=\{q_1,q_2\}$、$\mathscr{P}_2=\{q_3,q_4,q_5,q_6\}$ 和 $\mathscr{P}_3=\{q_7,q_8\}$。于是我们可以取 $P'=\{q_1,q_2,q_3,q_6,q_7,q_8\}$，而所得的缩减网 $\mathscr{N}_1'$输入在图 4(b) 中生成的发射序列比 $\mathscr{N}_1$ 更多（例如 adbc），尽管除公式（4b）以外的假设均得到满足（请注意，在此情形下，$\mathscr{P}_2\cap P'=\{q_3,q_6\}$ 且 (b, d) $\notin$ 使能$_{\{q_3,q_6\}}$）。

最后，不能省略公式（4c）。事实上，考虑图 4(c) 中的 $\mathscr{N}_2$，它包含两个分布式库所：$\mathscr{P}_1=\{r_1,r_2,r_3,r_4\}$ 和 $\mathscr{P}_2=\{r_5,r_6,r_7,r_8\}$。我们可以取 $P'=\{r_1,r_4,r_5,r_6,r_7,r_8\}$，由此得到的网 $\mathscr{N}_2'$（见图 4(d)）生成的发射序列比 $\mathscr{N}_2$ 更多（例如 ad），尽管除公式（4b）以外的假设都满足（注意，在这种情况下 $\mathscr{P}_1\cap P'=\{r_1,r_4\}$，且 (a, d) $\notin$ 禁用$_{\{r_1,r_4\}}$）。

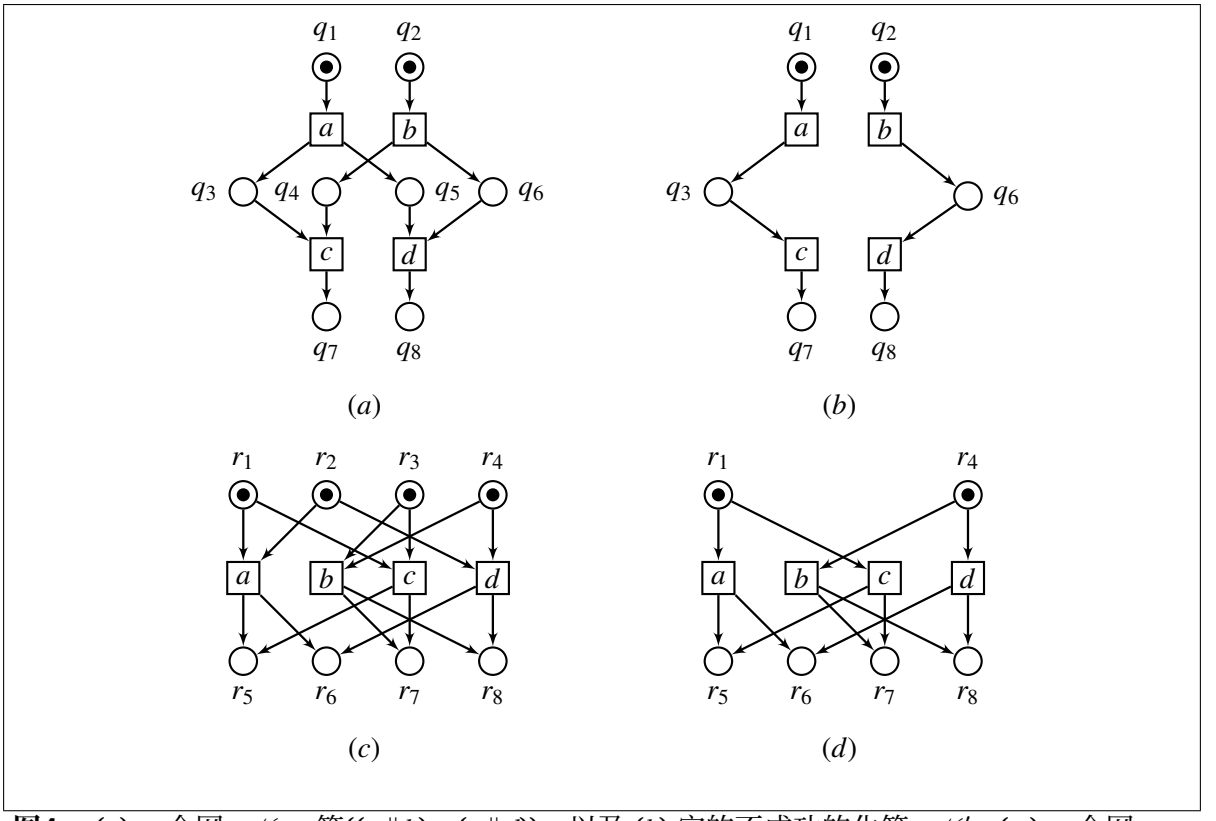

图4： (*a*) 一个网 $\mathscr{N}_1 =$箱$((a \parallel b);\ (c \parallel d))$，以及 (*b*) 它的不成功的化简 $\mathscr{N}_1'$。(*c*) 一个网 $\mathscr{N}_2 =$箱$((a \parallel b) \square (c \parallel d))$，以及 (*d*) 它的不成功的化简 $\mathscr{N}_2'$。

根据命题 3.5，总可以选择由单元素分布式库所构成的覆盖。但由于公式(4a)，这样就完全没有化简。一般而言，合适的分布式库所覆盖和P' 的选择并不唯一，因此有可能进行大幅改进。还值得注意的是，当只给出分布式库所的部分覆盖时，要应用定理 3.8，总可以用平凡单例分布式库所覆盖未覆盖库所。

4. 组合分布式库所

可以用组合方式推导出分布式库所。由于本文其余部分讨论的任何 $\mathscr{N}$ =(P, T, Fl, M_{init}) 都满足以下条件：没有两个不同的库所具有相同的输入变迁和相同的输出变迁（即 ${}^\bullet\mathrm{p} = {}^\bullet\mathrm{r} \wedge \mathrm{p}^\bullet = \mathrm{r}^\bullet$ 蕴含 $\mathrm{p} = \mathrm{r}$），因此库所 p 可以通过集合 ${}^\bullet\mathrm{p}$ 和 $\mathrm{p}^\bullet$ 来识别，所以 p 将记作 π_Z，其中 $\mathrm{Z} = \{\mathrm{t}^{out} \mid (\mathrm{p},\mathrm{t}) \in \mathrm{Fl}\} \cup \{\mathrm{t}^{in} \mid (\mathrm{t},\mathrm{p}) \in \mathrm{Fl}\}$ 是 p 的带标记的输入和输出变迁集合。在示例中，Z 将记作 $\pi_{z_1 \ldots z_k}$，其中 $\mathrm{z}_1 \ldots \mathrm{z}_k$ 是 Z 中元素的枚举。采用这种记法，$\mathscr{N}$ 的变迁和弧

变为隐式，如下所示：

$$\begin{aligned} T &= \{t \mid \exists \pi_Z \in P:\ t^{in} \in Z\} \quad \cup \quad \{t \mid \exists \pi_Z \in P:\ t^{out} \in Z\} \\ Fl &= \{(t,\pi_Z) \in T \times P \mid t^{in} \in Z\} \quad \cup \quad \{(\pi_Z,t) \in P \times T \mid t^{out} \in Z\}. \end{aligned} \tag{8}$$

因此，$\mathcal{N}$的结构可以由其库所集合表示。对于每个变迁 t，我们还记 $\mathcal{I}_t = \mathcal{S}_{\pi_{t^{in}}} = \{\pi_{t^{in}}\}$和 $\mathcal{O}_t = \mathcal{S}_{\pi_{t^{out}}} = \{\pi_{t^{out}}\}$。根据命题 3.5，$\mathcal{I}_t$和 $\mathcal{O}_t$是单元素分布式库所，使得 $ins^{\mathcal{I}_t} = rem_{\mathcal{O}_t} = \{t\}$ 且 $ins^{\mathcal{O}_t} = rem^{\mathcal{I}_t} = read^{\mathcal{I}_t} = read_{\mathcal{O}_t} = \varnothing$。我们还可以立即得到纯分布式库所的如下刻画。

命题 4.1. 分布式库所 $\mathcal{P}$ 是纯的，当且仅当对于每个变迁 t，有 $\{t^{in}, t^{out}\} \not\subseteq \bigcup\{Z \mid \pi_Z \in \mathcal{P}\}$。

证明：
这直接由定义得出；特别是由定义 3.2 得出。 □

我们刚才介绍的只是一个记号上的便利，便于简单地添加新库所和移除已有库所，而无需显式修改流关系。

例 4.2。考虑图 2(a)中的 $\mathcal{N}_0$。按照上述约定，$\mathrm{p}_1 = \pi_{a^{out}}$ 和 $\mathrm{p}_3 = \pi_{a^{in}\mathrm{c}^{out}\mathrm{e}^{out}}$。$\mathcal{N}_0$的库所有：$\pi_{a^{out}}$、$\pi_{b^{out}}$、$\pi_{a^{in}\mathrm{c}^{out}\mathrm{e}^{out}}$、$\pi_{b^{in}\mathrm{c}^{out}\mathrm{e}^{out}}$、$\pi_{a^{in}\mathrm{c}^{out}\mathrm{f\,out}}$、$\pi_{b^{in}\mathrm{c}^{out}\mathrm{f}^{out}}$、$\pi_{a^{in}\mathrm{d}^{out}\mathrm{e}^{out}}$、$\pi_{b^{in}\mathrm{d}^{out}\mathrm{e}^{out}}$、$\pi_{a^{in}\mathrm{d}^{out}\mathrm{f\,out}}$，以及 $\pi_{b^{in}\mathrm{d}^{out}\mathrm{f}^{out}}$。此外，$\mathrm{a} \in {}^\bullet\mathrm{p}_3$，因为 $\mathrm{p}_3 = \pi_Z$ 且 $\mathrm{a}^{in} \in \mathrm{Z}$。◇

在本节中，组合标记网的两个分布式库所 $\mathcal{P}$ 和 $\mathcal{P}'$时，我们假设它们是分离的，也就是说，它们连接到互不相交的变迁集合，即
${}^\bullet\mathcal{P}^\bullet \cap {}^\bullet\mathcal{P}'^\bullet = \varnothing$。

构造新分布式库所的第一种方法很简单，就是取它们的集合并。也就是说，$\mathcal{P} \oplus \mathcal{P}' = \mathcal{P} \cup \mathcal{P}'$是两个不相交分布式库所 $\mathcal{P}$和 $\mathcal{P}'$的并集。

The定理 4.3. 设 $\mathcal{P}$和 $\mathcal{P}'$是分离的分布式库所，且$U = {}^\bullet(\mathcal{P} \oplus \mathcal{P}')^\bullet$.

1. 输入$_{\mathcal{P}\oplus\mathcal{P}'}$ = 输入$_{\mathcal{P}}$ ∪输入$_{\mathcal{P}'}$，移除$_{\mathcal{P}\oplus\mathcal{P}'}$ = 移除$_{\mathcal{P}}$ ∪移除$_{\mathcal{P}'}$，以及读取$_{\mathcal{P}\oplus\mathcal{P}'}$ = 读取$_{\mathcal{P}}$ ∪读取$_{\mathcal{P}'}$

2. $ins_{\mathcal{P}} = \varnothing = ins_{\mathcal{P}'}$ 蕴含 $inseq^{\mathcal{P}\oplus\mathcal{P}'} = cinseq_{\mathcal{P}\oplus\mathcal{P}'} = \varnothing$，并且 $ins_{\mathcal{P}} \neq \varnothing \neq ins_{\mathcal{P}'}$ 蕴含

$$\begin{aligned} inseq_{\mathcal{P}\oplus\mathcal{P}'} &= \{\sigma \in U^+ \mid \sigma|_{{}^\bullet\mathcal{P}^\bullet} \in \{\lambda\} \cup inseq_{\mathcal{P}} \wedge \sigma|_{{}^\bullet\mathcal{P}'^\bullet} \in \{\lambda\} \cup inseq_{\mathcal{P}'}\} \\ cinseq_{\mathcal{P}\oplus\mathcal{P}'} &= \{\sigma \in U^+ \mid \sigma|_{{}^\bullet\mathcal{P}^\bullet} \in cinseq_{\mathcal{P}} \wedge \sigma|_{{}^\bullet\mathcal{P}'^\bullet} \in cinseq_{\mathcal{P}'}\} \end{aligned} \tag{9}$$

因此 $inseq_{\mathcal{P}\oplus\mathcal{P}'} \subseteq \mathrm{pref}(cinseq^{\mathcal{P}\oplus\mathcal{P}'})$。

3. $rem_{\mathcal{P}} = \varnothing = rem_{\mathcal{P}'}$ 蕴含 $outseq^{\mathcal{P}\oplus\mathcal{P}'} = coutseq_{\mathcal{P}\oplus\mathcal{P}'} = \varnothing$，并且 $rem_{\mathcal{P}} \neq \varnothing \neq rem_{\mathcal{P}'}$ 蕴含

$$\begin{aligned} outseq_{\mathcal{P}\oplus\mathcal{P}'} &= \{t \circ \sigma \in U^+ \mid t \in rem_{\mathcal{P}\oplus\mathcal{P}'} \wedge \sigma|_{{}^\bullet\mathcal{P}^\bullet} \in read^*_{\mathcal{P}} \cup read^*_{\mathcal{P}} \circ outseq_{\mathcal{P}} \\ &\qquad \wedge \sigma|_{{}^\bullet\mathcal{P}'^\bullet} \in read^*_{\mathcal{P}'} \cup read^*_{\mathcal{P}'} \circ outseq_{\mathcal{P}'}\} \\ coutseq_{\mathcal{P}\oplus\mathcal{P}'} &= \{t \circ \sigma \in U^+ \mid t \in rem_{\mathcal{P}\oplus\mathcal{P}'} \wedge \sigma|_{{}^\bullet\mathcal{P}^\bullet} \in read^*_{\mathcal{P}} \circ coutseq_{\mathcal{P}} \\ &\qquad \wedge \sigma|_{{}^\bullet\mathcal{P}'^\bullet} \in read^*_{\mathcal{P}'} \circ coutseq_{\mathcal{P}'}\} \end{aligned} \tag{10}$$

因此 输出序列$_{\mathcal{P}\oplus\mathcal{P}'}$ ⊆前缀(输入序列$^{\mathcal{P}\oplus\mathcal{P}'}$)。

4. 以下陈述等价:

(a) $\mathscr{P}\oplus\mathscr{P}'$ 是一个分布式库所。(b)
以下三种情况之一成立:

$$\begin{aligned} ins_{\mathscr{P}}=\varnothing=ins_{\mathscr{P}'} \quad &\wedge \quad rem_{\mathscr{P}}=\varnothing=rem_{\mathscr{P}'} \\ ins_{\mathscr{P}}=\varnothing=ins_{\mathscr{P}'} \quad &\wedge \quad rem_{\mathscr{P}}\neq\varnothing\neq rem_{\mathscr{P}'} \\ ins_{\mathscr{P}}\neq\varnothing\neq ins_{\mathscr{P}'} \quad &\wedge \quad rem_{\mathscr{P}}=\varnothing=rem_{\mathscr{P}'}\ . \end{aligned}$$

5. 以下陈述等价:

(a) $\mathscr{P}\oplus\mathscr{P}'$ 是一个纯分布式库所。(b) $\mathscr{P}$ 和 $\mathscr{P}'$ 也都是纯分布式库所，并且输入$_{\mathscr{P}}=\varnothing=$输入$_{\mathscr{P}'}$ 或移除$_{\mathscr{P}}=\varnothing=$移除$_{\mathscr{P}'}$。

此外，如果 $\mathscr{P}\oplus\mathscr{P}'$ 是纯的，那么有:

$$\begin{aligned} outseq_{\mathscr{P}\oplus\mathscr{P}'} &= \{\sigma\in U^{+}\mid \sigma|_{\bullet\mathscr{P}\bullet}\in\{\lambda\}\cup outseq_{\mathscr{P}}\wedge\sigma|_{\bullet\mathscr{P}'\bullet}\in\{\lambda\}\cup outseq_{\mathscr{P}'}\} \\ coutseq_{\mathscr{P}\oplus\mathscr{P}'} &= \{\sigma\in U^{+}\mid \sigma|_{\bullet\mathscr{P}\bullet}\in coutseq_{\mathscr{P}}\wedge\sigma|_{\bullet\mathscr{P}'\bullet}\in coutseq_{\mathscr{P}'}\}\ . \end{aligned} \tag{11}$$

证明:
(1,2,3) 这些部分直接由定义以及以下直观观察得出: $\mathscr{P}\oplus\mathscr{P}'$ 的行为是 $\mathscr{P}$ 和 $\mathscr{P}'$的行为的并发组合。

(4) 为证明蕴含关系（(a) $\Longrightarrow$ (b)），我们首先观察到，根据定义 3.2(2)，$ins_{\mathscr{P}}\neq\varnothing\neq rem_{\mathscr{P}'}$ 以及$rem_{\mathscr{P}}\neq\varnothing\neq ins_{\mathscr{P}'}$ 均不成立。其次，$ins_{\mathscr{P}}=\varnothing\neq ins_{\mathscr{P}'}$ 和$ins_{\mathscr{P}}\neq\varnothing=ins_{\mathscr{P}'}$ 也不成立，因为在这些情况下有$inseq_{\mathscr{P}\oplus\mathscr{P}'}\neq\varnothing$ 和$cinseq_{\mathscr{P}\oplus\mathscr{P}'}=\varnothing$。类似地，$rem_{\mathscr{P}}=\varnothing\neq rem_{\mathscr{P}'}$ 以及$rem_{\mathscr{P}}\neq\varnothing=rem_{\mathscr{P}'}$ 是不可能的。因此，(b) 成立。

蕴含式 ((a) $\Longrightarrow$(b)) 可由 (1、2、3) 部分以及如下观察得出：定义 3.2(2) 成立，因为 (b) 蕴含 $ins_{\mathscr{P}\oplus\mathscr{P}'}\times rem_{\mathscr{P}\oplus\mathscr{P}'}=\varnothing$。

(5) 由 (3、4) 部分和命题 4.1 可得（注意，根据定义 3.2(1)，$ins_{\mathscr{P}}=\varnothing=rem_{\mathscr{P}}$ 和 $ins_{\mathscr{P}'}=\varnothing=rem_{\mathscr{P}'}$ 不可能；$read_{\mathscr{P}}=\varnothing=read_{\mathscr{P}'}$，以及 ${}^{\bullet}\mathscr{P}^{\bullet}\neq\varnothing\neq{}^{\bullet}\mathscr{P}'^{\bullet}$）。□

为了支持并发系统的逐步构造（或层次分析），组件之间的某种同步通常发挥着关键作用。从网构造的角度看，不同进程线程之间适当的同步可以通过库所集合的笛卡尔积来实现。

定义 4.4。（笛卡尔积）
两个非空库所集合 Q和R的笛卡尔积 为 $Q\otimes R=\{\pi_{Z\cup Z'}\mid\pi_Z\in Q\wedge\pi_{Z'}\in R\}$。 ◇

例 4.5. 考虑图3(b)中的分布式库所 $\mathscr{P}'$。我们观察到

$$\begin{aligned} \mathscr{P}' &= \{\pi_{a^{in}c^{out}e^{out}},\pi_{b^{in}c^{out}e^{out}},\pi_{a^{in}c^{out}f^{out}},\pi_{b^{in}c^{out}f^{out}},\pi_{a^{in}d^{out}e^{out}},\pi_{b^{in}d^{out}e^{out}},\pi_{a^{in}d^{out}f^{out}},\pi_{b^{in}d^{out}f^{out}}\} \\ &= \{\pi_{a^{in}},\pi_{b^{in}}\}\otimes\{\pi_{c^{out}e^{out}},\pi_{d^{out}e^{out}},\pi_{c^{out}f^{out}},\pi_{d^{out}f^{out}}\} \\ &= \{\pi_{a^{in}},\pi_{b^{in}}\}\otimes(\{\pi_{c^{out}},\pi_{d^{out}}\}\otimes\{\pi_{e^{out}},\pi_{f^{out}}\})\ . \end{aligned}$$

图4(*a*)中 $\mathscr{N}_1$ 的分布式库所 $\mathscr{P}_1 = \{q_3, q_4, q_5, q_6\}$ 也有类似的分解；图4(*c*)中 $\mathscr{N}_2$ 的分布式库所 $\mathscr{P}_2 = \{r_1, r_2, r_3, r_4\}$ 亦然：

$$\begin{array}{rcl} \mathscr{P}_1 & = & \{\pi_{a^{in}c^{out}}, \pi_{b^{in}c^{out}}, \pi_{a^{in}d^{out}}, \pi_{b^{in}d^{out}}\} \quad = \quad \{\pi_{a^{in}}, \pi_{b^{in}}\} \otimes \{\pi_{c^{out}}, \pi_{d^{out}}\} \\ \mathscr{P}_2 & = & \{\pi_{a^{in}c^{in}}, \pi_{b^{in}c^{in}}, \pi_{a^{in}d^{in}}, \pi_{b^{in}d^{in}}\} \quad = \quad \{\pi_{a^{in}}, \pi_{b^{in}}\} \otimes \{\pi_{c^{in}}, \pi_{d^{in}}\} . \end{array}$$

◇

下一个结果列出笛卡尔积的一些有用性质。

定理 4.6. 设 $\mathscr{P}$ 和 $\mathscr{P}'$ 是分离的分布式库所。

1. $\mathscr{P}'' = \mathscr{P} \otimes \mathscr{P}'$ 是一个分布式库所，使得 输入$_{\mathscr{P}''}$ = 输入$_{\mathscr{P}}$ ∪ 输入$_{\mathscr{P}'}$, 输入$_{rem''}$ = 移除$_{\mathscr{P}}$ ∪ 移除$_{\mathscr{P}'}$, 且 读取$_{rem''}$ = 读取$_{\mathscr{P}}$ ∪ 读取$_{\mathscr{P}'}$，并且：

$$\begin{array}{rcl} inseq_{\mathscr{P}''} & = & inseq_{\mathscr{P}} \cup cinseq_{\mathscr{P}} \circ read^*_{\mathscr{P}''} \cup inseq_{\mathscr{P}'} \cup cinseq_{\mathscr{P}'} \circ read^*_{\mathscr{P}''} \\ cinseq_{\mathscr{P}''} & = & cinseq_{\mathscr{P}} \circ read^*_{\mathscr{P}''} \cup cinseq_{\mathscr{P}'} \circ read^*_{\mathscr{P}''} \\ outseq_{\mathscr{P}''} & = & outseq_{\mathscr{P}} \cup outseq_{\mathscr{P}'} \\ coutseq_{\mathscr{P}''} & = & coutseq_{\mathscr{P}} \cup coutseq_{\mathscr{P}'} \end{array} \tag{12}$$

2. 如果 $\mathscr{P}$ 和 $\mathscr{P}'$ 是纯的，那么 $\mathscr{P}''$ 是一个纯分布式位置，使得：

$$\begin{array}{rcl} inseq_{\mathscr{P}''} & = & inseq_{\mathscr{P}} \cup inseq_{\mathscr{P}'} \\ cinseq_{\mathscr{P}''} & = & cinseq_{\mathscr{P}} \cup cinseq_{\mathscr{P}'} . \end{array} \tag{13}$$

证明：
(1) 我们首先注意到，由于 $\mathscr{P}$ 和 $\mathscr{P}'$ 是分离的，对于每个 $\pi_{Z''} \in \mathscr{P}''$，都存在唯一的 $\pi_Z \in \mathscr{P}$和 $\pi_{Z'} \in \mathscr{P}'$，使得 $Z'' = Z \uplus Z'$。接着我们注意到，根据 $\mathscr{P} \neq \varnothing \neq \mathscr{P}'$，对于每个形如$t^{in}$ 或t^{out}的x，以下结论成立：

$$x \in \bigcup\{Z'' \mid \pi_{Z''} \in \mathscr{P}''\} \iff x \in \bigcup\{Z \mid \pi_Z \in \mathscr{P}\} \uplus \bigcup\{Z' \mid \pi_{Z'} \in \mathscr{P}'\} . \tag{14}$$

根据这些观察和定义，对于所有的 $t \in {}^\bullet\mathscr{P}^\bullet$ 和 $w \in {}^\bullet\mathscr{P}'^\bullet$，我们有：

$$\begin{array}{ll} {}^\bullet t \cap \mathscr{P}'' = ({}^\bullet t \cap \mathscr{P}) \otimes \mathscr{P}' & {}^\bullet w \cap \mathscr{P}'' = ({}^\bullet w \cap \mathscr{P}') \otimes \mathscr{P} \\ t^\bullet \cap \mathscr{P}'' = (t^\bullet \cap \mathscr{P}) \otimes \mathscr{P}' & w^\bullet \cap \mathscr{P}'' = (w^\bullet \cap \mathscr{P}') \otimes \mathscr{P} \end{array} \tag{15}$$

此外，对于所有的 t，$u \in {}^\bullet\mathscr{P}^\bullet$和 w，$v \in {}^\bullet\mathscr{P}'^\bullet$，我们有：

$$\begin{array}{ll} {}^\bullet t \cap {}^\bullet u \cap \mathscr{P}'' = ({}^\bullet t \cap {}^\bullet u \cap \mathscr{P}) \otimes \mathscr{P}' & {}^\bullet w \cap {}^\bullet v \cap \mathscr{P}'' = ({}^\bullet w \cap {}^\bullet v \cap \mathscr{P}') \otimes \mathscr{P} \\ t^\bullet \cap u^\bullet \cap \mathscr{P}'' = (t^\bullet \cap u^\bullet \cap \mathscr{P}) \otimes \mathscr{P}' & w^\bullet \cap v^\bullet \cap \mathscr{P}'' = (w^\bullet \cap v^\bullet \cap \mathscr{P}') \otimes \mathscr{P} \\ {}^\bullet t \cap u^\bullet \cap \mathscr{P}'' = ({}^\bullet t \cap u^\bullet \cap \mathscr{P}) \otimes \mathscr{P}' & {}^\bullet w \cap v^\bullet \cap \mathscr{P}'' = ({}^\bullet w \cap v^\bullet \cap \mathscr{P}') \otimes \mathscr{P} \end{array} \tag{16}$$

以及

$$\begin{array}{ll}(^\bullet t\cup t^\bullet)\cap\mathscr{P}''=((^\bullet t\cup t^\bullet)\cap\mathscr{P})\otimes\mathscr{P}' & (^\bullet w\cup v^\bullet)\cap\mathscr{P}''=((^\bullet w\cup v^\bullet)\cap\mathscr{P}')\otimes\mathscr{P}\\ (t^\bullet\cup u^\bullet)\cap\mathscr{P}''=((t^\bullet\cup u^\bullet)\cap\mathscr{P})\otimes\mathscr{P}' & (w^\bullet\cup v^\bullet)\cap\mathscr{P}''=((w^\bullet\cup v^\bullet)\cap\mathscr{P}')\otimes\mathscr{P}\,.\end{array}\tag{17}$$

通过方程式 (15)、方程式 (16) 和方程式 (17) 得到的性质，可以证明所需结果。例如，若 $t\in{}^\bullet\mathscr{P}^\bullet$，则根据方程式 (15)，有：

$$\begin{aligned}t\in ins_{\mathscr{P}''}&\iff t^\bullet\cap\mathscr{P}''\neq\varnothing={}^\bullet t\cap\mathscr{P}''\iff(t^\bullet\cap\mathscr{P})\otimes\mathscr{P}'\neq\varnothing=({}^\bullet t\cap\mathscr{P})\otimes\mathscr{P}'\\&\iff t^\bullet\cap\mathscr{P}\neq\varnothing={}^\bullet t\cap\mathscr{P}\iff t\in ins_{\mathscr{P}}\,.\end{aligned}$$

因此，$ins_{\mathscr{P}''}=ins_{\mathscr{P}}\cup ins_{\mathscr{P}'}$。类似地，我们可以证明 $rem_{\mathscr{P}''}=rem_{\mathscr{P}}\cup rem_{\mathscr{P}'}$，以及$read_{\mathscr{P}''}=read_{\mathscr{P}}\cup read_{\mathscr{P}'}$。

为了证明 $outseq_{\mathscr{P}}=outseq_{\mathscr{P}''}$，我们首先注意到 $rem_{\mathscr{P}}\cup rem_{\mathscr{P}'}=\{\sigma\in outseq_{\mathscr{P}''}\mid|\sigma|=1\}$，然后取 $t_1\ldots t_k\in outseq_{\mathscr{P}}\cap outseq_{\mathscr{P}''}(k\geq 1)$。根据方程式 (15)，有：

$$\mathscr{P}''\setminus(({}^\blacktriangleright t_1\cap\mathscr{P})\otimes\mathscr{P}'\cup\cdots\cup({}^\blacktriangleright t_k\cap\mathscr{P})\otimes\mathscr{P}')=(\mathscr{P}\setminus(({}^\blacktriangleright t_1\cup\cdots\cup{}^\blacktriangleright t_i)\cap\mathscr{P}))\otimes\mathscr{P}'\tag{18}$$

接下来我们考虑两种情形。

情形 *1:* $t_1\ldots t_k t\in outseq_{\mathscr{P}}$。那么 ${}^\bullet t\cap\mathscr{P}\subseteq\mathscr{P}\setminus(({}^\blacktriangleright t_1\cup\cdots\cup{}^\blacktriangleright t_i)\cap\mathscr{P})$。因此，根据方程式 (15) 和方程式 (18)，

$$\begin{aligned}{}^\bullet t\cap\mathscr{P}''&=({}^\bullet t\cap\mathscr{P})\otimes\mathscr{P}'\subseteq(\mathscr{P}\setminus(({}^\blacktriangleright t_1\cup\cdots\cup{}^\blacktriangleright t_k)\cap\mathscr{P})\otimes\mathscr{P}'\\&=\mathscr{P}''\setminus(({}^\blacktriangleright t_1\cap\mathscr{P})\otimes\mathscr{P}'\cup\cdots\cup({}^\blacktriangleright t_k\cap\mathscr{P})\otimes\mathscr{P}')=\mathscr{P}''\setminus({}^\blacktriangleright t_1\cap\mathscr{P}''\cup\cdots\cup{}^\blacktriangleright t_k\cap\mathscr{P}'')\,.\end{aligned}$$

因此 $t_1\ldots t_k t\in$输出序列$_{\mathscr{P}''}$。

情形 *2*：$t_1\ldots t_k t\in$输出序列$_{\mathscr{P}''}$。接下来注意到，$t\notin{}^\bullet\mathscr{P}'^\bullet$，这是由方程式(16)和$t_1\in\mathscr{P}$得出的。因此$t\in{}^\bullet\mathscr{P}^\bullet$，我们采用与情形 1 类似的方法继续证明。

因此，输出序列$_{\mathscr{P}}$ =输出序列$_{\mathscr{P}''}$。方程式(12)的其余部分可用类似的方法证明。

(2) 这一部分 由第 (1) 部分、命题 4.1、 $\mathscr{P}$ 与$\mathscr{P}'$的可分离性以及方程式(12).

□

直观地说，定理 4.6 中的这一对分布式库所具有这样的性质：进程线程通过 $\mathscr{P}$ 的输入序列或 $\mathscr{P}'$ 的输入序列进入$\mathscr{P}\otimes\mathscr{P}'$ 后，可以通过 $\mathscr{P}$ 的输出序列或 $\mathscr{P}'$的输出序列离开 $\mathscr{P}\otimes\mathscr{P}'$。

例 4.7。 回到例 4.5，库所集合 $\mathscr{P}'$、 $\mathscr{P}_1$ 和 $\mathscr{P}_2$ 的分解，连同定理 4.3 和 4.6 以及命题 3.5，表明这三个库所集合是分布式库所。◇

5. 盒代数和分布式库所

我们研究盒代数 [1] 的一个片段，重点关注由四种控制流算子导出的网。省略通信算子和同步算子，并不会削弱下文所得结果的重要性，因为这类算子生成的是新变迁，而非库所。

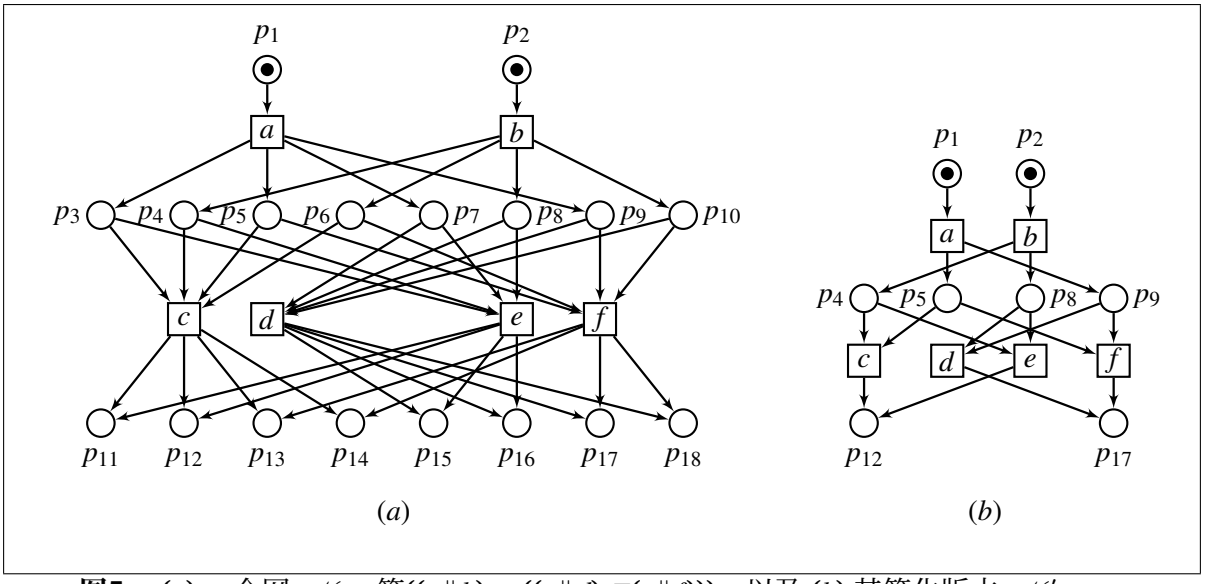

图5： (*a*) 一个网 $\mathcal{N}_3 =$箱$((a \,\|\, b);\ ((c \,\|\, d)\ \square (e \,\|\, f)))$，以及 (*b*) 其简化版本 $\mathcal{N}_3'$。

本文中的进程表达式（或 盒表达式）按以下方式导出：

$$E \ ::= \ a \ \mid \ E;E \ \mid \ E \,\square\, E \ \mid \ E \,\|\, E \ \mid \ [E * E * E] \tag{19}$$

其中 a 是一个原子动作。此外，由于我们只处理控制流方面，且动作不具有任何特殊的语义属性，因此我们假设在一个盒表达式中每个动作最多出现一次，这简化了从盒表达式组合构造网的过程。[1]

直观地说，a 表示一个可以执行原子动作a并终止的过程，E；F 表示两个过程的顺序组合，$E \,\square\, F$表示选择组合，$E \,\|F$表示并行组合，而 $[E * F * G]$ 表示迭代：首先执行E，然后执行F 任意次（包括零次），最后通过执行G终止整个构造。

方程式(19)中的语法允许使用会产生非安全网的盒表达式。然而，通过修改语法可以简单地避免这一问题：

$$\begin{array}{lcl} E & ::= & a \ \mid \ E;E \ \mid \ E \,\square\, E \ \mid \ [E * H * E] \ \mid \ E \,\|\, E \\ H & ::= & a \ \mid \ E;E \ \mid \ H \,\square\, H \ \mid \ [E * H * E] \end{array} \tag{20}$$

从盒表达式到适用于本文所用记法的盒子的标准转换分两步进行。首先，该转换为每个盒表达式 E 关联三个库所集合，$\Pi_E = (\Pi_E^e,\ \Pi_E^x,\ \Pi_E^i)$。然后，与 E关联的标记网定义如下：

$$\mathrm{box}(E) = (\Pi_E^e \cup \Pi_E^x \cup \Pi_E^i,\ T,\ Fl,\ \Pi_E^e)\ ,$$

[1]另一种方法是为动作关联唯一标签，例如使用动作在盒表达式语法树中的位置。

其中，T 是 E 中出现的动作，Fl 是根据方程式 (8) 得出的弧。集合 Π_E^e 包含入口位置 （初始时已标记），Π_E^x 包含出口位置（初始时为空），Π_E^i 包含内部位置 （初始时为空）。入口位置和出口位置在盒子的组合推导中发挥积极作用，而内部位置则不作任何更改地保留在所得盒子中。

Π_E 的语法驱动推导如下:

$$\begin{array}{lcl}
\Pi_a & = & (\mathscr{O}_a,\ \mathscr{I}_a,\ \varnothing) \\
\Pi_{E\parallel F} & = & (\Pi_E^e \cup \Pi_F^e,\ \Pi_E^x \cup \Pi_F^x,\ \Pi_E^i \cup \Pi_F^i) \\
\Pi_{E;F} & = & (\Pi_E^e,\ \Pi_F^x,\ \Pi_E^i \cup \Pi_F^i \cup (\Pi_E^x \otimes \Pi_F^e)) \\
\Pi_{E\,\square\, F} & = & (\Pi_E^e \otimes \Pi_F^e,\ \Pi_E^x \otimes \Pi_F^x,\ \Pi_E^i \cup \Pi_F^i) \\
\Pi_{[E*F*G]} & = & (\Pi_E^e,\ \Pi_G^x,\ \Pi_E^i \cup \Pi_F^i \cup \Pi_G^i \cup ((\Pi_E^x \otimes \Pi_F^x) \otimes (\Pi_G^e \otimes \Pi_F^e)))
\end{array} \tag{21}$$

例 5.1. 对于盒表达式（$a \parallel b$）；（$c \parallel d$），我们有:

$$\begin{array}{lcl}
\Pi_{(a\parallel b);(c\parallel d)} & = & (\Pi^e_{a\parallel b},\ \Pi^x_{c\parallel d},\ \Pi^i_{a\parallel b} \cup \Pi^i_{c\parallel d} \cup (\Pi^x_{a\parallel b} \otimes \Pi^e_{c\parallel d})) \\
& = & (\Pi^e_a \cup \Pi^e_b,\ \Pi^x_c \cup \Pi^x_d,\ \Pi^i_a \cup \Pi^i_b \cup \Pi^i_c \cup \Pi^i_d \cup ((\Pi^x_a \cup \Pi^x_b) \otimes (\Pi^e_c \cup \Pi^e_d))) \\
& = & (\mathscr{O}_a \cup \mathscr{O}_b,\ \mathscr{I}_c \cup \mathscr{I}_d,\ \varnothing \cup \varnothing \cup \varnothing \cup \varnothing \cup ((\mathscr{I}_a \cup \mathscr{I}_b) \otimes (\mathscr{O}_c \cup \mathscr{O}_d))) \\
& = & (\{\pi_{a^{out}}, \pi_{b^{out}}\},\ \{\pi_{c^{in}}, \pi_{d^{in}}\},\ \{\pi_{a^{in}c^{out}}, \pi_{b^{in}c^{out}}, \pi_{a^{in}d^{out}}, \pi_{b^{in}d^{out}}\})\,.
\end{array}$$

因此，box((a ∥b); (c ∥d))的库所为 $\pi_{a^{out}}$, $\pi_{a^{out}}$, $\pi_{c^{in}}$, $\pi_{d^{in}}$, $\pi_{a^{in}c^{out}}$, $\pi_{b^{in}c^{out}}$, $\pi_{a^{in}d^{out}}$ 和 $\pi_{b^{in}d^{out}}$。此外，初始标识为 $M_{init} = \{\pi_{a^{out}}, \pi_{b^{out}}\}$。因此，box((a ∥b) ; (c ∥d)) 是图 4(*a*)中的标记网 $\mathscr{N}_1$。

对于盒表达式（$a \parallel b$)□（$c \parallel d$），转换过程如下:

$$\begin{array}{lcl}
\Pi_{(a\parallel b)\,\square\,(c\parallel d)} & = & (\Pi^e_{a\parallel b} \otimes \Pi^e_{c\parallel d},\ \Pi^x_{a\parallel b} \otimes \Pi^x_{c\parallel d},\ \Pi^i_{a\parallel b} \cup \Pi^i_{c\parallel d}) \\
& = & ((\mathscr{O}_a \cup \mathscr{O}_b) \otimes (\mathscr{O}_c \cup \mathscr{O}_d),\ (\mathscr{I}_a \cup \mathscr{I}_b) \otimes (\mathscr{I}_c \cup \mathscr{I}_d),\ \varnothing) \\
& = & (\{\pi_{a^{out}c^{out}}, \pi_{a^{out}d^{out}}, \pi_{b^{out}c^{out}}, \pi_{b^{out}d^{out}}\},\ \{\pi_{a^{in}c^{in}}, \pi_{a^{in}d^{in}}, \pi_{b^{in}c^{in}}, \pi_{b^{in}d^{in}}\},\ \varnothing)\,.
\end{array}$$

因此，box((a ∥b)□(c ∥d))的库所为 $\pi_{a^{out}c^{out}}$, $\pi_{a^{out}d^{out}}$, $\pi_{b^{out}c^{out}}$, $\pi_{b^{out}d^{out}}$, $\pi_{a^{in}c^{in}}$, $\pi_{a^{in}d^{in}}$, $\pi_{b^{in}c^{in}}$ 和 $\pi_{b^{in}d^{in}}$。此外，初始标识为 $M_{init} = \{\pi_{a^{out}c^{out}}, \pi_{a^{out}d^{out}}, \pi_{b^{out}c^{out}}, \pi_{b^{out}d^{out}}\}$。因此，box((a ∥b)□(c ∥d)) 是图 4(c)中的标记网 $\mathscr{N}_2$。

转换的另外两个示例见图 1(*a*) 和图 5(*a*)。 ◇

从盒表达式 E 到网 box(E) 的转换会生成一个标记网，其中可以识别出分布式库所 distpl(E)，并以此为基础执行定理 3.8 中所述的归约过程。为此，我们定义一个由分布式库所组成的三元组 $\Delta_E = (\Delta_E^e,\ \Delta_E^x,\ \Delta_E^i)$，使得 Δ_E^e 覆盖 box(E) 的入口位置，Δ_E^x 覆盖出口位置，而 Δ_E^i 覆盖内部位置。

Δ_E 的语法驱动推导如下：

$$\begin{array}{lcl}
\Delta_a & = & (\{\mathscr{O}_a\},\ \{\mathscr{I}_a\},\ \varnothing) \\
\Delta_{E\parallel F} & = & (\Delta_E^e \cup \Delta_F^e,\ \Delta_E^x \cup \Delta_F^x,\ \Delta_E^i \cup \Delta_F^i) \\
\Delta_{E;F} & = & (\Delta_E^e,\ \Delta_F^x,\ \Delta_E^i \cup \Delta_F^i \cup \{\Pi_E^x \otimes \Pi_F^e\}) \\
\Delta_{E\,\square\, F} & = & (\{\Pi_E^e \otimes \Pi_F^e\},\ \{\Pi_E^x \otimes \Pi_F^x\},\ \Delta_E^i \cup \Delta_F^i) \\
\Delta_{[E*F*G]} & = & (\Delta_E^e,\ \Delta_G^x,\ \Delta_E^i \cup \Delta_F^i \cup \Delta_G^i \cup \{(\Pi_E^x \otimes \Pi_F^x) \otimes (\Pi_F^e \otimes \Pi_G^e)\})\ .
\end{array} \tag{22}$$

于是，我们将E 的分布式库所 定义为 $\mathrm{distpl}(E) = \Delta_E^e \cup \Delta_E^x \cup \Delta_E^i$。

注意，公式（22）可视为一种从盒表达式到箱的替代翻译，该转换通过分布式库所而非普通位置来实现。

例 5.2. 参见图4(a,c)，我们有：

$$\begin{array}{lcl}
\mathrm{distpl}((a\parallel b);(c\parallel d)) & = & \{\{q_1\},\{q_2\},\{q_3,q_4,q_5,q_6\},\{q_7\},\{q_8\}\} \\
 & = & \{\{\pi_{a^{out}}\},\{\pi_{b^{out}}\},\{\pi_{a^{in}c^{out}},\pi_{b^{in}c^{out}},\pi_{a^{in}d^{out}},\pi_{b^{in}d^{out}}\},\{\pi_{c^{in}}\},\{\pi_{d^{in}}\}\} \\
\mathrm{distpl}((a\parallel b)\,\square\,(c\parallel d)) & = & \{\{r_1,r_2,r_3,r_4\},\{r_5,r_6,r_7,r_8\}\} \\
 & = & \{\{\pi_{a^{out}c^{out}},\pi_{a^{out}d^{out}},\pi_{b^{out}c^{out}},\pi_{b^{out}d^{out}}\},\{\pi_{a^{in}c^{in}},\pi_{a^{in}d^{in}},\pi_{b^{in}c^{in}},\pi_{b^{in}d^{in}}\}\}\ .
\end{array}$$

此外，在图5(a)中，$\Delta_{(a\parallel b);((c\parallel d)\,\square\,(e\parallel f))} = \{\{p_1,p_2\},\{p3,\ldots,p_{10}\},\{p12,\ldots,p_{18}\}\}$。 ◇

于是，我们得到箱及其分布式库所的一些基本性质。

命题 5.3. 设E 是一个由公式（20）中的语法定义的盒表达式，$\mathrm{box}(E) = (P, T, Fl, M_{init})$，且$M$ 是 $\mathrm{box}(E)$ 的一个可达标识。

1. 箱(E)是一个安全标记网。
2. $\Pi_E^e(= M_{init}) = \{p \in P \mid {}^\bullet p = \varnothing\}$ and $\Pi_E^x = \{p \in P \mid p^\bullet = \varnothing\}$.
3. Π_E^x 是从M可达的一个标识。
4. $M \subseteq \Pi_E^e$蕴含$M = \Pi_E^e$，且 $\Pi_E^x \subseteq M$蕴含$M = \Pi_E^x$。
5. $\Delta_E^e = \{\Pi_E^e\}$和 $\Delta_E^x = \{\Pi_E^x\}$。
6. distpl(E)是 一组两两不相交的纯分布式库所，覆盖箱(E)中的库所 .
7. Π_E^e 和 Π_E^x 是分离的分布式库所，当E 由公式（20）中语法的第二行生成时。
8. 如果 $\mathscr{P} \neq \mathscr{P}' \in \mathrm{distpl}(E)$，则 ${}^\bullet\mathscr{P} \cap {}^\bullet\mathscr{P}' = \mathscr{P}^\bullet \cap \mathscr{P}'^\bullet = \varnothing$。

证明：
(1,2,3,4) 这些部分是在 [1, 5]中证明的结果。

(5,6,7,8) 这些部分可通过对E 的结构以及语法驱动的定义方程式（20）、方程式（21）和方程式（22）进行同时归纳得到，并使用部分 (1,2,3,4)、定理 4.3 和 4.6、命题 3.5，以及盒表达式中每个动作至多使用一次这一假设。□

6. 余图与分布式库所

为了有效化简分离的纯分布式位置 Δ_E（这些位置将盒(E)的库所集合划分开来），我们将使用没有自环的无向图$G=(V,A)$，其中V是有限顶点集合，A是边集合。仅由一个顶点v构成的图$(\{v\},\varnothing)$将记作v。

图G的补图运算记作$\overline{G}$；两个图G和G'的不交并记作$G \uplus G'$。两个图G和G'的联结是由它们的不交并构造而成的图$G - G'$，其构造方式是在G的每个顶点与G'的每个顶点之间添加一条边。请注意，余图可以表示为由单个顶点和空余图通过不交并与联结算子构成的表达式，其表示大小即为该表达式的长度。

图G中的团是由clq中的顶点构成的非空集合，其中任意两个顶点之间都有边相连（注意，单顶点图也是一个团）。若团clq不是图G中任何其他团的子集，则称其为极大团（或极大团）。我们将其记作$clq \in \mathrm{maxclq}(G)$。

图中的一组团构成一个边团覆盖（ECC），如果对于每条边，至少存在一个团同时包含该边的两个端点（在我们的情形中，还将添加覆盖孤立顶点的单点团）。边团覆盖问题（ECCP）旨在使用数量最少的极大团来求得一个 ECC。ECCP 的一种变体是部分ECCP（PECCP），其中一些边被假定为已经覆盖。请注意，就我们的目的而言，可以使用求解（P）ECCP 的启发式算法，因为即使将每条边分别覆盖的平凡 ECC 覆盖也至多产生（二次于顶点数的）数量的团，因此其结果仍将是多项式的。

分布式库所所在的箱与下一节定义的一类图的 ECCP 之间存在直接联系。

定义 6.1。（余图 [15]）
补可约图（余图）递归定义如下：（i）单顶点图是余图，空图也是余图；（ii）余图的补图是余图；（iii）余图的不交并是余图。$\diamond$

由于联结运算可以通过不交并和补图运算来表示（注意$G - G' = \overline{G \uplus G'}$），因此，任何通过反复应用联结和不交并运算（从单顶点图开始）得到的图都是余图。

库所集合的表示，以及通过笛卡尔积生成库所集合，都可以使用隐式方式完成，即使用顶点由输入 和输出标记的图。

定义 6.2。（连接图）
连接图 $\Gamma=$ (V,A) 的顶点形式为a^{in} 和a^{out}，其中a 是盒表达式定义中使用的动作，而 $\Pi_\Gamma = \{\pi_{clq} \mid clq \in \mathrm{maxclq}(\Gamma)\}$ 是由 Γ生成的库所 。注意：每个$clq = \{a_1^{in},\ldots,a_k^{in},b_1^{out},\ldots,b_l^{out}\}$ 生成 $\pi_{clq} = \pi_{a_1^{in}\ldots a_k^{in}b_1^{out}\ldots b_l^{out}}$。 $\diamond$

下一个结果揭示了对不相交位置 $\oplus$ 和 $\otimes$进行的两种运算，与连接图上的不交并和联结运算之间的基本联系。

命题 6.3。 设 Γ 和 Γ' 是两个连接图，且 Π_Γ 和 $\Pi_{\Gamma'}$ 是彼此分离的纯分布式库所。则：

1. $\Pi_{\Gamma\uplus\Gamma'}$ 是一个纯分布式库所，满足 $\Pi_{\Gamma\uplus\Gamma'}=\Pi_\Gamma\oplus\Pi_{\Gamma'}$，条件是 $^\bullet(\Pi_\Gamma)=^\bullet(\Pi_{\Gamma'})=\varnothing$，或者 $(\Pi_\Gamma)^\bullet=(\Pi_{\Gamma'})^\bullet=\varnothing$。
2. $\Pi_{\Gamma-\Gamma'}$ 是满足 $\Pi_{\Gamma-\Gamma'}=\Pi_\Gamma\otimes\Pi_{\Gamma'}$ 的纯分布式库所。

证明：
这可由定理 4.3 和 4.6 以及以下两个显然性质得出：

$$\begin{aligned}\mathrm{maxclq}(G-G') &= \{clq\cup clq' \mid clq\in\mathrm{maxclq}(G)\wedge clq'\in\mathrm{maxclq}(G')\}\\ \mathrm{maxclq}(G\uplus G') &= \mathrm{maxclq}(G)\cup\mathrm{maxclq}(G')\,,\end{aligned}$$

其中G 和 G' 是互不相交的余图。 □

因此， Δ_E 中的分布式库所（通过反复应用叉积和并集得到）可以由适当的连接图生成。具体而言，首先为每个盒表达式 E关联三个余图 $\Gamma_E=(\Gamma^e_E、\ \Gamma^x_E、\ \Gamma^i_E)$，其中 Γ^e_E、 Γ^x_E 和 Γ^i_E 的含义分别与 $\Delta_E=(\Delta^e_E、\ \Delta^x_E、\ \Delta^i_E)$ 中的相应分量相同，例如 $\Pi_{\Gamma^i_E}=\Delta^i_E$。

我们如下生成 Γ_E：

$$\begin{aligned}\Gamma_a &= (a^{out},\, a^{in},\, \varnothing)\\ \Gamma_{E\parallel F} &= (\Gamma^e_E\uplus\Gamma^e_F,\, \Gamma^x_E\uplus\Gamma^x_F,\, \Gamma^i_E\uplus\Gamma^i_F)\\ \Gamma_{E;F} &= (\Gamma^e_E,\, \Gamma^x_F,\, \Gamma^i_E\uplus\Gamma^i_F\uplus(\Gamma^x_E-\Gamma^e_F))\\ \Gamma_{E\,\square\, F} &= (\Gamma^e_E-\Gamma^e_F,\, \Gamma^x_E-\Gamma^x_F,\, \Gamma^i_E\uplus\Gamma^i_F)\\ \Gamma_{[E*F*G]} &= (\Gamma^e_E,\, \Gamma^x_G,\, \Gamma^i_E\uplus\Gamma^i_F\uplus\Gamma^i_G\uplus((\Gamma^x_E-\Gamma^x_F)-(\Gamma^e_F-\Gamma^e_G)))\,.\end{aligned}\tag{23}$$

接下来，我们将E 的连接图定义为$CG_E=\Gamma^e_E\uplus\Gamma^x_E\uplus\Gamma^i_E$。请注意，该表示（作为余图表达式）的大小与盒表达式的大小呈线性关系。

命题 6.4。 设E 为一个由方程式（20）中的语法定义的盒表达式。

1. Γ^e_E、 Γ^x_E、 Γ^i_E是互不相交的余图，使得 $\Pi_{\Gamma^e_E}=\Delta^e_E$、 $\Pi_{\Gamma^x_E}=\Delta^x_E$和 $\Pi_{\Gamma^i_E}=\Delta^i_E$。
2. CG_E 是一个余图，使得 $\Pi_{CG_E}=\Delta_E$。

证明：
根据E 的结构进行归纳，并结合方程式（20）、方程式（21）、方程式（22）和方程式（23）中的语法驱动定义，以及命题 6.3 和“在给定的盒表达式中，每个动作至多使用一次”这一假设，即可得到该结果。请注意，推导 Δ_E 时组合得到的分布式库所彼此分离，因为对于由方程式（20）中语法的第二行生成的E， Π^e_E 和 Π^x_E 是彼此分离的分布式库所（见命题 5.3（7））。□

例 6.5. 对于盒表达式（$a\parallel b$）；（$c\parallel d$），我们有：

$$\begin{aligned}\Gamma_{(a\parallel b);(c\parallel d)} &= (\Gamma^e_{a\parallel b},\, \Gamma^x_{c\parallel d},\, \Gamma^i_{a\parallel b}\uplus\Gamma^i_{c\parallel d}\uplus(\Gamma^x_{a\parallel b}-\Gamma^e_{c\parallel d}))\\ &= (\Gamma^e_a\uplus\Gamma^e_b,\, \Gamma^x_c\uplus\Gamma^x_d,\, \Gamma^i_a\uplus\Gamma^i_b\uplus\Gamma^i_c\uplus\Gamma^i_d\uplus((\Gamma^x_a\uplus\Gamma^x_b)-(\Gamma^e_c\uplus\Gamma^e_d)))\\ &= (a^{out}\uplus b^{out},\, c^{in}\uplus d^{in},\, (a^{in}\uplus b^{in})-(c^{out}\uplus d^{out}))\,.\end{aligned}$$

因此，如图6（a）所示，$CG_{(a\parallel b);(c\parallel d)}=a^{out}\uplus b^{out}\uplus c^{in}\uplus d^{in}\uplus(a^{in}\uplus b^{in})-(c^{out}\uplus d^{out})$。 ◇

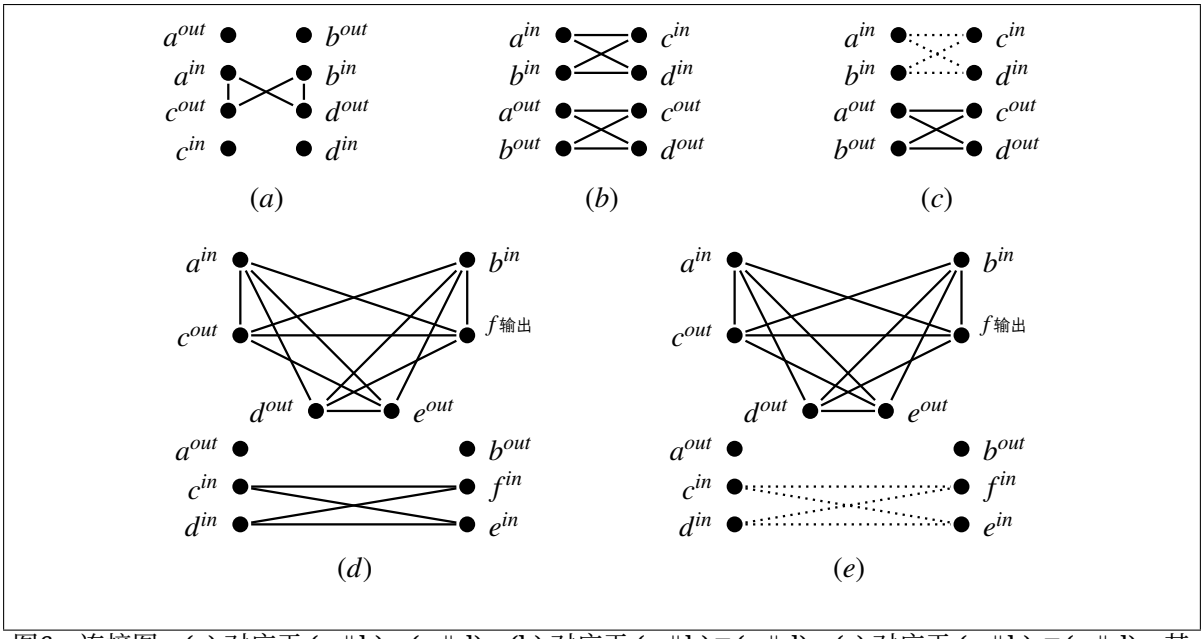

图6：连接图：(a) 对应于 (a ‖b)；(c ‖d)，(b) 对应于 (a ‖b)□(c ‖d)，(c) 对应于 (a ‖b) □(c ‖d)，其中虚线边已被覆盖，(d) 对应于 (a ‖b)；((c ‖d)□(e ‖f))，以及 (e) 对应于 (a ‖b)；((c ‖d)□(e ‖f))，其中虚线边已被覆盖。

7. 基于余图的归约

现在，我们可以介绍一个三阶段过程，用于构造由公式（20）中的语法定义的盒表达式 E 所对应的盒 E 的约简行为等价版本。

第一阶段。 使用方程式(23)，推导连接图 CG_E。

第二阶段： 应用任意算法，为CG_E寻找 $\mathscr{C}$ 的 ECCP 解。

第三阶段： 利用方程式(8)构造标记网 $\mathscr{N}_{\mathscr{C}}$，其库所集合为$P = \{\pi_{clq} \mid clq \in \mathscr{C}\}$，初始标识为$M_{init} = \{p \in P \mid {}^\bullet p = \varnothing\}$。

由于CG_E的顶点数为 $2n$，其中n是E中出现的动作数，因此至多需要 $2n(2n-1)/2 = n(2n-1)$ 个团来构成合适的 $\mathscr{C}$，所以所得网中至多有$n(2n-1)$ 个库所。有时，库所数可以小得多，例如$O(log\ n)$，如图1(*d*)中的示例所示。这显然优于原始转换的结果，后者可能产生指数规模的 box(E)。此外，还可以应用以下两项优化。

优化 I. 在阶段 I 结束时，将形如a^{in} 的顶点对之间的所有边标记为已覆盖。然后，在阶段 II 中，使用任意算法求解 PECCP。

优化 II。 在第 III 阶段结束时，删除 $\{p \in P \mid p^\bullet = \varnothing\}$ 中的所有库所。

整个过程的有效性将在下一个结果中得到证明。

定理 7.1。 $\mathcal{N}_{\mathscr{C}}$ 是一个安全标记网，其生成的可达图与箱(E)相同。

证明:
我们首先证明，定理 3.8 的表述中所作的假设对于以下对象成立：(i) $\mathcal{N}$ =箱(E) =(P, T, Fl, M_{init})；(ii) $\mathscr{P}_p = \mathscr{P}$和 $\mathscr{P}'_p = \mathscr{P}_p \cap \Pi_{\mathscr{C}}$，对于所有 $\mathscr{P} \in$ distpl(E) 以及 $p \in \mathscr{P}$；(iii) $P' = \Pi_{\mathscr{C}}$；以及 (iv) $\mathcal{N}' = \mathcal{N}_{\mathscr{C}}$。

(No注： $\mathcal{N}$、 $\mathscr{P}_p$、 $\mathcal{N}'$、P' 和 $\mathscr{P}'_p$ 的作用与定理 3.8 中相同 .)

根据命题 5.3(1)， $\mathcal{N}$是一个安全标记网。此外，根据命题 5.3(6)，distpl(E)是一个两两不相交的分布式库所集合，覆盖了P中的库所。因此，对于每个$p \in P$， $\mathscr{P}_p$ 都有明确定义。由于$M_{init} \in$ distpl(E)，且 distpl(E)中的分布式库所两两不相交， $\mathcal{N}$的初始标识符合要求（见命题 5.3(5,6)）。

接下来我们注意到， $\mathcal{N}' = (\mathrm{P}', \mathrm{T}, \mathrm{Fl}', \mathrm{M}'_{init}) = (\mathrm{P}', \mathrm{T}, \mathrm{Fl}|_{T\times P'} \cup \mathrm{P}' \times \mathrm{T}, \mathrm{M}_{init} \cap \mathrm{P}')$ 是一个标记网。事实上，由于每个库所 $\mathrm{p} \in \mathrm{P}'$ 都具有 π_{clq} 的形式，且 $\mathrm{clq} \neq \varnothing$，因此 p 至少有一个输入变迁或至少有一个输出变迁。此外，每个变迁 $\mathrm{t} \in \mathrm{T}$ 都至少有一个前置库所，因为 t^{out} 是 CG_E 中的一个顶点，因此至少存在一个 $\mathrm{clq} \in \mathscr{C}$，使得 $\mathrm{t}^{out} \in$clq。因此， $\mathcal{N}'$ 是一个标记网。

接下来我们注意到，方程式(4a)成立，因为对于每个 $p \in P$ 和 $t \in {}^\bullet(\mathscr{P}_p)^\bullet$，都有如下结论：

- $t \in {}^\bullet(\mathscr{P}_p)$ 意味着存在 $\pi_Z \in \mathscr{P}'_p$，使得 $t^{in} \in Z$。- $t \in (\mathscr{P}_p)^\bullet$ 意味着存在 $\pi_Z \in \mathscr{P}'_p$，使得 $t^{out} \in Z$。

确实，假设 $t \in {}^\bullet(\mathscr{P}_p)$。那么存在 $clq \in \mathscr{C}$，使得$t^{in} \in clq$。根据命题 5.3(6) 和命题 6.4，存在 $\mathscr{P} \in$ distpl(E)，使得 $\pi_{clq} \in \mathscr{P}$。显然，$t \in {}^\bullet\mathscr{P}$。因此，根据命题 5.3(8)，$\mathscr{P}_p = \mathscr{P}$。（若 $t \in (\mathscr{P}_p)^\bullet$，则可用类似的方法处理。）

为证明公式（4b），假设 $t \in {}^\bullet(\mathscr{P}_p)$ 且 $u \in (\mathscr{P}_p)^\bullet$。那么，根据命题 6.4，在 CG_E 中，t^{in} 与 u^{out} 之间存在一条边。因此，存在 $clq \in \mathscr{C}$，使得 t^{in}，$u^{out} \in clq$。随后，我们按照上述方法证明 $\pi_{clq} \in \mathscr{P}'_p$。（我们以类似的方式，利用形式为$t^{out}$的顶点之间边的覆盖来证明公式（4c）。）

总而言之，根据目前为止的证明，定理 3.8(1,2)成立。此外，由命题 5.3(3,5,6)可知，定理 3.8(3)成立。因此，$RG_{\mathcal{N}} \cong RG_{\mathcal{N}'}$。

现在我们来讨论基本过程的两种优化。

首先，我们注意到，优化 I 不会改变结论，因为上述证明并未用到形式为t^{in} 的顶点之间的边覆盖。

为证明优化II的正确性，设 $\mathrm{P}'' = \{\mathrm{p} \in \mathrm{P}' \mid \mathrm{p}^\bullet \neq \varnothing\}$ 且 $\mathcal{N}'' = (\mathrm{P}'', \mathrm{T}, \mathrm{Fl}'', \mathrm{M}''_{init}) = (\mathrm{P}'', \mathrm{T}, \mathrm{Fl}|_{T\times P''} \cup \mathrm{P}'' \times \mathrm{T}, \mathrm{M}_{init} \cap \mathrm{P}'')$。显然， $\mathcal{N}''$是一个标记网。我们将证明 $\mathcal{N}''$ 和 $\mathcal{N}'$ 的可达图同构，即 $\mathrm{RG}_{\mathcal{N}'} \cong \mathrm{RG}_{\mathcal{N}''}$。

我们首先注意到 $\mathrm{fseq}(\mathcal{N}') = \mathrm{fseq}(\mathcal{N}'')$，这由 $P' \setminus P'' \subseteq \Pi^x_E$ 以及Π^x_E 是没有输出变迁的库所这一事实得出（见命题 5.3(2)）。因此，由于$RG_{\mathcal{N}'}$ 和$RG_{\mathcal{N}''}$都是确定性的，只需证明：对于所有 $\sigma, \sigma' \in \mathrm{fseq}(\mathcal{N}') = \mathrm{fseq}(\mathcal{N}'')$，发射序列$\sigma$和 σ'在当且仅当 σ和 σ'在 $\mathcal{N}''$中导向相同的标识。我们首先注意到，从左到右的蕴含成立，因为$P'' \subseteq P'$。

为证明从右到左的蕴含，假设 σ 和 σ' 在 $\mathcal{N}''$中导向相同的标识，而在 $\mathcal{N}'$中分别导向两个不同的标识M 和M'。

不失一般性，假设存在 $\mathrm{p} \in \mathrm{M} \setminus \mathrm{M}' \subseteq \Pi_E^x$。由已证明的 $\mathrm{fseq}(\mathcal{N}) = \mathrm{fseq}(\mathcal{N}')$ 以及命题 5.3(3)，存在 $\sigma'' = \mathrm{t}_1 \ldots \mathrm{t}_k$，使得 $\mathrm{M}_{init}[\sigma' \circ \sigma''\rangle_{\mathcal{N}} \Pi_E^x$。令 M'' 满足 $\mathrm{M}_{init}[\sigma'\rangle_{\mathcal{N}} \mathrm{M}''$。由于 $\mathrm{p} \notin \mathrm{M}'$ 且 $\mathrm{P}' \subseteq \mathrm{P}$，我们也有 $\mathrm{p} \notin \mathrm{M}''$。因此，由于 $\mathrm{p} \in \Pi_E^x$，我们得到 $^\bullet\mathrm{p} \cap \{\mathrm{t1}, \ldots, \mathrm{t}_k\} \neq \varnothing$。

接着我们注意到，由 $\mathrm{fseq}(\mathcal{N}) = \mathrm{fseq}(\mathcal{N}') = \mathrm{fseq}(\mathcal{N}'')$，可得 $\sigma' \circ \sigma'' \in \mathrm{fseq}(\mathcal{N}'')$。此外，由于 σ 和 σ' 在 $\mathcal{N}''$ 中导向相同的标识，$\sigma \circ \sigma'' \in \mathrm{fseq}(\mathcal{N}'')$。因此 $\sigma \circ \sigma'' \in \mathrm{fseq}(\mathcal{N}')$。这一点与 $^\bullet p \cap \{t1, \ldots, t_k\} \neq \varnothing$ 以及 $p \in M$ 一起，违背了 $\mathcal{N}'$ 的安全性和 $p^\bullet = \varnothing$。

因此，我们可以得出结论：$RG_{\mathcal{N}} \cong RG_{\mathcal{N}'} \cong RG_{\mathcal{N}''}$。 □

请注意，PECCP 显然属于 NP，而且许多现有的 ECCP 算法都可以改编用于求解 PECCP，例如 [20]。此外，由于 ECCP 在一般图上是 NP 完全问题，且它是 PECCP 的特例，因此 PECCP 在一般图上也是 NP 完全问题。然而，对于从盒表达式导出的连接图——这是一类限制非常严格的图——许多 NP 完全问题都可以在多项式时间内求解。尽管如此，据我们所知，ECCP 或 PECCP 是否在余图上属于 NP 完全问题，仍是一个悬而未决的问题。还请注意，在对控制流进行建模时，并不要求结果最优，因此，能够计算出较小但不一定最小覆盖的快速启发式算法就足够了。事实上，简单的 ECC 覆盖方法先用单独的团覆盖每条边，再任意将其扩展为极大团，这已经避免了笛卡尔积构造所导致的指数级爆炸。

例 7.2。图 6(b) 显示了 (a ‖b)□(c ‖d) 的连接图。此外，图 6(c) 展示了应用第一次优化后的相同连接图。求解 PECC 的一种可能结果是得到以下覆盖：$\{\mathrm{a}^{out},\mathrm{c}^{out}\}$, $\{\mathrm{a}^{out},\mathrm{d}^{out}\}$, $\{\mathrm{b}^{out},\mathrm{c}^{out}\}$, $\{\mathrm{b}^{out},\mathrm{d}^{out}\}$, $\{\mathrm{a}^{in},\mathrm{c}^{in}\}$，以及 $\{\mathrm{b}^{in}, \mathrm{d}^{in}\}$。因此，得到的解是图 4(c) 中的 $\mathcal{N}_1$，其中已移除 r_5 和 r_7。◇

例 7.3。图 6(a) 显示了 E =(a ‖b); ((c ‖d)□(e ‖f)) 的连接图（注意，box(E) 显示在图 5(a) 中）。图 6(b) 显示了将一些边标记为已覆盖后的相同连接图（优化 I）。求解 PECC 的一种可能结果是得到以下覆盖：$\{\mathrm{a}^{out}\}$, $\{\mathrm{b}^{out}\}$, $\{\mathrm{a}^{in}, \mathrm{c}^{out}, \mathrm{f}$ 输出$\}$, $\{\mathrm{a}^{in}, \mathrm{d}^{out}, \mathrm{f}$ 输出$\}$, $\{\mathrm{b}^{in}, \mathrm{c}^{out}, \mathrm{e}^{out}\}$, $\{\mathrm{b}^{in}, \mathrm{d}^{out}, \mathrm{e}^{out}\}$, $\{\mathrm{c}^{in}, \mathrm{e}^{in}\}$，以及$\{\mathrm{d}^{in}, \mathrm{f}^{in}\}$。因此，得到的解是图 5(b) 中的 $\mathcal{N}_3'$；经过优化II后，我们得到图 2(b) 中所示的$\mathcal{N}_1'$。◇

例 7.4. 对于盒表达式 $[a \,\|\, b * (c \,\|\, d);\, e * f \,\|\, g]$，注意到$\Gamma_{w \| z} = (w^{out} \uplus z^{out}, w^{in} \uplus z^{in}, \varnothing)$ 对所有动作 w和 z均成立，我们得到如下结果：

$$
\begin{aligned}
&\Gamma_{[a \| b * (c \| d);\, e * f \| g]} \\
&= (\Gamma^e_{a \| b},\ \Gamma^x_{f \| g},\ \Gamma^i_{a \| b} \uplus \Gamma^i_{(c \| d);\, e} \uplus \Gamma^i_{f \| g} \uplus ((\Gamma^x_{a \| b} - \Gamma^x_{(c \| d);\, e}) - ((\Gamma^e_{f \| g} - \Gamma^e_{(c \| d);\, e})))) \\
&= (a^{out} \uplus b^{out},\ f^{in} \uplus g^{in},\ \Gamma^x_{c \| d} - \Gamma^e_e \uplus ((a^{in} \uplus b^{in}) - \Gamma^x_e) - (((f^{out} \uplus g^{out}) - \Gamma^e_{c \| d})))) \\
&= (a^{out} \uplus b^{out},\ f^{in} \uplus g^{in},\ (c^{in} \uplus d^{in}) - e^{out} \uplus ((a^{in} \uplus b^{in}) - e^{in}) - (((f^{out} \uplus g^{out}) - (c^{out} \uplus d^{out}))))) .
\end{aligned}
$$

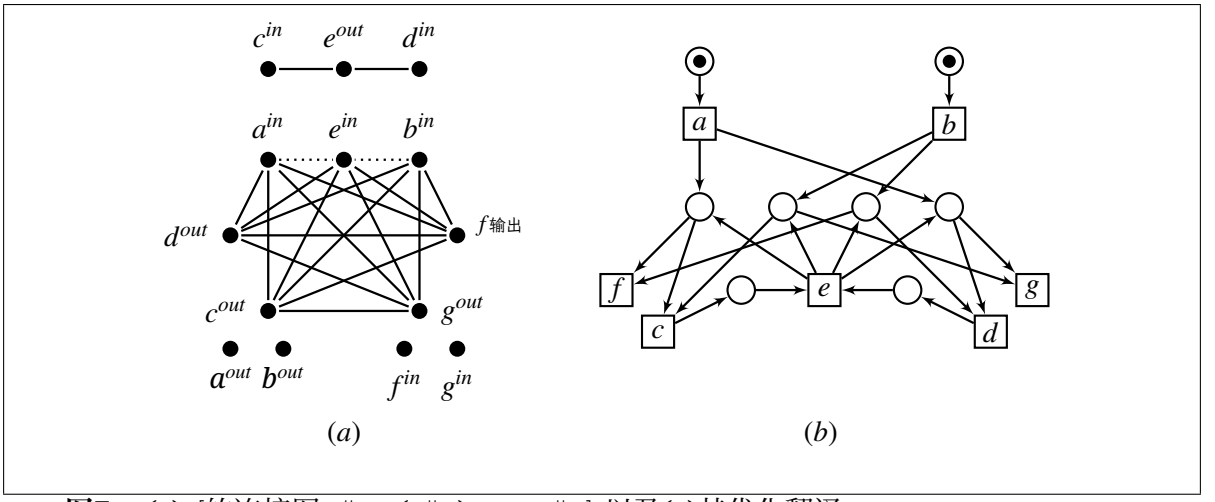

图7： (*a*) [的连接图$a \parallel b * (c \parallel d)$； $e * f \parallel g]$, 以及(*b*)其优化翻译。

因此我们有：

$$CG_{[a \parallel b*(c \parallel d);\, e*f \parallel g]}$$
$$= \quad a^{out} \uplus b^{out} \uplus f^{in} \uplus g^{in} \uplus ((c^{in} \uplus d^{in}) - e^{out}) \uplus ((a^{in} \uplus b^{in}) - e^{in}) - (((f^{out} \uplus g^{out}) - (c^{out} \uplus d^{out})))$$

如图7(*a*)所示，虚线表示已被覆盖的边。该余图有14个极大团，以下10个构成一个最小ECC：

$$\{a^{out}\} \quad \{b^{out}\} \quad \{c^{in}, e^{out}\} \quad \{a^{in}, e^{in}, f^{out}, c^{out}\} \quad \{a^{in}, e^{in}, d^{out}, g^{out}\}$$
$$\{f^{in}\} \quad \{g^{in}\} \quad \{d^{in}, e^{out}\} \quad \{b^{in}, e^{in}, g^{out}, c^{out}\} \quad \{b^{in}, e^{in}, d^{out}, f^{out}\}$$

然后，经过优化II，我们得到图7(*b*)中的网。 ◇

8. 结论

未来的工作中，我们计划将本文所得结果扩展到采用其他具有实际应用价值的算子的盒表达式。此外，我们还希望开发有效的算法，以识别一般安全标记网中的分布式库所。

参考文献

[1] Best E, Devillers RR, Koutny M. Petri网代数. 理论计算机科学专著. EATCS丛书. 施普林格，2001.

[2] Goltz U、Mycroft A。《CCS与Petri网的关系》。载于：Paredaens J（编），《自动机、语言与编程》，第11届专题讨论会，比利时安特卫普，1984年7月16日至20日，论文集，计算机科学讲义第172卷。施普林格，1984年，第196–208页。

[3] Khomenko V、Meyer R、Hüchting R。将π演算FCP转换为安全Petri网的多项式转换。《计算机科学中的逻辑方法》，2013年。**9**(3)。[4] Olderog E。CCSP的Petri网操作语义。载于：Rozenberg G（编），《佩特里网进展 1987》，涵盖第七届欧洲佩特里网应用与理论研讨会，英国牛津，1986年6月，计算机科学讲义第266卷。施普林格，1986年，第196–223页。[5] Best E、Devillers RR、Koutny M。《盒代数》 = Petri网 +进程表达式。《信息与计算》，2002年。*178*(1)：44–100。[6] Milner R。《通信系统演算》。施普林格，1980年。[7] Hoare C。《通信顺序进程》。普伦蒂斯-霍尔，1985年。[8] Valmari A。状态爆炸问题。载于：Reisig W、Rozenberg G（编），《佩特里网讲座 I：基本模型》，《佩特里网进展》，各卷基于在达格施图尔举行的佩特里网高级课程，1996年9月，计算机科学讲义第1491卷。施普林格，1996年，第429–528页。[9] Khomenko V、Koutny M、Yakovlev A。避免控制流Petri网模型中的指数级爆炸。载于：Bernardinello L、Petrucci L（编），《佩特里网与并发的应用与理论》—第43届国际会议，PETRI NETS 2022，挪威卑尔根，2022年6月19日至24日，论文集，计算机科学讲义第13288卷。施普林格，2022年，第261–277页。[10] Chan A、Sokolov D、Khomenko V、Lloyd D、Yakovlev A。突发自动机：使用突发模式规约进行速度无关综合的框架。《IEEE集成电路与系统计算机辅助设计汇刊》，2023年。**42**(5)：1560–1573。[11] Cortadella J、Kishinevsky M、Kondratyev A、Lavagno L、Yakovlev A。异步控制器与接口的逻辑综合。施普林格，2002年。[12] Best E、Devillers RR、Hall JG。盒演算：一种具有多标签通信的新型因果代数。载于：Rozenberg G（编），《佩特里网进展 1992》，DEMON项目，计算机科学讲义第609卷，第21–69页。施普林格，1992年。[13] van Glabbeek RJ、Goltz U。基于因果关系的模型中的动作精化。载于：de Bakker JW、de Roever WP、Rozenberg G（编），《分布式系统的逐步精化》，模型、形式化方法、正确性，REX研讨会，荷兰莫克，1989年5月29日至6月2日，论文集，计算机科学讲义第430卷。施普林格，1989年，第267–300页。[14] Khomenko V、Koutny M、Yakovlev A。精简Petri盒：控制流的紧凑Petri网模型。载于：Klin B、Lasota S、Muscholl A（编），第33届并发理论国际会议，CONCUR 2022，2022年9月12日至16日，波兰华沙，*LIPIcs*第243卷。达格施图尔城堡—莱布尼茨信息学中心，2022年，第8:1–8:16页。[15] Lerchs H。关于团与核。技术报告，计算机科学系，多伦多大学，1971年。[16] Badouel É、Bernardinello L、Darondeau P。基本网系统的综合问题是NP完全问题。《理论计算机科学》，1997年。**186**(1-2)：107–134。[17] Badouel É、Bernardinello L、Darondeau P。Petri网综合。《理论计算机科学丛书》。EATCS丛书。施普林格，2015年。[18] Pietkiewicz-Koutny M。使用最小区域综合ENI系统。载于：Sangiorgi D、de Simone R（编），CONCUR ’98：并发理论，第9届国际会议，法国尼斯，1998年9月8日至11日，论文集，计算机科学讲义第1466卷。施普林格，1998年，第565–580页。

[19] Khomenko V, Koutny M, Yakovlev A. 分布式库所与安全网约简. 收录于：Amparore EG, Mikulski L（编），《佩特里网与并发的应用与理论——第46届国际会议，PETRI NETS 2025》，法国巴黎，2025年6月22日至27日，会议论文集，第15714卷，计算机科学讲义。施普林格，2025年，第265–286页。

[20] Gramm J, Guo J, HüffnerF, Niedermeier R. 团覆盖的数据约简与精确算法. *ACM* 实验算法学杂志，2009年。**13**。