arXiv:2601.18745v1 [cs.LO] 26 Jan 2026

参数化程序的对称证明

Ruotong C
hengchengrt@cs.toronto.edu
多伦多大学 多伦多，加拿大

Azadeh Farzan
azadeh
@cs.toronto.edu 多伦多
大学 多伦多，加拿大

摘要

我们研究基于丰富的一类拓扑构建的无限状态参数化程序的安全性验证问题。我们引入一种新的证明系统，称为参数化证明空间，该系统利用此类程序中的底层对称性。这是一种局部的对称性概念，使证明系统能够在程序拓扑中重用针对同构邻域的证明论证。我们证明了该证明系统相对于一类全称量化不变式具有精巧的相对完备性结果。我们还研究了这些证明的算法构造问题。受模型论经典结果的启发，我们提出一种构造方法：在满足某些条件时，可以用一个无穷极限程序代替参数化程序族进行验证，并保证验证过程可靠且完备。此外，我们展示了如何构造这些证明，并在无需为证明或程序对底层拓扑进行公理化的情况下，针对这些程序检查证明。最后，我们给出使该算法成为判定过程的条件。

1 引言

本文聚焦于无限状态参数化程序的算法安全性验证问题。这类程序广泛出现于诸多场景，从设备驱动程序、文件系统，到分布式程序和高度并行的 GPU代码。经典地，参数化程序$\mathcal{P}$ 是一个无限程序族$\{P_n\}_{n\in\mathbb{N}_+}$，其中 P_n 表示包含 n 个线程的程序族成员。我们关注一类安全性性质，其违规可编码为违反某个（线程）局部断言。若一个参数化程序族中的每个成员都是安全的，则该参数化程序就是安全的。除程序数据域之外，这个新的无限维度给验证问题带来了新的挑战。该问题通常不可判定 [5], ，但由于这类程序用途广泛，拥有可靠的半判定过程十分理想。

验证参数化程序的技术挑战，也可以换个角度看作发现能够证明其安全性的全局归纳不变式。一般而言，这些归纳不变式是带有任意量词的语句，且量词交替——

国家 [42], 这使得即使检查候选不变式的归纳性也变得棘手，更不用说自动发现不变式了。在目标程序属于一类受限程序的情况下，人们已经在自动发现纯全称量化的不变式方面取得了一些成功 [21, 29], ，这类程序基于通过共享内存进行通信的复制的相同线程。

我们的目标是研究一类丰富的无限状态参数化程序，它们基于一类丰富的底层拓扑。例如，考虑环形拓扑上的线程，其中包含四个线程的实例如图1所示。每个圆形节点对应一个线程，该线程由模板实例化而来；每个矩形节点则对应一个共享变量（其取值来自无限数据域）。

这类示例不属于 [16, 21, 29] 的程序模型。除了程序族和数据域这两个无限维度之外，还有一个额外的因素——拓扑——需要进行形式化建模，并在验证过程中加以运用。

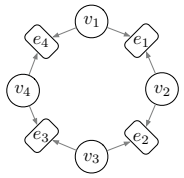


图 1：环

我们提出了一种新的证明系统，称为参数化证明空间，它可以针对不同的底层拓扑实例化为参数。作为参数。该工作的关键洞见是，利用一种基于底层拓扑族固有对称性的通用对称性概念，是定义这一证明系统的关键。值得注意的是，这种对称性捕获了证明系统从拓扑族中所需的全部信息，而无需对这些拓扑进行进一步的公理化。

参数化证明空间可以视为证明空间的推广，这些证明空间由 [21], 引入，仅适用于称为星形拓扑（见示例4.2）的固定拓扑族：一组相同的复制线程，它们共享数量有界的全局变量，并以此作为通信手段。证明空间是一组有效的霍尔三元组，在顺序组合、合取和对称性下封闭；也就是说，它是这一演绎系统的理论。在较高层次上，各项操作定义如下，其中 $\sigma/\sigma_1/\sigma_2$是程序命令序列：

- 顺序组合：如果 $\{P_1\}\ \sigma_1\ \{P_2\}$ 和 $\{P\prime_2\}\ \sigma_2\ \{P_3\}$ 是有效三元组，那么 $\{P_1\}\ \sigma_1;\sigma_2\ \{P_3\}$ 也是有效三元组，其中 $P_2 \vdash P\prime_2$ 和 $\vdash$ 是组合蕴涵[1]。
- 合取：如果 $\{P_1\}\ \sigma\{Q_1\}$ 和 $\{P_2\}\ \sigma\{Q_2\}$ 有效，那么 $\{P_1 \wedge P_2\}\ \sigma\ \{Q_1 \wedge Q_2\}$ 也有效。
- 对称性：如果 $\{P\}\ \sigma\{Q\}$ 有效，那么任何通过对三元组的三个分量中的线程标识符进行一致重命名而得到的三元组 $\{P\prime\}\ \sigma\prime\{Q\prime\}$ 也有效。

如果一个证明空间 H 能由有限个三元组生成，则称其为有限生成的。若对于规约 Pre/Post，H 证明运行集 R 正确，则对 R 中的每个运行 σ，霍尔三元组 $\{\text{Pre}\}\ \sigma\{\text{Post}\}$ 都属于 H；也就是说，可以从公理三元组出发，使用顺序组合、合取和对称规则推导出 $\{\text{Pre}\}\ \sigma\{\text{Post}\}$。

参数化证明空间保留合取规则和顺序规则，并提出一条新的参数化对称规则，以取代证明空间中的对称规则；对于星形拓扑以外的拓扑，该规则并不可靠。可以通过利用底层拓扑的同构来定义对称规则，从而弥补这一不可靠性。然而，这种对称性概念过于严格，无法赋予证明系统足够的推广能力。

[1]无需解释原子谓词，即可检查出现在 P'_2 中的合取项是否是出现在 P_2中的合取项的子集

回顾图1中的环形示例。假设每个共享变量（位于矩形节点上）都是一个布尔值 t，表示一个令牌。考虑一个命令 σ，它将一个真令牌从右向左传递：

```
assume t(r); t(r) := false; t(l) := true
```

其中，r和 l分别指右侧和左侧的矩形节点，而霍尔三元组 $\{\neg t(e_1)\}\ \sigma :\ v_3\ \{\neg t(e_1)\}$ 是有效的，因为执行 σ 的节点 v_3 不应影响 e_1处布尔值变量的值。自然可以预期，对称规则能够推出霍尔三元组 $\{\neg t(e_4)\}\ \sigma :\ v_3\ \{\neg t(e_4)\}$ 的有效性，该三元组出于同样的原因是有效的。然而，环上的同构不可能固定 v_3 并将 e_1 映射到 e_4。如果我们希望在大小为 5 的环中也得到相同类型的有效霍尔三元组，情况会变得更加棘手。不存在将大小为 4 的环中的节点映射到大小为 5 的环中的节点的同构。

我们的新对称规则采用了一种局部的对称性概念，该概念基于程序底层结构中的同构邻域。我们证明，新规则可靠地（定理 5.1）为一个新的邻域生成一个新的有效霍尔三元组，而该新邻域基于某个对称邻域。采用局部对称性十分重要，我们通过一个形式化的相对完备性定理对这一主张作出精确定义。直观地说，我们证明，只要能够使用某种全称量化不变量证明程序是安全的，就能够使用有限生成的参数化证明空间证明程序是正确的（定理 6.1）。

我们在 [7] 第6.1节将这些全称量化不变式定义为 Ashcroft 不变式的推广 [21, 29]；文献中已有的 Ashcroft 不变式则成为星形拓扑下这些不变式的一个特例。我们证明，如果存在广义 Ashcroft 不变式，就可以提取出有效的霍尔三元组（引理6.3 和6.4）。然而，棘手之处在于证明：有限多个这样的霍尔三元组足以生成一个参数化证明空间，从而证明整个参数化程序正确。这要求拓扑族满足一个与不变式的宽度 k（即全称量词的数量）相关的条件。直观地说，这意味着拓扑族（1）包含足够多的小实例，使得（2）在任意大小的实例中，大小大约为 k 的任意邻域都可以被其中一个小实例覆盖。条件（1）提供有限基，条件（2）则保证这个基足以满足要求（定理6.5）。直观地说，当拓扑族满足这一条件时，就可以通过对程序数据理论进行有限次查询，检查广义 Ashcroft 不变式是否适用于参数化程序；这些查询与参数化证明空间的有限基密切相关。

某些拓扑（如环）不满足这一条件。幸运的是，可以通过将缺失的子结构（例如环的片段）添加到族中，同时保持参数化程序的安全性，从而将其中大多数向下封闭。我们在第6.4节将这种构造形式化，从而扩大相对完备性结果的适用范围。

作为一种证明系统，证明空间的算法演绎需要论证该证明空间能够证明某个参数化程序族中的每个成员都是正确的。为了实现有效算法，必须处理参数化证明空间和参数化程序族中存在的多个无限维度。为了确保参数化证明空间可由算法有限表示，我们需要对非常一般的对称性概念施加额外条件。我们的方法有一个关键优势：虽然拓扑在参数化证明空间中起着核心作用，但我们无需对拓扑进行公理化。相反，我们使用局部对称等价关系的等价类，这些等价类可以通过基于拓扑中具体 节点（元组）的无量词公式来表示。

为了解决族中包含无穷多个成员所带来的问题，我们引入极限程序。极限程序是一个单独的无穷程序，它（1）以某种形式化意义模拟参数化族中所有成员的行为，并且（2）不包含可能改变安全性判定的额外行为。由于底层拓扑具有这样一种隐含性质，即参数化族中较小的实例可以嵌入到较大的实例中，因此这两个条件都可以得到满足。我们在第7.1 节提出了极限程序的一种选择，使其满足（1）和（2），并给出形式化嵌入性质的模型论条件；在这些条件下，极限存在、可以构造，并且具有有限表示。更具体地说，在某些条件下，我们可以构造一个无穷拓扑，将参数化族中所有成员的拓扑无冗余地嵌入其中；这样，相应的程序就能模拟族中所有成员的行为，同时不会引入额外行为。

一旦我们有了可有限表示的参数化证明空间和可有限表示的极限程序，就可以将以下操作性证明规则实现为验证算法：

$$\frac{\mathcal{L}(P_{\text{limit}}) \subseteq \mathcal{L}(\mathcal{H}),\ \mathcal{H} \text{ is correct by construction}}{\forall i \in \mathcal{I} : P_i \text{ is safe}} \qquad \text{(OR)}$$

其中，L(–) 用于同时表示极限程序的行为，以及由证明空间 H 证明正确的行为。

为了有效表示参数化证明空间和极限程序，我们对谓词自动机（见 [21]中引入）进行推广，使其能够针对所选的拓扑族进行参数化定义。随后我们证明，参数化谓词自动机保留了谓词自动机的闭包性质，因此可将规则 OR 中的包含性检查转化为对单个自动机的语言进行空性检查（定理 7.3）。

我们按照良结构转移系统中反向可达性的思路，给出一种用于该空性检查的算法。该算法依赖于良拟序 的存在，以保证终止。我们给出能够保证终止并由此得出空性判定过程的结构条件（定理 7.7）。当空性可判定时，操作性证明规则 OR 可以实现为一种用于验证参数化程序的细化式算法。虽然参数化证明空间不足以证明程序正确，但空性检查能够得出反例。这些反例是有限轨迹，可以使用标准验证方案进行验证。由此得到的具体霍尔三元组随后可以加入参数化证明空间的生成元集合中。这正是我们用具体生成元集合来形式化定义证明空间的主要原因。

最后，我们将关注任意拓扑上布尔参数化程序安全性验证的可判定性问题 [4]。首先，我们证明，对于任意拓扑族，这些程序的参数化证明空间都是完备的

（定理 6.8）。此外，如果底层拓扑满足算法可判定性的条件，我们的算法框架便能给出一个判定过程（推论 7.9）。对于这一特定类别的拓扑族上的参数化布尔程序而言，这是一个新的结果（见第 2 节和8节）。

2 一个可判定拓扑示例

为了说明我们的贡献，我们使用一个并行计算示例，介绍一类基于新拓扑族的参数化程序。这些拓扑允许通过参数化证明空间进行可判定证明，而先前的研究并未涵盖这些证明空间 [4, 21]。图2（a）展示了并行图像处理中可分离滤波器卷积的一个实例。在某些条件下，具有 $O(m^2)$ 个操作的滤波器可以在 $O(2m)$ 个步骤中作为两个独立滤波器的序列执行。考虑将 n^2 网格视为一幅灰度图像，目标是通过将每个 k^2 分区替换为其最大值来降低图像质量。由于操作max满足结合律，因此可以先对每一行进行卷积，再对每一列进行卷积。这个并行程序的底层拓扑是一个由图2（b）所示形式的树组成的森林。每棵树对应于在一个不同 $k \times k$ 方形上执行的计算。树中的每个方形节点代表一个数组单元，每个圆形节点运行图2（c）所示的代码模板。直观地说，底行的方形（从左到右）是一个 $k \times k$ 分区中的单元格，按行编号；中间行的方形是第一次卷积阶段后所得数组中的单元格，按列编号。根节点是输出数组中记录卷积结果的单元格。

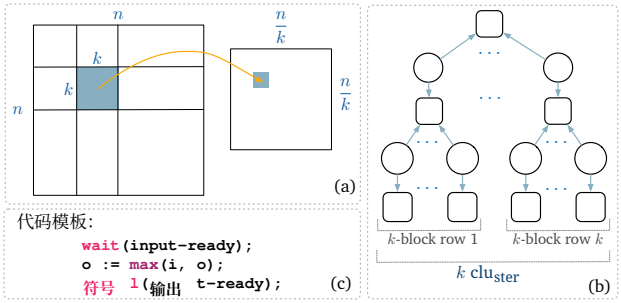


图 2：两阶段卷积（a、c）。拓扑（b）。

首先，极限程序存在于该程序族（带有参数 k 和 n）中，并满足定理7.4 的条件；它可以构造为 Fraïssé 极限（见命题7.1）。底层的（极限）拓扑由 ω 个（即可数无穷多个）深度为 5 的树副本组成，其中每棵树的分支因子为ω，取代 k。顺便指出，对于任意固定深度、任意分支的树族，该结论同样成立（证明概要见附录A ）。

此外，对于该程序的任意一元的参数化证明空间，借助 *Kruskal* 树定理，状态空间具有良拟序。这意味着，检查任意一元的参数化证明空间候选都是可判定的（见引理7.6）。

当数组数据有限时，例如，灰度图像中的灰度级数量有界，那么本文的结果最终得出了一种验证该程序的新判定过程：我们证明，存在一个通用的一元参数化证明空间，可以证明该程序正确（参见定理 6.8），且检查该证明是可判定的（参见推论7.9）。

3 背景

我们回顾模型论中的相关基本概念。一个词汇表由谓词符号和函数符号组成，并假设存在一个从符号到其元数的全局映射 ar。我们将等号视为逻辑符号。不含函数符号的词汇表称为关系型。一个 V-结构 $\mathbb{A}$ 被定义为一个底集，并配备对 V 中符号的解释。为简便起见，我们也使用 $\mathbb{A}$ 来指代 $\mathbb{A}$的底集。

词汇表 V[$\mathbb{A}$] 是通过为 $\mathbb{A}$中的每个元素添加一个新的常量符号，对词汇表 V 进行扩展得到的。当将 V-结构 $\mathbb{A}$ 视为 V[$\mathbb{A}$]-结构时，每个 $a \in \mathbb{A}$ 对应的常量符号都解释为 a。

设 $\mathbb{A}$，$\mathbb{B}$ 是两个 V-结构。嵌入 *embedding* $f\colon \mathbb{A} \hookrightarrow \mathbb{B}$ 是一个保持并反映所有谓词和函数的单射；若这样的嵌入存在，我们称 $\mathbb{A}$ 是 $\mathbb{B}$的嵌入子结- 构。如果 $\mathbb{A}$ 的底集是 $\mathbb{B}$底集的子集，且包含映射是从 $\mathbb{A}$ 到 $\mathbb{B}$的嵌入，我们就称 $\mathbb{A}$ 是 $\mathbb{B}$ 的子结构。同构 *isomorphism* 是满射嵌入。 $\mathbb{A}$ 的自同构*automorphism* 是从 $\mathbb{A}$ 到其自身的同构。

由 $\mathbb{A}$ 底集的子集 A 生成的子结构是包含 A 且在应用 V 中的函数后仍封闭的最小子结构。若任意两个有限生成子结构之间的同构都能扩展为一个完整的自同构，我们就称 $\mathbb{A}$ 是齐次的。

4 参数化并发程序

本节的目标是正式定义参数化程序，即定义在底层拓扑族之上的程序族。在第 4.1 节中，我们将拓扑正式定义为（参数化）逻辑结构，然后形式化程序的语法和语义。接着，在第4.2 节中，我们将正式定义参数化程序，并为其定义一种对称性概念，随后将在第4.2节中用它来定义参数化证明空间。

4.1 拓扑上的程序

我们的程序记作 $\mathbf{P} = P, [\![-]\!]$，由语法部分 P 和语义部分 $[\![-]\!]$ 组成。直观而言，语法是在程序（通信）拓扑上定义的；程序拓扑正式定义为结构 $\mathbb{T}$，它建立在某个词汇表 $\mathrm{V}_{\mathrm{Node}}$之上，并包含足够的信息，以精确定义程序的句法运行集合；也就是说，这些运行是程序执行的一系列程序命令，不附带任何含义。为方便起见，我们将这一结构称为拓扑。反过来，$[\![-]\!]$ 定义在数据域 $\mathbb{D}$之上；数据域是建立在某个词汇表 $\mathrm{V}_{\mathrm{Data}}$之上的结构，用来刻画程序变量中存储并由程序命令操作的数据类型。

形式上，拓扑是一个（可能是）无限的逻辑结构 $\mathbb{T}$，其底层元素是网络[2]中的节点 。我们将

[2]如果$\mathcal{V}_{\mathrm{Node}}$ 中的所有函数符号的元数都是1，那么就底层集合而言， $\mathbb{T}$ 的子结构类确实是 $\mathbb{T}$上的一个拓扑，从而使其成为一个拓扑空间。尽管如此，我们还是选择将通信拓扑建模为逻辑结构，因为这样更容易将其整合到证明系统中。

由一组节点 V 生成的子结构称为 V 的邻域，记作 $N(V)$。我们有时会滥用记号，将节点元组 $\vec{v}$ 视为一个集合，因此令 $N(\vec{v})$ 表示元组 $\vec{v}$ 中这些节点构成的集合的邻域。

为了基于拓扑构建完整的程序语法，我们进一步为每个节点分配一个变量[3]和一个进程 $\mathbb{T}$。然后，我们假设有一个有限的命令集 Σ。每个命令 $\sigma \in \Sigma$ 通过以一个节点为索引，成为一个程序命令 v，该节点来自 $\mathbb{T}$：$\langle \sigma : v \rangle$。

程序的全局状态是每个运行中进程的每个变量赋值和程序位置。对于后者，我们固定一个（有限）程序位置集 $\mathbf{L}$，并为初始位置设置一个不同的位置 l_{init}。然后，我们写作 $\text{GlobalState}(\mathbb{T} = \mathbb{T} \to \times\, \mathbb{T} \to) : (\mathbf{L})$ $\mathbb{D}()$。为了形式化 $[\![-]\!]$，我们还为每个节点 v 定义 $\overset{v}{N} = N\, v \to \mathbb{D}$ $\text{LocalData}() : ()$。

一个拓扑上的并发程序 $\mathbb{T}$ 形式上是一个二元组 $\mathbf{P} = \langle P, [\![-]\!] \rangle$，其中 $P : \mathbb{T} \to 2^{\Sigma}$ 给出程序的句法运行，而 $[\![-]\!] : \prod_{\langle \sigma : v \rangle \in \Sigma \times \mathbb{T}} \text{LocalData}(v) \times \text{LocalData}(v)$ 描述（带索引的）命令的语义；对于参数化程序（定义 4.2）中的这些命令，我们还施加了进一步的条件。

程序的句法运行。 该程序实际上由一组节点构成，每个节点并行运行一些代码。这里，我们形式化地定义句法运行集，直观来说，它是来自每个节点的（句法）运行交错而成的集合。

令函数 $\text{src,tgt} : \Sigma \to \mathbf{L}$ 将一条命令映射到其源位置和目标位置。直观来说，这些函数区分控制流图中同一命令的不同出现位置。结合映射 $P : \mathbb{T} \to 2^{\Sigma}$，它们会在位置映射 $\mathbb{T} \to \mathbf{L}$ 之间诱导出一个转移关系，直观来说，这相当于一个乘积控制流图：对于任意一对位置映射 λ, λ'，我们定义 $\lambda \overset{\sigma : v}{\longrightarrow} \lambda'$，如果 σ 属于 $P(v)$，位置 $\lambda(v)$ 是 σ 的源位置，并且 $\lambda' = \lambda \oplus \{v \mapsto \text{tgt}(\sigma)\}$。

然后，我们可以定义 $\mathbf{P}$ 的句法运行集，更具体地说，是基于其句法分量 P 定义的句法运行集；它由所有运行 $\sigma_1 : a_1 \cdots \sigma_n : a_n \in (\Sigma \times \mathbb{T})^*$ 构成，其中存在位置映射 $\lambda_0, \ldots, \lambda_n : \mathbb{T} \to \mathbf{L}$ 满足 $\lambda_0 \equiv l_{\text{init}}$，且对于所有 i 都有 $\lambda_{i-1} \overset{\sigma_i : a_i}{\longrightarrow} \lambda_i$。该句法运行集记作 $\mathbf{L}(P)$。

语义。 语义 $[\![-]\!] : \prod_{\langle \sigma,\ v \rangle \in \Sigma \times \mathbb{T}} \text{LocalState}(v) \times \text{LocalState}(v)$ 描述带索引命令 $\sigma : v$ 如何改变邻域 $N(v)$ 上的赋值。对于任何更大的邻域 $V \supset N(v)$，将 $\sigma : v$ 的语义扩展到 V 上的赋值，并规定该命令不会访问（即读取或写入）$N(v)$ 以外的任何内容；形式上，令 $\text{Global}_V[\![\sigma : v]\!]$ 为

$$\left\{ (\mu, \mu') \in (V \to \mathbb{D})^2 : (\mu, \mu')|_{N(v)} \in [\![\sigma : v]\!],\ \mu|_{\overline{N(v)}} = \mu'|_{\overline{N(v)}} \right\},$$

其中，记法 $(\mu,\ \mu')|_X$ 是 $(\mu|_X,\ \mu'|_X)$ 的简写。通过关系复合，我们将 $\text{Global}_V[\![-]\!]$ 从单个命令提升到任意运行 $\rho \in (\Sigma \times \mathbb{T})^*$，其节点均包含在 V 中。

转移关系。 程序 $\mathbf{P} = P, [\![-]\!]$ 在拓扑 $\mathbb{T}$ 上的语法和语义会在全局状态 $\lambda, \mu \in \text{GlobalState}(\mathbb{T})$ 上定义一个转移关系：对于任意带索引命令 $\sigma : v$，若 $\lambda \overset{\sigma : v}{\longrightarrow} \lambda'$ 且 $(\mu, \mu') \in \text{Global}_{\mathbb{T}}[\![\sigma : v]\!]$，则称 $\lambda, \mu \xrightarrow{\sigma\, v} : \lambda', \mu'$。若对于某个 $\sigma : v$，有 $\lambda, \mu \xrightarrow{\sigma\, v} : \lambda', \mu'$，则称 $\lambda, \mu \to \lambda', \mu'$。

[3]为了编码多个局部变量，该变量可以具有结构类型。

安全性规约。 本文关注一类安全性规约，其中，违反安全性可通过某个线程中一个（局部的）断言被违反来见证。为了对该规约进行形式化编码，我们假设程序位置集 $\mathbf{L}$ 中有一个不同于其他位置的**错误**位置，记为 l_{err}。程序的**错误运行**集合记为 $\mathbf{L}_{\text{err}}(P)$，它是句法运行集合 $\mathbf{L}(P)$ 的一个子集，其中每个运行都以一个位置映射结束，且该映射的值域包含 l_{err}。

运行 $\rho \in \mathbf{L}(P)$ 是不可行的，当且仅当 $\text{Global}_{\mathbb{T}}[\![\rho]\!]$ 不包含任何一对 $(\mu,\ \mu')$；也就是说，从任何初始状态出发，都无法到达错误位置上的状态。我们称程序 P，$[\![-]\!]$ 是安全的，当且仅当每个错误运行都不可行；等价地，对于值域包含 l_{err} 的任意位置映射 $\lambda_{\text{init}} \equiv l_{\text{init}}$ 和 λ_{err}，以及任意赋值 μ 和 μ'，关系 $\lambda_{\text{init}},\ \mu \to^* \lambda_{\text{err}},\ \mu'$ 均不成立。

4.2 一族拓扑上的参数化程序

在本文的语境中，**参数化程序** 是一个程序族，其语法和语义在可形式化定义的意义上具有对称性。这种对称性概念通过局部同构及其拉回定义。

设 $\mathbb{T}$ 和 $\mathbb{T}'$ 是定义在相同词汇表上的两个结构。给定元组 $\vec{a} \in \mathbb{T}^d$ 和 $\vec{b} \in \mathbb{T}'^d$，同构 $\beta : N(\vec{a}) \to N(\vec{b})$ 称为**局部的**，如果 $\beta(\vec{a}) = \vec{b}$（β 逐分量适用）。从 $N(\vec{a})$ 到 $N(b)$ 的局部同构 $\vec{\beta}$，可由节点元组 $\vec{a}$ 和 b 这一对唯一确定：

命题 4.1. 设 $\mathbb{T}$ 和 $\mathbb{T}'$ 是具有相同词汇表的两个结构。对于任意 $\vec{a} \in \mathbb{T}^d$ 和 $\vec{b} \in \mathbb{T}'^d$，从 $N(\vec{a})$ 到 $N(\vec{b})$、将 $\vec{a}$ 映射到 $\vec{b}$ 的同构至多有一个。

因此，简而言之，我们称之为从 $\vec{a}$ 到 b 的局部同构，它在 $\mathbb{T}$ 上的元组之间诱导出一个（等价）关系：

定义 4.1（局部对称性 $\simeq$）。设 $\mathbb{T}$ 和 $\mathbb{T}'$ 是具有相同词汇表的两个结构。对于 $\mathbb{T}$ 上的任意节点元组 $\vec{a}$ 以及 $\mathbb{T}'$ 上的 $\vec{b}$，如果 $\vec{a}$ 和 b 长度相同，且存在从 $\vec{a}$ 到 $\vec{b}$ 的局部同构，则称它们**局部对称**，记作 $\vec{a} \simeq b$。

例 4.1。图 1 的环形拓扑可以用词汇表 $\mathbf{V}_{\text{Node}} = \{d, p\}, \{l, r\}$ 建模，其中 d 和 p 是一元谓词符号，用于指示节点是数据（矩形）还是进程（圆形）；l 和 r 是一元函数符号，表示进程节点的左侧和右侧节点；对于数据节点 e，我们令 $l(e) = r(e) = e$。我们有 $(e_1, v_3, e_1) \simeq (e_4, v_3, e_4)$，但 $(e_1, v_3, e_1) \not\simeq (e_2, v_3, e_2)$。

首先，我们观察到，$\simeq$-等价元组满足相同的无量词公式集合，反之亦然：

命题 4.2。 设 $\mathbb{T}$ 和 $\mathbb{T}'$ 是词汇表 $\mathbf{V}$ 上的结构。对于任意 $\vec{a} \in \mathbb{T}^d$ 和 $\vec{b} \in \mathbb{T}'^d$，我们有 $\vec{a} \simeq \vec{b}$，当且仅当对于每个无量词 V- 公式 $\varphi[x_1, \ldots, x_d]$，$\mathbb{T} \models \varphi(\vec{a}) \Leftrightarrow \mathbb{T}' \models \varphi(\vec{b})$。

此外，如果对于所有维度 d，$\mathbb{T}^d/\simeq$ 都是有限的，那么每个等价类都可以由唯一的无量词公式定义：

命题 4.3。 对于任意 $d \in \mathbb{N}_+$，如果商集 $\mathbb{T}^d/\simeq$ 是有限的，那么 $\simeq$ 的每个等价类都可由无量词 V-公式定义。

局部同构 $\beta\colon N(\vec{a}) \to N(b)$ 通过 $\mu \mapsto \mu \circ \beta$ 在赋值之间诱导出一个映射 $\beta^*\colon (N(\vec{b}) \to \mathbb{D}) \to (N(\vec{a}) \to \mathbb{D})$。我们以自然的方式将 β^* 提升到赋值元组的集合。

直观地说，参数化程序是定义在具有相同词汇表的结构上的程序族，其中任意一对局部对称节点运行相同的代码，并且各个命令具有相同的语义。形式化地说：

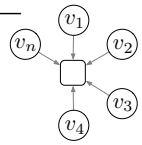
定义 4.2（参数化程序）。给定用于拓扑的词汇表 $\mathrm{V_{Node}}$，以及基于词汇表 $\mathrm{V_{Data}}$的数据域 $\mathbb{D}$，参数化程序是满足以下条件的程序族：

- 该程序族中的每个程序 P，$[\![-]\!]$ 都定义在词汇表 $\mathrm{V_{节点}}$上的某个拓扑、数据域 $\mathbb{D}$以及相同的命令集 Σ 和位置集 $\mathbf{L}$之上。
- 对于该程序族中定义在 $\mathbb{T}$ 上的任意一对程序 P，$[\![-]\!]$ 和定义在 $\mathbb{T}'$ 上的 P'，$[\![-]\!]'$，对于任意 $a \in \mathbb{T}$ 和 $a' \in \mathbb{T}'$，若 $a \simeq a'$ 经由 β 运行，即 $a' = \beta(a)$，则 $P(a) = P'(a')$，且 $[\![\sigma\colon a]\!] = \beta^*[\![\sigma\colon a']\!]'$ 对每个 $\sigma \in \Sigma$ 均成立。

注意，第二个要点也适用于一对相同的程序。我们称程序 程序**P** 是局部对称的，如果{**P**} 满足定义4.2。显然，参数化程序的每个成员都是局部对称的。

如果一个参数化程序族中的每个程序都是安全的，我们就称该参数化程序是安全的。我们将参数化程序写作指标集 $\{\mathbf{P}_i\}_{i\in I}$，其中 I 可以是无限集。

例 4.2。我们来看看参数化程序最经典的实例——线程不可区分的共享内存并发程序 [21],是定义 4.2的一个特例。右侧展示了该拓扑。



我们使用常量符号 g 表示全局（共享）变量，它们存放在中间的方形节点上。n 圆形节点上有 n 线程。形式上，词汇表为 $\mathrm{V_{Node}} = \varnothing$，$\{\mathsf{g}\}$，与含有 n 个线程的程序相对应的 $\mathrm{V_{Node}}$-结构 $\mathbb{T}_n$ 的论域为 $\{-1,\ 0,\ \ldots\ldots,\ n-1\}$，其中 -1 是 g 的解释，自然数表示 n 线程。

我们对参数化程序的定义有意设置得非常一般。值得注意的是，参数化程序族不必包含某种通信拓扑类型的每个实例所对应的程序。例如，给定由例 $\mathcal{P}$ 4.2描述的参数化程序 $\mathbb{T}_{42}$，如果我们从$\mathcal{P}$中移除对应于 $\mathbb{T}_{42}$ 的程序，所得结果仍然是一个参数化程序。在第 6.4 节中，我们将利用这种一般性所带来的灵活性。

在证明特定结果需要时，我们会对程序族施加限制。例如，相对完备性定理要求对拓扑施加限制（定理 6.5 和命题6.7）。

5 参数化证明空间

我们将在第1 节概述参数化证明空间是如何根据程序底层拓扑所定义的局部对称性概念而参数化的证明空间。我们在定义局部对称性时，形式化地定义了任意结构的局部对称性：4.1。要定义对称规则，第一步是确定适用于我们的参数化证明空间的底层结构，以便利用其局部对称性。

回想一下，我们的程序参数化族形如 $\{P_i,\ [\![-]\!]_i\}_{i\in I}$，其中每个 P_i，$[\![-]\!]_i$ 都基于拓扑 $\mathbb{T}_i$。然而，我们希望证明空间能够为该程序族中的每个成员提供正确性证明。因此，我们取一个（通常是无限的）结构 $\mathbb{T}$，其共同词汇表为 $\mathrm{V_{Node}}$，并且每个 $\mathbb{T}_i$ 都是它的嵌入子结构。我们称 $\mathbb{T}$ 为通用结构。

当词汇表中没有常量符号[4], 时，这样的结构可以直接定义在每个 $\mathbb{T}_i$的论域的不交并上；构造通用结构的另一种方法见第 7.1 节。我们将每个 $\mathbb{T}_i$ 对应到$\mathbb{T}$ 中与 $\mathbb{T}_i$同构的一个子结构。

回顾一下，我们通过局部对称关系来定义参数化程序 $\simeq$。在第 5.2 节中，我们基于同一个局部对称关系，在通用结构 $\mathbb{T}$ 上介绍新的对称规则。在此之前，我们先定义通用结构中的有效霍尔三元组。

5.1 通用结构中的霍尔三元组

带索引命令的语义可以扩展到通用结构 $\mathbb{T}$ 中的元素：对于任意的 σ：$v \in \Sigma \times \mathbb{T}$，对于任意的 i 和 $u \in \mathbb{T}_i$，若存在从 v 到 u 的局部同构 β

$$[\![\sigma : v]\!] = \begin{cases} \varnothing & \text{if no such } i \text{ and } u \text{ exist} \\ \beta^*[\![\sigma : u]\!]_i & \text{otherwise} \end{cases}$$

通用结构的定义允许 $\mathbb{T}$ 中存在不对应于任何 $\mathbb{T}_i$ 中任何节点的节点，这就是第一种情况可能发生的原因。

我们考虑前置条件和后置条件均为关于以节点为索引的变量的断言的霍尔三元组。在第 4.1 节中，我们不失一般性地假设每个节点 u 上都有唯一的变量；我们用 $\mu(u)$ 表示该变量，其中 μ 是元数为 1 的形式符号。利用 $\mathrm{V_{Data}}$-公式 $\phi[x_1,\ldots,x_n]$，我们构造一组公式，记为断言，其形式为$\phi[\mu(u_1)\ldots,\mu(u_n)]$ 其中 u_1，...，u_n 是 $\mathbb{T}$ 中的节点。对于断言 φ，令 $\mathrm{nodes}(\varphi)$ 表示 φ 中出现的节点集合。

给定 $\mathbb{D}$ 和 $\mathbb{T}$，断言 φ 的真值仅取决于 μ 的解释，该解释由赋值 $\mu\ \ U \to$ 给出：$\mathbb{D}$ U $\mathbb{T}$ 其中 是包含节点(φ)的子集。我们按标准方式定义可满足关系 $\mu \models \varphi$。

霍尔三元组的形式为

$$\left\{\phi[\overrightarrow{\mu(u)}]\right\}\ \langle\sigma_1 : v_1\rangle \cdots \langle\sigma_n : v_n\rangle\ \left\{\psi[\overrightarrow{\mu(w)}]\right\},$$

其中 $\longrightarrow\mu(u)$ 是元组 $\mu(u_1)\ldots,\mu(u_n)$ 的简写。令 V为邻域 $N(\vec{u}\vec{v}\vec{w})$。上述霍尔三元组在以下情况下有效：对于任意赋值 μ,μ'，若满足 $(\mu,\mu') \in \mathrm{Global}_V[\![\sigma_1 : v_1 \cdots \sigma_n\colon v_n]\!]$，则若 $\mu \models \phi[\overrightarrow{u}\ \mu(\,)],$，那么 $\mu' \models \psi[\overrightarrow{w}\ \mu(\,)]$。

5.2 新对称规则

直观而言，我们的对称规则表示：如果某个节点元组对应一个有效的霍尔三元组，那么对于任何与该元组局部对称的其他元组，同一个霍尔三元组也有效。形式化地说：

[4]将拓扑建模为逻辑结构时，总是可以用解释为常值函数的一元函数符号替换常量符号。

$$\text{S-PARAM}\quad \frac{\left\{\phi[\overrightarrow{\mu(u)}]\right\}\ \langle\sigma_1:v_1\rangle\cdots\langle\sigma_n:v_n\rangle\ \left\{\psi[\overrightarrow{\mu(w)}]\right\} \qquad \begin{array}{c}|\vec{u}|=|\vec{u}'|\\ |\vec{w}|=|\vec{w}'|\end{array}}{\left\{\phi[\overrightarrow{\mu(u')}]\right\}\ \langle\sigma_1:v_1'\rangle\cdots\langle\sigma_n:v_n'\rangle\ \left\{\psi[\overrightarrow{\mu(w')}]\right\} \qquad \vec{u}\vec{v}\vec{w}\simeq\vec{u}'\vec{v}'\vec{w}'}$$

例5.1（例4.1的延续）。我们将所有大小的环的不交并 $\mathbb{T}_{\text{rings}}$ 用作通用结构。对于第1节中的令牌传递示例，通过 S-param，我们可以从 $\{\neg t(e_1)\}\ \sigma\colon v_3\ \{\neg t(e_1)\}$ 推导出任意霍尔三元组 $\{\neg t(e)\}\ \{\sigma\colon v\}\ \{\neg t(e')\}$，只要 $\mathbb{T}_{\text{rings}}\models\chi(e,v,e')$，其中 $\chi(e,v,e') := (e=e'\wedge d(e)\wedge p(v)\wedge l(v)\neq e\wedge r(v)\neq e)$，它定义了 (e_1,v_3,e_1) 的 $\simeq$- 等价类（见命题 4.2）。特别地，可以推导出霍尔三元组 $\{\neg t(e_4)\}\ \{\sigma\colon v_3\}\ \{\neg t(e_4)\}$。

定理 5.1（S-param 的可靠性）。对于来自 $\mathbb{T}$ 的任意元组 $\vec{u}$、$\vec{v}$、$\vec{w}$、$\vec{u}'$、$\vec{v}'$ 和 $\vec{w}'$，规则 S-param 是可靠的。

5.3 参数化证明空间及其指称

我们终于可以正式定义参数化证明空间了。

定义5.1（参数化证明空间）。$\mathbb{T}$-证明空间是一个由霍尔三元组构成的集合 H，且在序列、合取和 S-param 这些规则下封闭。若 H 中的所有霍尔三元组均有效，则称 H有效。参数化证明空间是任意一个 $\mathbb{T}$-证明空间。

从算法上说，我们关注的是有限生成的参数化证明空间[5]。

定义5.2（有限生成）。$\mathbb{T}$-证明空间 H 由一组 H 霍尔三元组生成，是指 H 是包含 H 的最小 $\mathbb{T}$-证明空间。集合 H 称为 H 的一个基。若一个 $\mathbb{T}$-证明空间存在一个有限生成基，则称其为有限生成。

证明空间是一种证明，因为它能够为一组行为的正确性提供形式论证。

定义5.3（参数化证明空间的语言）。参数化证明空间 H 的语言 记为 L(H)，是满足 $\{\top\}\ \rho\ \{\bot\}\in$ H 的单词 $\rho\in(\Sigma\times\mathbb{T})^*$ 的集合。参数化证明空间语言 是以这种方式得到的语言。

直接观察可知，任何证明空间语言都具有如下意义下的局部对称性：将 $\simeq$ 的定义从 $\mathbb{T}$ 上的元组提升到 $\Sigma\times\mathbb{T}$ 上的单词，规定 $\sigma_1\colon v_1\cdots\sigma_n\colon v_n\simeq\sigma_1'\colon v_1'\cdots\sigma_{n'}'\colon v_{n'}'$ 当且仅当 $(\sigma_1,\ldots,\sigma_n)=(\sigma_1',\ldots,\sigma_{n'}')$ 且 $(v_1,\ldots,v_n)\simeq(v_1',\ldots,v_{n'}')$ 时，有

定理 5.2。证明空间语言在等价关系 $\simeq$ 下封闭。

由于我们已将每个 $\mathbb{T}_i$ 与 $\mathbb{T}$的某个子结构对应起来，参数化族中某个程序的语言 $\mathrm{L}(P_i)$ 也可以视为定义在字母表 $\Sigma\times\mathbb{T}$ 上。随后，借助语言包含性检查，证明空间为证明参数化程序的安全性提供了工具。

[5]在 [21], 出于算法方面的考虑，基中的霍尔三元组还受到其他句法限制。为明确划分关注点，我们在此取消这些限制；验证算法将在第 7 节讨论。

定理 5.3（可靠安全性证明规则）。若存在通用结构 $\mathbb{T}$ 和有效的 $\mathbb{T}$-证明空间 H，使得 $\bigcup_{i\in\mathcal{I}}\mathrm{L}\ \ \mathrm{err}(P_i)\subset\mathrm{L(H)}$，则参数化程序 $\mathrm{P}=\{P_i,[\![-]\!]_i\}_{i\in\mathcal{I}}$ 是安全的。

对于一个 $\mathbb{T}$-证明空间$\mathcal{H}$ 要成为一项有效的证明，它必须能够有限表示，并且检查$\bigcup_{i\in\mathcal{I}}\mathcal{L}\ \ \mathrm{err}(P_i)\subset\mathcal{L}(\mathcal{H})$ 必须是可判定的。我们将在第7节讨论这两个问题。

6 参数化证明空间的完备性

本节研究参数化证明空间作为证明系统的表达能力。具体而言，本节的大部分内容致力于建立一个相对完备性结果，该结果针对基于一族全称量化归纳不变式的安全性证明，适用于任意数据域。此外，我们还将在第6.5 节给出仅含布尔数据域的程序的完备性结果。

Ashcroft 不变式最初是在固定线程程序的背景下定义的，对于这类程序，作为一种证明方法，它们是相对完备的（在 Cook 意义下）[7]。它们也可以用作星形拓扑上参数化程序的全称量化归纳不变式 [21, 29],，但它们并不完备。在 [21], 中，作者证明了有限生成的（非参数化）证明空间相对于 Ashcroft 不变式是完备的。为了将参数化证明空间的表达能力与 Ashcroft 不变式联系起来，我们在第6.1 节和第6.4节探讨两个问题：（1）对于我们更丰富的一类参数化程序，Ashcroft 不变式的自然推广是什么？（2）[21] 中确立的证明空间与 Ashcroft 不变式之间的对应关系，是否仍适用于参数化证明空间及这些不变式?

我们在第6.1节提出了 Ashcroft 不变式的一种推广。然而，要在推广后的 Ashcroft 不变式与有限生成的参数化证明空间之间建立对应关系，还需要对拓扑族施加一个额外条件，以赋予它们有限性。我们先给出下文相对完备性结果的非正式表述，并在本节中逐步加以精确化，最终得到定理6.5。

定理 6.1（相对完备性，非正式表述）。设 P 是一个以通用结构 $\mathbb{T}$ 为基础、定义在有限拓扑 T 上的参数化程序。对于任意能够证明 P 安全的广义 Ashcroft 不变式 Φ，如果拓扑族 T 满足某个条件 C（与 Φ 中量词的数量有关），那么就存在一个有限生成的 $\mathbb{T}$证明空间，能够证明 P 安全。

条件 C 直观地说，这意味着在$\mathcal{T}$ 中存在有限多个小型拓扑，使得$\mathcal{T}$ 中的任意拓扑都可以分解为这些拓扑的同构副本之并。此外，该分解的粒度与公式 Φ的宽度有关；也就是说，与其中（全称）量词的数量有关。当 C 对于一个参数化程序本身不成立时，可以通过一种简单的构造来修改这些拓扑。该构造的行为类似于向下闭包计算：它会将所有缺失的小型拓扑加入该族，同时保留原程序的安全性或不安全性。在第6.4节中，我们介绍这一构造，并论证对于广泛的一类参数化程序，该条件都成立。

6.1 广义 Ashcroft 不变式

广义 *Ashcroft* 断言定义为词汇表 $\mathrm{V}_{\textsc{Ashcroft}}$ 上第一阶公式的一个片段，该词汇表包含三种类型：节点、数据和位置。我们先定义词汇表，然后说明界定该片段的语法限制。最后，给定一个参数化程序，广义 *Ashcroft* 不变式是一个归纳的广义 Ashcroft 断言，可用于证明该程序安全。下文中，除非可能引起歧义，我们通常省略“广义”一词。

词汇表 $\mathrm{V}_{\mathrm{Node}}$（类似地，$\mathrm{V}_{\mathrm{Data}}$）只有一个类型 Node（分别为 Data）：$\mathrm{V}_{\mathrm{Node}}$ 中的 a k元关系符号的类型为 Node^k，而 a k元函数符号的类型为 $\mathrm{Node}^k \to \mathrm{Node}$。此外，我们定义词汇表 $\mathrm{V}_{\mathrm{Loc}}$，它由一个类型 Loc 以及 $\mathbf{L}$ 中每个元素对应的一个 Loc 类型常量符号组成。

词汇表 $\mathrm{V}_{\textsc{Ashcroft}}$ 是 $\mathrm{V}_{\mathrm{Node}}$、$\mathrm{V}_{\mathrm{Data}}$ 和 $\mathrm{V}_{\mathrm{Loc}}$ 的不交并，此外还包含两个额外的函数符号 $\mu : \mathrm{Node} \to \mathrm{Data}$ 和 $\mathrm{l} : \mathrm{Node} \to \mathrm{Loc}$。约定 u、v 和 w 表示 $\mathrm{V}_{\mathrm{Node}}$-类型变量，而 x、y 和 z 表示 $\mathrm{V}_{\mathrm{Data}}$-类型变量。

QF-Ashcroft 片段由关于 $\mathrm{V}_{\textsc{Ashcroft}}$ 的一阶公式构成，其中不对 Node 进行量化，且每个自由变量的类型都是 Node。宽度为 *width* k 的 Ashcroft 断言（记作 Ashcroft$[k]$）是如下形式的句子

$$\forall u_1, \ldots, u_k.\, \varphi[u_1, \ldots, u_k],$$

其中 φ 是一个 QF-Ashcroft 公式。Ashcroft 断言集合 Ashcroft 是所有 $k \in \mathbb{N}$对应的 Ashcroft$[k]$ 的并集。

回顾一下，我们始终固定 $\mathrm{V}_{\mathrm{Data}}$-结构 $\mathbb{D}$ 以及控制位置集 $\mathbf{L}$，并将其视为 $\mathrm{V}_{\mathrm{Loc}}$-结构 $\mathbb{L}$。因此，对于特定拓扑，即 $\mathrm{V}_{\mathrm{Node}}$-结构 $\mathbb{T}$（不一定是通用结构），Ashcroft 断言 Φ 中的符号均已解释，除了 l 和 μ；因此，Φ 的真值由一对 λ，μ 给出，其中 $\lambda\colon \mathbb{T} \to \mathbb{L}$ 是位置映射，$\mu\colon \mathbb{T} \to \mathbb{D}$ 是赋值。给定 $\mathbb{T}$，我们以标准方式定义可满足关系 $\lambda,\ \mu \models \Phi$以及在 $\mathbb{T}$下的蕴含关系，记作 $\Phi \models \Phi'$。尽管这两个关系都以 $\mathbb{T}$为参数，但由于 $\mathbb{T}$ 可由上下文明确得知，我们在记号中省略它。

定义 6.1（广义 Ashcroft 不变式）。对于程序$\mathbf{P} = P$，$[\![-]\!]$，广义 *Ashcroft* 不变式是一个公式 $\Phi \in$Ashcroft，使得

Initialization $\forall u.\, \mathrm{l}(u) = l_{\mathrm{init}} \models \Phi$.
Continuation For any state $\langle \lambda, \mu \rangle$, if $\langle \lambda, \mu \rangle \models \Phi$ and $\langle \lambda, \mu \rangle \to \langle \lambda', \mu' \rangle$, then $\langle \lambda', \mu' \rangle \models \Phi$.
Safety $\Phi \models \forall u.\, \mathrm{l}(u) \neq l_{\mathrm{err}}$.

如果 Φ 是某个参数化程序 $\{\mathbf{P}_i\}_{i \in I}$ 的 Ashcroft 不变式（即它是所有$\mathbf{P}_i$的 Ashcroft 不变式），我们就称 Φ 为该参数化程序的 Ashcroft 不变式。

根据定义，参数化程序存在 Ashcroft 不变式意味着该程序是安全的。

6.2 由 Ashcroft 不变式得到霍尔三元组

要证明定理 6.1，主要有两个挑战。我们需要：(1) 从广义 Ashcroft 不变式中获得 有效的 霍尔三元组；(2) 证明有限个此类三元组能够生成一个足以证明程序安全的参数化证明空间。

关于(1)，可以观察到 QF-Ashcroft 与 断言之间在语法上具有相似性（见第5.1节），但两者之间存在两个差距：

（i）断言中的公式是闭合的，其节点来自通用结构；而 QF-Ashcroft 中的公式使用 Node 类的自由变量。（ii）断言只对数据进行推理，而在 QF-Ashcroft 中，可以通过函数符号 l 访问每个节点的控制位置。

本节通过引入扩展 Hoare 三元组作为中间对象，弥合上述差距，以应对前两个挑战。给定一个关于拓扑 $\mathbf{T} = \{\mathbb{T}_i\}_{i \in I}$、适用于 $\mathbf{P} = \{\mathbf{P}_i\}_{i \in I}$ 的 Ashcroft 不变式 Φ，我们为每个有限拓扑 $\mathbb{T}_i$ 关联一个由有效扩展 Hoare 三元组组成的有限集 $H_\Phi^{\mathbb{T}_i}$。（引理 6.3）随后，我们选取有限拓扑的有限子集 $\{\mathbb{T}_j\}_{j \in \mathbf{J}}$（第 6.3 节），并将扩展 Hoare 三元组$\bigcup_{j \in \mathbf{J}} H_\Phi^{\mathbb{T}_j}$转换为标准 Hoare 三元组（引理 6.4），从而构造一个有限生成的证明空间，足以证明 P 的安全性。

第 5.1 节中的霍尔三元组使用如下形式的断言：$\phi[\overrightarrow{v}\ \mu(\,)]$,，其中 $\phi[\vec{x}]$是一个 $\mathrm{V}_{\mathrm{Data}}$-公式。扩展断言的形式为$\psi[\overrightarrow{v}\ \mu(\,) \longrightarrow \mathrm{l}(v)]$,，其中 l 被解释为位置映射，且 $\psi[\vec{x}, \vec{l}]$是一个 $\mathrm{V}_{\textsc{Ashcroft}}$-公式。基于这些扩展断言，我们定义了扩展 Hoare 三元组和扩展参数化证明空间。为避免冗长，我们省略其形式化定义。直观来看，不难理解为何参数化证明空间的三条推导规则仍然可靠：添加符号 l，可以看作是为每个维护程序计数器的节点 a引入一个辅助变量 $\mathrm{l}(a)$。

扩展断言与 QF-Ashcroft 公式之间存在一种简单的联系。直观而言，简化 $\varphi[\vec{a}]$中的所有已解释项和子公式，恰好会得到一个等价的扩展断言。形式化地说：

命题 6.2。给定一个 $\mathrm{V}_{\mathrm{Node}}$-结构 $\mathbb{T}$，对于任意 QF-Ashcroft 公式 $\varphi[u_1, \ldots, u_n]$和 $\vec{a} \in \mathbb{T}^n$，存在唯一的（模等价意义下）扩展断言 $\phi[\mu(b),\ \mathrm{l}(b)]$,其中 b是来自$N(\vec{a})$的一个元组，使得对于任意从 $\vec{a}$ 出发的局部同构 β，扩展断言 $\phi[\overrightarrow{\beta\ b}\ \mu(\,(\,)) \longrightarrow \mathrm{l}(\beta(b))]$等价于 $\varphi[\beta(\,\vec{a})]$。

稍微滥用一下记号，我们将该扩展断言简称为 $\varphi[\vec{a}]$。那么，在有限拓扑 $\mathbb{T}_i \subset \mathbb{T}$上，宽度为 k的 Ashcroft 断言 Φ：$= \forall \vec{u}.\, \varphi[\vec{u}]$可以转化为具有相同模型集合的扩展断言，即 $\bigwedge_{\vec{a} \in \mathbb{T}_i^k} \varphi[\vec{a}]$。接下来，我们可以将 Φ 与一组扩展 Hoare 三元组 $H_\Phi^{\mathbb{T}_i}$相关联，定义如下：对于所有带索引命令 σ：$a_0 \in \Sigma \times \mathbb{T}_i$、元组 $\vec{a} \in \mathbb{T}_i^k$和节点 $b \in \mathbb{T}_i$,

$$\left\{\bigwedge_{a \in \mathbb{T}_i} \mathrm{l}(a) = l_{\mathrm{init}}\right\} \ \epsilon\ \{\varphi[\vec{a}]\} \qquad (\textsc{Init})$$

$$\left\{\bigwedge_{\vec{a} \in \mathbb{T}_i^k} \varphi[\vec{a}]\right\}\ \langle \sigma : a_0 \rangle\ \{\varphi[\vec{a}]\} \qquad (\textsc{Cont})$$

$$\left\{\bigwedge_{\vec{a} \in \mathbb{T}_i^k} \varphi[\vec{a}]\right\} \ \epsilon\ \{\mathrm{l}(b) \neq l_{\mathrm{err}}\} \qquad (\textsc{Safe})$$

只要 Ashcroft 断言 Φ 是 Ashcroft 不变式，它们就是有效的：

引理 6.3。 设P 是定义在有限拓扑 $\mathbb{T}_i \subset \mathcal{T}$上的程序。对于任意 Ashcroft 断言 Φ， $H_\Phi^{\mathbb{T}_i}$中的扩展 Hoare 三元组在 Φ 是P的 Ashcroft 不变式时都是有效的。

给定一个参数化程序的 Ashcroft 不变式，该引理提供了一种方法，可以从有限拓扑上的程序实例中提取有效的扩展 Hoare 三元组。然而，它并不保证有限个这样的扩展 Hoare 三元组就足以生成一个能够证明整个参数化程序正确的 $\mathbb{T}$-证明空间。我们将在下一节讨论这一问题。

最后，我们注意到，扩展 Hoare 三元组可以转换回标准 Hoare 三元组。

引理 6.4。 如果存在一个有限生成的扩展 $\mathbb{T}$-证明空间，能够证明 P 安全，那么就存在一个有限生成的标准 $\mathbb{T}$-证明空间，能够证明 P 安全。

请注意，本文正文中并未正式说明扩展 $\mathbb{T}$-证明空间证明 $\mathcal{P}$ 安全的含义；该定义在附录 B.3中本引理的证明里形式化给出。

6.3 保证有限基

假设 $\mathbb{T}_j$ 是一个有限拓扑。让我们考虑这样一个问题：如何利用 $H_\Phi^{\mathbb{T}_j}$（通过将不变式 Φ 实例化到 $\mathbb{T}_j$上所得到的霍尔三元组集合），来帮助证明族中另一个程序$\mathbf{P}_i$（$i \neq j$，且 $\mathbb{T}_i$ 可能是无限的）的安全性？假设根据引理 **6.3**，我们已知 P_j 具有以下形式的扩展 Hoare 三元组（Cont）。如果存在同构 β：$\mathbb{T}_j \to \mathbb{T}_i'$，使得 $\mathbb{T}_i'$是 $\mathbb{T}_i$的子结构，且包含 a_0 和 $\vec{a}$，那么由 $H_\Phi^{\mathbb{T}_j}$ 得到的有效霍尔三元组

$$\left\{\wedge_{\vec{a}\in\mathbb{T}_j^k} \varphi[\beta^{-1}(\vec{a})]\right\}\ \langle \sigma : \beta^{-1}(a_0)\rangle\ \{\varphi[\beta^{-1}(\vec{a})]\} \tag{1}$$

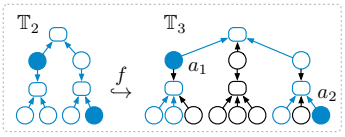
通过对称规则 S-Param 推广为三元组

$$\left\{\wedge_{\vec{a}\in\mathbb{T}_i'^k} \varphi[\vec{a}]\right\}\ \langle \sigma : a_0\rangle\ \{\varphi[\vec{a}]\} \tag{2}$$

这意味着对于 $\mathbf{P}_i$，形式为（Cont）的霍尔三元组是有效的。由（2）从（1）推导出 同构 β，是由拓扑族的性质保证的。下面的定义将 秩 设为 $k+1$（k为 Φ的宽度），刻画了拓扑族的一项精确性质，从而保证对于任意 $\mathbb{T}_i \in \mathcal{S}$、$\mathbb{T}_j \in \mathcal{T}$，以及所有三种形式的扩展 Hoare 三元组，都可以完成这一过程：

定义 6.2。 设 $k \in \mathbb{N}_+$ 和 T 是共同词汇表上的一个拓扑族。T 的秩为 k 的基是 T 的一个子集 S，满足：对于任意 $\mathbb{T} \in$T 和 $\vec{a} \in \mathbb{T}^k$，都存在某个 $\mathbb{S} \in$S 以及嵌入 f: $\mathbb{S} \hookrightarrow \mathbb{T}$，使得 $\vec{a} \in f(\mathbb{S})^k$。

例 6.1。 考虑对图中拓扑的一种简化2，其中每个$\mathbb{T}_i$ 都是一棵树，且每个矩形节点都有 i不可区分的



可将圆形节点作为子节点。$\{\mathbb{T}_1, \mathbb{T}_2\}$ 是 $\{\mathbb{T}_i\}_{i\in\mathbb{N}_+}$ 的秩为 2 的基。图中展示了一个元组 $(a_1, a_2) \in \mathbb{T}_3^2$，以及一个嵌入 f: $\mathbb{T}_2 \hookrightarrow \mathbb{T}_3$，其像包含 a_1, a_2。

注 1。 并非每个拓扑族都存在有限基：若 T 是所有大小的环，那么对于任意 $k \in \mathbb{N}_+$，T 的唯一秩为 k 的基就是 T 本身，因为任何环都无法嵌入另一个环中。

现在，我们可以通过定义条件 6.1，将非正式的定理完整地形式化 C：$\mathcal{T}$ 有一个秩为 $k+1$ 的有限基，由有限拓扑构成，其中 k 是 Ashcroft 不变式的宽度。注意，k 越大，该条件就越严格。定理的完整表述如下：

定理 6.5（相对完备性）。 设 P $= \{\mathrm{P}_i\}_{i\in\mathcal{I}}$ 是拓扑族 T $= \{\mathbb{T}_i\}_{i\in\mathcal{I}}$ 上的一个参数化程序，其通用结构为 $\mathbb{T}$。若 P 存在一个宽度为 k 的 Ashcroft 不变式 Φ，且 T 有一个秩为 $k+1$、由有限拓扑构成的有限基，那么就存在一个有限生成的 $\mathbb{T}$-证明空间，可以证明 P 是安全的。

定理6.5的拼接拓扑6.4

正如备注 1中所讨论的，环之类的拓扑不存在任何秩的有限基。我们给出一种简单的构造，将参数化程序 $\mathcal{P}$ 扩展为一个程序，使得所得程序所在的拓扑具有秩 k的有限基。我们的构造核心是子程序这一概念，它们是将程序限制在其拓扑的某个子结构上得到的：

定义 6.3 (子程序)。**给定一个定义在命令集 和拓扑 上的程序** $\mathbf{P} = P, [\![-]\!]$，对于 $\mathbb{T}$ 的任意子结构 $\mathbb{S}$，定义在 $\mathbb{S}$ 上的 子程序 是程序 $\mathbf{P}|_{\mathbb{S}} = P|_{\mathbb{S}}, [\![-]\!]|_{\Sigma\times\mathbb{S}}$。我们也称 P 是 $\mathbf{P}|_{\mathbb{S}}$ 的 超程序。

有趣的是，参数化程序 P 在子程序下的闭包 CL(P)（我们称之为向下闭包）在安全性方面可以替代原始参数化程序进行可靠且完备的 验证。如果我们只考虑通过参数化证明空间得出的可证性，这一结论仍然成立。

定理 6.6。 对于任意参数化程序 P 和 P′，如果 P $\subset$P′，并且 P′ 中的每个程序都是 P 中某个程序的子程序，那么 P 安全当且仅当 P′ 安全；此外，如果 $\mathbb{T}$ 是 P 和 P′ 的通用结构，那么任意 $\mathbb{T}$- 证明空间 H 证明 P 的安全性，当且仅当它证明 P′的安全性。

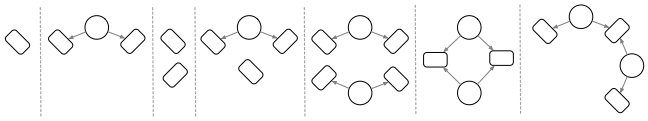
第二个结论成立，是因为通用结构中的霍尔三元组的有效性与所考察的句法运行无关，而且添加子程序不会改变句法运行的语言并集（参见定理5.3）。

为确立定理 6.5 所要求的条件，我们通过引入一个数 $k \in \mathbb{N}_+$作为参数来细化向下闭包的概念。对于任意拓扑 $\mathbb{T}$，令 $\mathrm{sub}_k(\mathbb{T})$ 表示由 k 个（不一定彼此不同的）节点生成的 $\mathbb{T}$ 的子结构集合。定义 CL(P k) 为从 P 得到的参数化程序：对每个 $i \in$I，将 P_i 在 $\mathrm{sub}_k(\mathbb{T}_i)$ 中各个拓扑上的所有子程序添加进去。下面的命题给出了 CL(P k) 的拓扑的一个秩为 k 的基，并提供了确保其有限性的条件。

命题 6.7。 对于任意 $k \in \mathbb{N}_+$ 以及共同词汇表 V 上的拓扑族 T，若对任意 $\mathbb{T} \in$T，都有 $\mathrm{sub}_k(\mathbb{T}) \subset$T，则$\bigcup_{\mathbb{T}\in\mathcal{T}} \mathrm{sub}_k(\mathbb{T})$ 是 T 的一个秩为 k 的基。此外，若 V 是有限的，且存在某个 $n \in \mathbb{N}$，使得每个结构的基数

在 $\bigcup_{\mathbb{T}\in\mathcal{T}} \mathrm{sub}_k(\mathbb{T})$ 中都不超过 n，则该基在同构意义下是有限的。

本文中出现的所有参数化程序 $\mathcal{P}$，其上述 CL($\mathcal{P}$ k) 的基（在同构意义下）都是有限的，且由有限拓扑构成。下图展示了环形拓扑（至少有两个进程节点）上秩为 2 的基，其中命题 6.7 中的界可以取为 6。



在这些情形下，相对完备性定理可简化为如下结论：对于任意 $k \in \mathbb{N}$，若 CL(P $k+1$) 存在宽度为 k 的 Ashcroft 不变式，则对于任意用于 P 的通用结构 $\mathbb{T}$，CL(P $k+1$) 因而 P 都具有有限生成的 $\mathbb{T}$-证明空间。

6.5 参数化布尔程序的完备性

本节讨论参数化布尔程序的参数化证明空间的完备性。需要注意的是，如第 4.1 节所述，验证问题会转化为可覆盖性 问题：我们证明所有到达错误位置的程序轨迹均不可行。众所周知，在星形拓扑下，参数化布尔程序的这一问题是可判定的（这是 Petri 网上的可覆盖性问题可判定性的推论 [31]）；并且在 [21, 第 6 节。3], 中提到（并在之后的 [20, 定理 2]中证明），（非参数化）证明空间对于此类参数化布尔程序的验证是完备的。因此，一个自然的问题是：参数化证明空间是否也具有这一性质?

在我们更一般的设定中，参数化布尔程序是一种参数化程序，其中每个节点都有有界数量的变量，且这些变量全都是布尔值；每条带索引命令只能访问（即读取和写入）有界数量的变量。

我们为范围广泛的参数化布尔程序建立了类似的完备性结果。据我们所知，这也使我们首次证明了此特定类别程序的验证问题是可判定的。

形式化地说，我们的参数化布尔程序满足以下条件：(1) 数据域 $\mathbb{D}$ 是固定长度为 n 的位向量，$\mathrm{V}_{\mathrm{Data}}$ 中每个 2^n 值都有一个常量符号；(2) 每条带索引命令的（局部）语义都是确定性的，并且读取和写入有限个变量；(3) 所有变量均初始化为 0。[6]

要求变量初始化且满足数据确定性只是为了方便，并不失一般性：任何未初始化且具有数据非确定性的布尔值程序，都可以轻松转换为一个所有变量均初始化为 0 的布尔值程序，其中任何数据非确定性都被控制流非确定性所取代。

[6]尽管我们对安全性的形式化定义关注的是从任意赋值开始的程序运行，但可以纳入这种初始化设定，正如我们在附录 B.3. 的证明中所展示的那样。

证明论证围绕为给定的命令集合（可用于任何程序）构造一个极大有限生成的参数化证明空间展开。形式化地说：

定理 6.8（参数化布尔程序的完备性）。任何具有通用结构 $\mathbb{T}$ 的安全参数化布尔程序 P，都有一个 $\mathbb{T}$ 证明空间，可以证明其安全性；该证明空间仅取决于 $\mathbb{T}$、命令集及其语义。此外，若 $\mathbb{T}^2/\simeq$ 有限，则该证明空间是有限生成的。

证明思路是利用如下形式的霍尔三元组

$$\left\{\bigwedge_{i=1}^{k} \mathfrak{u}(c_i) = x_i\right\}\ \langle \sigma : a\rangle\ \{\mathfrak{u}(b) = x_0\},$$

其中 x_0，……，$x_k \in \mathbb{D}$，且前置条件是最弱前置条件。当 $\mathbb{T}^2/\simeq$ 有限时，只需考虑有限多个节点对（a，b）。

请注意，定理 6.8 并未讨论有效性问题，我们将把它留到推论 7.9再讨论；届时我们将探讨参数化证明空间相关的算法问题：在关于 $\mathbb{T}$的某些条件下，定理 6.8 中的构造会产生一个判定过程，即可以计算最大参数化证明空间的基，并且证明检查是可判定的。

7 验证算法

定理5可被解释为使用参数化证明空间验证参数化程序的一条可靠证明规则。本节将探讨如何以算法方式实现这类证明规则。关键在于，为两种语言提供有限表示：证明空间的指称，以及参数化程序的句法轨迹集合，并设计一种算法来执行两者之间的包含性检查。一个自然的想法是使用数据自动机，因为参数化程序和参数化证明空间的行为都是命令带索引字母表上的词，而这些索引来自一个无界域。

首先，着眼于如何表示参数化程序的句法运行集合，我们需要克服有限表示无限并集这一挑战。为此，我们考虑无限并集收敛到某个“极限”的有趣情形。更具体地说，我们引入一个极限程序 P_∞，它是 P 的所有成员的超程序，使得在安全性验证中，P_∞ 可以可靠且完备地替代 P。

其次，对于程序和证明空间，我们都需要让自动机以某种方式感知底层拓扑。为此，我们提出对谓词自动机 [21]进行推广；与参数化证明空间类似，它以对称性的选择为参数，记作 $\mathbb{T}$-谓词自动机。实际上，[21] 的谓词自动机在 $\mathbb{T}$ 为 $\mathbb{N}$ 且词汇表为空时，便是其特例。随后我们证明，在满足有限可表示性的一些条件下，运行在 $\mathbb{T}$ 上的（极限）程序的所有错误运行，以及由有限 $\mathbb{T}$-证明空间的规范化基（定义7.4）生成的语言，都可由 $\mathbb{T}$-谓词自动机识别。我们论证，$\mathbb{T}$-谓词自动机具有与谓词自动机相同的基本闭包性质，因此包含性检查可归约为空性检查。我们给出了空性检查的算法解法，但由于$\mathbb{N}$-谓词自动机的空性 [21]不可判定，该解法无法构成判定过程。我们概述了空性检查可判定的条件。

7.1 极限程序

直观而言，对于参数化程序 $\mathbf{P} = \{\mathbf{P}_i\}_{i\in\mathbb{N}}$（即索引集 I 为 $\mathbb{N}$），如果对于每个 $m > n$，每个程序 $\mathbf{P}_n$ 都可以视为$\mathbf{P}_m$的子程序，那么就可以取程序序列$\mathbf{P}_1$,$\mathbf{P}_2$,...的“极限”$\mathbf{P}_\infty$。用$\mathbf{P}_\infty$进行验证是可靠的，因为极限程序包含每个$\mathbf{P}_i$的所有行为。要保证完备性，极限程序不应引入任何未在某个$\mathbf{P}_i$中出现的额外行为，这就要求对“极限”作出审慎的定义。

我们首先定义共同词汇表上的一类逻辑结构的极限。这是一个较弱的极限概念，但足以用于证明可靠性和完备性。

定义 7.1（逻辑结构的极限）。设 $\{\mathbb{T}_i\}_{i\in\mathcal{I}}$ 是共同词汇表 V 上的一类逻辑结构。V 上的逻辑结构 $\mathbb{T}_\infty$ 称为 $\{\mathbb{T}_i\}_{i\in\mathcal{I}}$ 的极限，如果每个结构 $\mathbb{T}_i$ 都能嵌入 $\mathbb{T}$，且 $\mathbb{T}$ 的每个有限生成子结构都能嵌入某个 $\mathbb{T}_i$。

例 7.1。对于例 4.2中的星形拓扑，极限可以看作一颗有无限多个分支的星；表示全局变量的节点是星的中心，并且有无限多个线程节点。这正是用来验证的程序[21]，其中参数化族被替换为该极限。另一个例子是第 2节已经讨论过的极限。

我们注意到，对于任意给定的结构类，极限不一定存在：极限结构首先必须是一个通用结构（第 5.1 节），但例如，环形拓扑的任何通用结构都必然包含每个环的一个不相交副本；因此，某些有限生成子结构（如两个不相交的环）无法嵌入该类中的任何结构。

模型论中成熟的 *Fraïssé* 极限 概念（例如，参见 [28, 第 6.1] 节）提供了一种系统地构造极限的方法：

命题 7.1 (Fraïssé 极限)。设 $\mathcal{T}$ 是共同词汇表上的有限逻辑结构类，且 $\mathsf{CL}(\mathcal{T})$ 是 $\mathcal{T}$ 中成员的子结构类。若 $\mathsf{CL}(\mathcal{T})$ 是一个可数的 Fraïssé 类，那么它的 Fraïssé 极限就是定义 $\mathcal{T}$7.1 所述意义下的极限。

给定一个参数化程序 $\mathbf{P} = \{P_i, [\![-]\!]_i\}_{i\in\mathcal{I}}$，其定义在拓扑族 $\mathbf{T} = \{\mathbb{T}_i\}_{i\in\mathcal{I}}$ 上，T 的任意极限 $\mathbb{T}_\infty$ 都会导出其上的一个极限程序 P_∞，$[\![-]\!]_\infty$ $\mathbb{T}_\infty$。直观而言，极限程序会识别不同拓扑中的节点。正如预期，其语法和语义通过局部同构来定义：对于 $\mathbb{T}_\infty$ 中的任意节点 v，以及某个 $\mathbb{T}_i$ 中的节点 u，若存在从 u 到 v 的局部同构 β，则设 $P_\infty(v) = P_i(u)$，并令 $[\![\sigma\colon v]\!]_\infty = \beta^*[\![\sigma\colon u]\!]_i$，对所有 $\sigma \in \Sigma$ 均成立。其定义良好性很容易验证。为了在计算中访问 P_∞ 和$[\![-]\!]_\infty$，我们假设存在一个可计算函数，它将 $\mathbb{T}_\infty$ 中的每个节点映射到 T 中一个局部对称节点。

注意，极限结构 $\mathbb{T}_\infty$ 上的命令语义定义方式与通用结构上的相同，不同之处在于，$\mathbb{T}_\infty$ 中的每个节点都与某个 $\mathbb{T}_i$中的某个节点局部对称。对于一个通用结构，其中 $\mathbb{T}_\infty$ 可被识别为一个子结构（例如，$\mathbb{T}_\infty$ 本身），我们证明其可靠性和完备性：

定理 7.2（极限程序的可靠性与完备性）。对于任意参数化程序 P 及其拓扑 $\mathbb{T}_\infty$ 上的极限程序 $\mathbf{P}_\infty$，P 安全当且仅当 $\mathbf{P}_\infty$ 安全。此外，如果 $\mathbb{T}$ 是 P 的一个通用结构，且 $\mathbb{T}_\infty$ 可被识别为一个子结构，那么任意 $\mathbb{T}$-证明空间 H 证明 P 的安全性，当且仅当它证明$\mathbf{P}_\infty$的安全性。

后一个结论成立的直观原因是，语言 $\mathrm{L}_{\mathrm{err}}(P_\infty)$ 和 $\bigcup_{i\in\mathcal{I}} \mathrm{L}_{\mathrm{err}}(P_i)$ 具有相同的 $\simeq$-闭包。

7.2 参数化谓词自动机

设 V 是一个词汇表，$\mathbb{T}$ 是一个 V-结构。我们仍将 $\mathbb{T}$ 中的元素称为节点。为了在这些节点上进行计算，我们作出两个基本假设：节点和 V 中的符号都可以有限表示（即可以编码为比特串），并且这些符号的解释是可计算的。

一个 $\mathbb{T}$-**谓词自动机**（PA）可识别无限字母表 $\Sigma\times\mathbb{T}$上的语言，其中 Σ 是一个有限集。一个 $\mathbb{T}$-PA 配备了一个与 V[$\mathbb{T}$]不相交的有限关系词汇表 Q，其状态记作 $\mathrm{F}^{\mathbb{T}}_{\mathrm{atom}}(\mathrm{Q})$，是形如 $q(a_1,\ldots,a_{\mathrm{ar}(q)})$（$q\in\mathrm{Q}$，$a_1,\ldots,a_{\mathrm{ar}(q)}\in\mathbb{T}$）的（未解释的）原子公式。令 $\mathrm{F}_+(\mathrm{V},\mathrm{Q})$ 表示所有无量词（V ∪Q）公式的集合，其中每个 $q(t_1,\ldots,t_{\mathrm{ar}(q)})$（$q\in\mathrm{Q}$，$t_j$是项）的出现均为正。形式化地，我们如下定义 $\mathbb{T}$-谓词自动机：

定义 7.2($\mathbb{T}$-谓词自动机)。**</style}一个 -谓词自动机**(PA) 是一个五元组 $\mathrm{Q}\ \Sigma\times\mathbb{T},\ \delta,\ \varphi_{\mathrm{start}},\ F$，其中 Q 是有限关系词汇表，$\Sigma\times\mathbb{T}$ 是一个字母表，$\varphi_{\mathrm{start}}\in\mathrm{F}_+(\mathrm{V},\mathrm{Q})$ 是一个没有自由变量的初始公式，$F\subset\mathrm{Q}$ 是接受谓词符号的集合，并且 $\delta\colon\mathrm{Q}\times\Sigma\to\mathrm{F}_+(\mathrm{V},\mathrm{Q})$ 是一个句法转移函数，满足对于任意 $q\in\mathrm{Q}$ 和 $\sigma\in\Sigma$，$\delta(q,\sigma)$ 的自由变量均属于集合 $\{v_0,\ldots,v_{\mathrm{ar}(q)}\}$。

一个 $\mathbb{T}$-PA 的元数 是关系词汇表 Q 中谓词元数的最大值。如果元数为 1，$\mathbb{T}$-PA 就称为一元的。

回顾一下，$\mathrm{F}^{\mathbb{T}}_{\mathrm{atom}}(\mathrm{Q})$ 是 $\mathbb{T}$-PA 的状态。令 $\mathrm{F}^{\mathbb{T}}(\mathrm{Q})$ 表示这些状态的所有无否定布尔组合。那么，每个句法转移函数 $\delta\colon\mathrm{Q}\times\Sigma\to\mathrm{F}_+(\mathrm{V},\mathrm{Q})$ 都会诱导出一个语义转移函数 $\hat{}\delta\colon\mathrm{F}^{\mathbb{T}}_{\mathrm{atom}}(\mathrm{Q})\times(\Sigma\times\mathbb{T})\to\mathrm{F}^{\mathbb{T}}(\mathrm{Q})$，其定义如下：

$$\hat{\delta}(q(a_1,\ldots,a_{\mathrm{ar}(q)}),\langle\sigma:a_0\rangle)\equiv\delta(q,\sigma)[v_i\mapsto a_i : i=0,\ldots,\mathrm{ar}(q)]. \tag{TF}$$

定义7.2强调了无量词公式，因为使用这些公式可以在无需任何额外假设的情况下计算语义转移函数，并且足以满足我们表示错误运行和证明空间语言的目的：二者都关于 $\simeq$对称，并且 $\simeq$-等价类 $\mathbb{T}^d$ 可以用由 d 个变量构成的无量词公式集合表示（命题4.2）。

由 $\mathbb{T}$-*PA* 识别的语言。我们现在可以定义 $\mathbb{T}$-PA 的语言，并讨论由 $\mathbb{T}$-PA 识别的语言类的性质。

定义 7.3（配置）。设 $A=\mathrm{Q}\ \Sigma\times\mathbb{T},\ \delta,\ \varphi_{\mathrm{start}},\ F$ 是一个 $\mathbb{T}$-PA。A 的一个配置C 是 $\mathrm{F}^{\mathbb{T}}_{\mathrm{atom}}(\mathrm{Q})$ 的有限子集，并将其视为 C 中公式的合取。若 C 是 φ_{start} 的析取范式（DNF）中的一个立方项，则称 C 为初始；若对于所有 $q(a_1,\ldots,a_{\mathrm{ar}(q)})$，$q\in F$C，则称 C接受。

谓词自动机 $A=\mathrm{Q}\ \Sigma\times\mathbb{T},\ \delta,\ \varphi_{\mathrm{start}},\ F$ 在配置上导出一个转移关系：$\mathrm{C}\xrightarrow{\sigma\ a}\mathrm{C}'$ 当且仅当 C′ 是 $\bigwedge_{t\in\mathrm{C}}\hat{}\delta(t,\sigma:a)$ 的析取范式中的一个立方项。若存在某个 $\sigma\colon a$，使得 $\mathrm{C}\xrightarrow{\sigma\ a}\mathrm{C}'$，我们就记作 $\mathrm{C}\to\mathrm{C}'$，并称 C′ 为 C 的一个后继。若存在一个配置序列，单词 $\tau=\sigma_1\colon a_1\cdots\sigma_n\colon a_n$ 就被 A 接受，其中

配置序列 $C0, \ldots, C_n$ 满足：C_n 为初始配置，C_0 为终态配置，并且对于 $r = n, \ldots, 1$，$C_{r-1} \xleftarrow{\sigma_r:a_r} C_r$；注意，谓词自动机从右向左读取。$A$ 的语言 $L(A)$ 是被 A 接受的词的集合，而 $\mathbb{T}$-谓词自动机语言（PAL）是某个 $\mathbb{T}$-PA 的语言。$\mathbb{T}$-PAL 与谓词自动机语言 [21] 具有相同的闭包性质。

定理 7.3（闭包性质）。 $\mathbb{T}$-PAL 在交集和补运算下封闭，一元的 $\mathbb{T}$-PAL 也如此。任意 $\mathbb{T}$-PAL 都是局部对称的，即在等价关系 $\simeq$ 下封闭。

7.3 错误运行和作为$\mathbb{T}$-谓词自动机语言的证明空间语言

根据定理4.2，如果参数化程序 P 在极限结构 $\mathbb{T}_\infty$ 上存在极限程序 P_∞，那么在定理 5.3 中，我们无需检查包含关系 $\bigcup_{i\in I} L_{err}(P_i) \subset L(H)$，而可以取 $\mathbb{T}_\infty$ 作为通用结构并检查 $L_{err}(P_\infty) \subset L(H)$，从而避开构造语言的无限并。本节将证明，在某些有限性条件下，$L_{err}(P_\infty)$ 和 $L(H)$ 都可由 $\mathbb{T}_\infty$-PA 识别。我们的构造推广了命题 6.3 和 6.4]，且不依赖于 $\mathbb{T}_\infty$ 是否作为极限结构获得。

对于错误运行语言，我们的构造出现在以下定理的证明中：

定理 7.4（错误运行作为 $\mathbb{T}$-PAL）。 设 $\mathbb{T}$ 是词汇表 V_{Node}上的一个拓扑。设 $P, [\![-]\!]$ 是 $\mathbb{T}$上的一个局部对称并发程序。如果 $\mathbb{T}/\simeq$ 是有限的，那么 $L_{err}(P)$ 是一个一元的 $\mathbb{T}$-PAL。

证明依赖于这样一个事实：如果 $\mathbb{T}/\simeq$ 是有限的，那么每个命令在 P 下的原像都可由一个无量词公式定义；这可由命题 4.3 得出。

要将参数化证明空间构造为一个 $\mathbb{T}$-PAL，我们要求证明空间由一个范式下的有限基生成：

定义 7.4. 如果霍尔三元组的形式为 $\{\phi[\overrightarrow{a}\mu()]\}\ \sigma:\ b\ \{\psi[\overrightarrow{c}\mu()]\}$，其中 $\{a_i\}_i \subset N(b, \vec{c})$，且 $\psi[\vec{x}]$ 在语法上不是合取，则该霍尔三元组处于范式 。如果（参数化证明空间的）一个基中的每个霍尔三元组都处于范式，我们就称该基处于范式。

始终可以通过将三元组 $\{\phi\}\ \sigma:\ b\ \{\psi_1 \wedge \psi_2\}$ 拆分为 $\{\phi\}\ \sigma:\ b\ \{\psi_1\}$ 和 $\{\phi\}\ \sigma: b\ \{\psi_2\}$，并对前置条件中每个冗余节点 a_i 的数据信息添加存在量化，将包含单条命令的霍尔三元组化为范式。可以合理地假设，在算法细化方案中（采用 [21] 的风格），新发现的霍尔三元组始终只包含单条命令。

定理 7.5（参数化证明空间作为 $\mathbb{T}$-PAL）。 设 $\mathbb{T}$ 是词汇表 V_{Node}上的一个拓扑。设 H 是由一个有限基生成、处于范式的 $\mathbb{T}$-证明空间。如果对每个 $d \in \mathbb{N}_+$，$\mathbb{T}^d/\simeq$ 均为有限集，则 $L(H)$ 是一个 $\mathbb{T}$-PAL。

在 $L(H)$ 可由一元的 $\mathbb{T}$-PA 识别这一特殊情形下，我们也称 H 为一元的。

7.4 证明检查

由定理7.3可知，极限程序的行为与参数化证明空间之间的包含性检查可以归约为检查 $\mathbb{T}$-PA 的语言是否为空。因此，本节将聚焦于这一问题。一般的 $\mathbb{N}$-PA 非空性问题是不可判定的 [21, 命题 6.10]，且这种不可判定性也遗传到了 $\mathbb{T}$-PA。在 [21], 中提出了一种用于判定 $\mathbb{N}$-PA 非空性问题的半算法：它从初始配置出发探索可达配置；该算法对于非空性是可靠且完备的。该算法采用了良结构转移系统 [1, 24]中向后覆盖性检查的思路：为搜索空间赋予一个覆盖预序，作为剪枝策略。该算法的核心思想可以推广到 $\mathbb{T}$-PA。不过请注意，在更丰富的 $\mathbb{T}$-PA 空间中，还有许多其他不可判定性来源。

引理 7.6. 设 $\mathbb{T}$ 是一个 V-结构，Q 是与 $V[\mathbb{T}]$ 不相交的有限关系词汇表。假设在配置集合 $2^{\mathcal{F}^{\mathbb{T}}_{atom(Q)}}$ 上存在一个预序 $\preceq$，使得对于任意 $\mathbb{T}$-谓词自动机 $Q\ \Sigma \times \mathbb{T}, \delta, \varphi_{start}, F$，均有

(1) $\preceq$ 是可判定的。(2)（本质有限分支）给定任意配置 C，可以有效计算出 C 的后继中的一个有限子集 $\mathscr{C}$，使得对于 C 的任意后继 C'，都存在某个 $\bar{C}' \in \mathscr{C}$，使得 $\bar{C}' \preceq C'$。(3)（向下兼容性）对于任意配置 C 和 $\bar{C}$，若 $\bar{C} \preceq C$，则如果 C 是接受的，$\bar{C}$ 也是接受的；并且对于任意配置 C'，若 $C \to C'$，则存在某个配置 $\bar{C}'$，使得 $\bar{C} \to \bar{C}'$。

因此，对于 $\mathbb{T}$-PA 非空性存在一个半算法；对于其可达配置在 $\preceq$ 下构成良拟序的 $\mathbb{T}$-PA 类，该半算法将成为一个判定过程。

接下来需要找到一个满足引理 7.6 中条件的预序。可以将 [21] 中的覆盖预序推广。

定义 7.5（覆盖）。 设 $\mathbb{T}$ 是一个 V-结构，Q 是一个与 $V[\mathbb{T}]$ 不相交的有限关系词汇表。我们定义覆盖预序 $\preceq$如下：如果存在 π 的一个自同构 $\mathbb{T}$，使得对于所有 q（a_1, ……, $a_{ar(q)}$）$\in C$，都有 q（π（a_1）……, π（$a_{ar(q)}$）$\in C'$，则称配置 C 覆盖 C'，记作 $C \preceq C'$。

很容易验证，覆盖预序满足向下兼容性；特别是，在定义中，如果将自同构 π 替换为局部同构，这一点将不再成立。然而，当我们使用一般的 $\mathbb{T}$ 而不是结构 $\mathbb{N}$ 进行计算时，人们通常认为理所当然的基本操作会变得并非易事：(i) 判定覆盖预序；(ii) 枚举配置的代表性后继（即本质有限分支性质）。

以下定理给出了以算法方式实现这些操作的充分条件。条件（1）将判定覆盖预序的问题归结为判定 $\simeq$。条件（2）和（3）在定理7.5 关于 $\mathbb{T}$的条件中加入了可计算性考量；特别是，条件（2）确保 $\simeq$ 是可判定的。

T定理 7.7（$\mathbb{T}$-PA 空性）。假设 $\mathbb{T}$ 满足以下条件：

（1） $\mathbb{T}$是齐次的 .

(2) $\mathbb{T}^d/\simeq$ 对每个 $d \in \mathbb{N}_+$ 都是有限的，并且存在一种算法，输入 d 后输出一个无量词公式列表，用以定义这些等价类（其存在性由命题 4.3 保证）。

(3) 在 $\mathbb{T}$ 中，任意无量词 V[$\mathbb{T}$]-公式的可满足性都是可判定的。

那么，覆盖关系满足引理 7.6 中的条件 (1)–(3)。

例如，一元的 $\mathbb{N}$-PAs（其中 $\mathbb{N}$ 的词汇表为空）满足定理 7.7 中的三个条件，以及引理 7.6 中的良拟序条件，因此，[21] 的可判定性结果是本文结果的一个特例。再举一例，采用适当的 $\mathcal{V}_{\text{Node}}$（见附录 A），第 2 节中的拓扑有唯一的可数齐次极限 $\mathbb{T}_{\text{forest}}$。一元的 $\mathbb{T}_{\text{forest}}$-PAs 也满足全部四个条件。

更一般地，若 $\mathbb{T}$ 是一个极限结构，且它是由满足命题7.8中条件的 Fraïssé 类的可数 Fraïssé 极限得到的，那么 $\mathbb{T}$ 满足定理 7.7中的所有条件：

命题 7.8（[11] 中的定理 7.20）。设 T 是有限词汇表上的一个可数 Fraïssé 类，其极限为 $\mathbb{T}$。若对于每个$k \in \mathbb{N}$，T 中至多有 k 个生成元的结构（在同构意义下）只有有限多个，且给定 k即可计算出这些结构的列表，那么 $\mathbb{T}$ 中的元素可有限表示，并且 $\mathbb{T}$ 具有一个带参数的一阶可判定理论，这些参数取自 $\mathbb{T}$的底集。

作为引理 7.6、定理 7.7和定理 6.8的推论，我们概述了参数化布尔程序的安全性可判定的条件，从而完成了第 6.5节中关于参数化布尔程序的参数化证明空间完备性的讨论。

推论 7.9（参数化布尔程序的可判定性）。固定一个满足定理 7.7 中条件的结构 $\mathbb{T}$。此外，假设对于任意与 V[$\mathbb{T}$] 不相交且符号元数至多为 1 的有限关系词汇表 Q，覆盖预序在配置集合 $2^{\mathcal{F}^{\mathbb{T}}_{\text{atom}}(Q)}$ 上是良拟序。那么，对于任意以 $\mathbb{T}$ 为极限结构的参数化布尔程序，其安全性都是可判定的。

请注意，推论 7.9 中的条件由例 7.1中的极限结构满足，这意味着星形拓扑上的参数化布尔程序具有可判定性。

8 相关工作

参数化程序的安全性验证已得到广泛研究。由于该问题通常不可判定 [5]，已有大量研究 [2]通过限制拓扑、通信原语或安全性验证的表达能力，来研究可判定片段 [3, 4, 9, 10, 19, 26, 35]。我们的重点不是提出新的可判定片段或复杂度结果。因此，我们只与和本文贡献密切相关的研究进行比较。

程序验证中的对称性推理。我们对对称性的表述受到对网络中的局部对称性 [27, 38] 以及并发程序组合式验证 [39, 40]的研究启发。在所有这些研究中，都以某种形式体现了这样一种思想：不进行全局推理，而是针对进程的小邻域 进行推理。在 [39], 中，这一思想被专门用作参数化程序的组合式验证方法，其中组合式截断值 由形式化为群胚的局部对称性推导得出。我们的局部对称性所诱导的群胚，其任意一对对象之间至多有一条箭头（这是命题4.1的推论）。

在模型检测中，对称性约简 [17] 通过构造并检查商结构来取代原始结构，旨在解决具有许多同构组件的并发系统中的状态爆炸问题。在 [17], ，研究重点是令牌传递环以及在旋转下保持不变的正确性性质，并为此给出了截断结果。较弱的对称性概念，例如虚拟对称性 [15], ，也得到了研究。在[37]，研究了有限状态参数化系统中识别对称性的逆问题（称为对称模式）；对称模式是状态上的模拟预序，由转换器编码。我们的工作采用了不同的对称性概念，该概念基于同构邻域。

参数化程序验证。出于充分的理由，关于参数化无限状态程序验证的大多数研究都属于演绎验证这一广泛领域 [8, 14, 25, 33, 34, 42, 44]；也就是说，用户会以某种方式参与其中，提供所需的不变式或验证条件，而底层范式则是完成／检查这一证明。相比之下，提出参数化证明空间是为了自动发现此类不变式，或者至少在第 7 节算法的可判定性边界内发现其中的一个子类。在这一类工作中，[8] 尤为突出，因为它面向丰富的一类拓扑，并将拓扑的每个实例都建模为逻辑结构，而无需诉诸公理化。这里有一个相对完备性结果：在满足不变式的一个语义条件的前提下，安全性问题的 EPR 编码相对于广义 Ashcroft 不变式的一个真子集是完备的。相比之下，我们关于参数化证明空间相对完备性的条件，则与不变式的语法（即宽度）以及拓扑族中是否存在（局部的）对称性有关。

对于无限状态程序的算法验证，有一些研究旨在自动但不完全地生成全称量化不变式 [6, 18, 43]。多层次线程模块化证明 [22, 29]针对星形拓扑上的输入参数化程序 P，提出了一种完备算法；只要存在带有 k 全称量词的 Ashcroft 不变式，该算法便可奏效。

对于在一系列丰富拓扑结构上运行的参数化有限状态程序，文献[4], 给出了若干可判定性与复杂度结果，其中包括涉及通过成对会合通信的程序。人们普遍认为，这类通信足以对共享内存通信建模。然而，其可判定性结果 [4, 定理 10,11] 仅适用于其所称的同质拓扑结构，这与我们的同质拓扑结构有所不同；尤其是，它们不包含我们第2节示例中的树形拓扑结构，而本文证明了这些结构是可判定的。最早研究多种拓扑结构的论文是 [3], 一篇关于参数化验证的开创性论文，针对广播通信下的若干拓扑结构分别给出了可判定性结果。

无限字母表上的自动机。我们的工作扩展了 [21], 中引入的谓词自动机，它们又与 （交替）寄存器自动机（*ARA*） [11, 13, 23, 30, 41] 密切相关。[20], 将这种联系形式化，将自动机的元数与可用寄存器的数量联系起来

。在最常见的形式中，寄存器自动机在字母表 $\Sigma \times \mathbb{N}$ 上识别语言，其中 Σ 是有限的，而 $\mathbb{N}$ 是相等原子的集合。然而，为了涵盖多种对称性，我们的扩展—— $\mathbb{T}$-谓词自动机——以逻辑结构 $\mathbb{T}$ 为参数，并在 $\Sigma \times \mathbb{T}$ 上识别语言。更一般地，在 [12], 轨道有限交替自动机 中提出了这类自动机；它们读取 轨道有限名义集合 上的单词，该集合可以为寡态的 $\Sigma \times \mathbb{T}$ 编码一个字母表，见 $\mathbb{T}$ [11, 例 5。8]。我们对逻辑结构 $\mathbb{T}$ 的使用并不限于寡态结构，但命题 7.8 和定理 7.7 中的条件都蕴含寡态性。

参考文献

[1] Parosh Aziz Abdulla、K¯arlis Čer¯ans、Bengt Jonsson 和 Yih-Kuen Tsay。2000年。良拟序域上程序的算法分析。Information and Computation 160, 1 2000 109 127 10 1006 1999 2843（2001年7月），–。

[2] doi：。/inco.。Parosh A. Abdulla 和 Giorgio Delzanno。。参数化验证。International Journal on Software Tools for Technology Transfer 18，5（2016年10月），469–473。doi:10.1007/s10009-016-0424-3[3] Parosh Aziz Abdulla、Frédéric Haziza 和 Lukáš Holík。2013年。少数换取全部。载于《验证、模型检测与抽象解释》，Roberto Gia-cobazzi、Josh Berdine 和 Isabella Mastroeni（编）。Springer，柏林，海德堡，476–495。doi:10.1007/978-3-642-35873-9_28[4] Benjamin Aminof、Tomer Kotek、Sasha Rubin、Francesco Spegni 和 Helmut Veith。2018年。会合系统的参数化模型检测。Distributed Computing 31, 3 2018 187 222 10 1007 00446 017 0302 6（2018年6月），–。

[5] doi：。/s---Krzysztof R. Apt 和 Dexter C. Kozen。。有限状态并发系统自动验证的局限。Inform. Process. Lett. 22，6（1986年5月），307–309。doi: 10 1016 0020 0190 86 90071 2。/ - ( ) - Tamarah Arons、Amir Pnueli、Sitvanit Ruah、Ying Xu 和 Lenore Zuck。2001年。使用自动计算的归纳断言进行参数化验证？ 载于《计算机辅助验证》，Gérard Berry、Hubert Comon 和 Alain Finkel（编）。Springer，柏林，海德堡，221 234 10 1007 3 540 44585 4_19–。

[7] doi：。/---E. A. Ashcroft。。证明关于并行程序的断言。J. Comput. System Sci. 10, 1 1975 110 135 10 1016 0022 0000 75 80018 3（2001年2月），–。

[8] doi：。/S - ( ) - Rylo Ashmore、Arie Gurfinkel 和 Richard Trefler。。参数化一阶协议的局部推理。载于《NASA形式化方法》，Julia M. Badger 和 Kristin Yvonne Rozier（编）。施普林格国际出版社，Cham，36–53。doi: 10 1007 978 3 030 20652 9_3。/----A. R. Balasubramanian。。无限状态广播网络中的可覆盖性参数化验证。Information and Computation 278（2021年6月），104592 10 1016 2020 104592。doi：。/j.ic.。

[10] Thomas Ball、Sagar Chaki 和 Sriram K. Rajamani。2001年。多线程软件库的参数化验-证。载于《系统构造与分析的工具和算法》，Gerhard Goos、Juris Hartmanis、Jan Van Leeuwen、Tiziana Margaria 和 Wang Yi（编）。第2031卷。施普林格柏林海德堡，柏林，海德堡，158 173 10 1007 3 540 45319 9_12–。doi：。/---Mikołaj Bojańczyk。。略无限集合。（）。[12] Mikolaj Bojanczyk、Laurent Braud、Bartek Klin 和 Slawomir Lasota。2012年。迈-向名义计算。载于《第39届ACM SIGPLAN-SIGACT编程语言原理年会论文集》（POPL '12）。计算机协会，纽约，纽约州，美国，401–412。doi:10.1145/ 2103656.2103704[13] S. Demri 和 R. Lazic。2006年。带冻结量词和寄存器自动机的线性时序逻辑。载于《第21届IEEE计算机科学逻辑年会》（LICS' 06）。17–26。doi:10.1109/LICS.2006.31[14] Tayfun Elmas、Shaz Qadeer 和 Serdar Tasiran。2009年。原子动作演算。载于《第36届ACM SIGPLAN-SIGACT编程语言原理年会论文集》（POPL '09）。计算机协会，纽约，纽约州，美国，2 15 10 1145 1480881 1480885–。doi：。/。E.A.

[15] Emerson、J.W. Havlicek 和 R.J. Trefler。。虚拟对称性约简。载于《第十五届IEEE计算机科学逻辑年会论文集》（目录号 99 36332 121 131 10 1109 2000 855761 CB）。–。doi：。/LICS.。E.

[16] Allen Emerson 和 Kedar S. Namjoshi。。关于环的推理。载于《第22届ACM SIGPLAN-SIGACT编程语言原理研讨会论文集》（POPL '95）。计算机协会，纽约，纽约州，美国，85 94 10 1145 199448 199468–。doi：。/。E. Allen E merson 和 A. Prasad Sistla。。对称性与模型检测。Formal Methods in System Design 9, 1 1996 105 131 10 1007 00625970（2001年8月），–。

[18] doi：。/BF Michael Emmi、Rupak Majumdar 和 Roman Manevich。。事务内存的参数化验证。载于《第31届ACM SIGPLAN编程语言设计与实现大会论文集》（PLDI '10）。计算机协会，纽约，纽约州，美国，134–145。doi:10.1145/1806596.1806613[19] Javier Esparza、Pierre Ganty 和 Rupak Majumdar。2016年。参数化异步共享内存系统的验证。《ACM学报》63卷1期（2016年2月），10:1–10:48。doi:10.1145/2842603[20] Azadeh Farzan 和 Zachary Kincaid。2025年。谓词自动机之美。载于《追求洞见与优雅：纪念 Andreas Podelski 65 岁生日文集》。第14765卷。《计算机科学讲义》。[21] Azadeh Farzan、Zachary Kincaid 和 Andreas Podelski。2015年。无界并行性的证明空间。载于第42届ACM SIGPLAN-SIGACT编程语言原理年会论文集。ACM，印度孟买，407 420 10 1145 2676726 2677012–。doi：。/。

[22] Azadeh Farzan、Dominik Klumpp 和 Andreas Podelski。2024年。交换性简化参数化程序的证明。《ACM编程语言汇刊》8, 2024 2485 2513 10 1145 3632925 POPL（1月），–。

[23] doi：。/。Diego Figueira。。有限词与树上的交替寄存器自动机。《计算机科学逻辑方法》第8卷第1期（2012年3月），907。doi:10. 2168 8 1 22 2012 /LMCS-（：）A. Finkel 和 Ph. Schnoebelen。2001年。无处不在的良结构转移系统！《理论计算机科学》256卷1期（2001年4月），63–92。doi:10.1016/S0304-3975(00)00102-X[25] Cormac Flanagan 和 Stephen N. Freund。2020年。用于阻塞与非阻塞并发软件的 Anchor 验证器。《用于"阻塞与非阻塞并发软件的 Anchor 验证器"的软件制品》4，OOPSLA（2020年11月），156:1–156:29。doi:10.1145/3428224[26] Steven M. German 和 A. Prasad Sistla。1992年。关于多进程系统的推理。《ACM学报》39, 3 1992 675 735 10 1145 146637 146681（7月），–。

[27] doi：。/。Martin Golubitsky 和 Ian Stewart。。网络的非线性动力学：群胚形式体系。《美国数学会通报》43卷03期（2006年5月），305–365。doi:10.1090/S0273-0979-06-01108-6[28] Wilfrid Hodges。1997年。《简明模型论》。剑桥大学出版社，剑桥；纽约。[29] Jochen Hoenicke、Rupak Majumdar 和 Andreas Podelski。2017年。多层级线程模块化：组合式验证中的一颗明珠。载于第44届ACM SIGPLAN编程语言原理研讨会论文集。ACM，法国巴黎，473 485 10 1145 3009837 3009893–。doi：。/。Michael Kaminski 和 Nissim Francez。。有限记忆自动机。《理论计算机科学》134卷2期（1994年11月），329–363。doi:10.1016/0304-3975(94)90242-9[31] Richard M. Karp 和 Raymond E. Miller。1969年。并行程序模式。《计算机与系统科学杂志》3, 2 1969 147 195 10 1016 0022 0000 69 80011 5（5月），–。

[32] doi：。/S - ( ) - Zachary Kincaid。。并行程序的并行证明。博士论文。[33] Bernhard Kragl、Constantin Enea、Thomas A. Henzinger、Suha Orhun Mutluergil 和 Shaz Qadeer。2020年。异步程序的归纳序列化。载于第41届ACM SIGPLAN编程语言设计与实现会议论文集（PLDI 2020）。计算机协会，纽约，美国，227 242 10 1145 3385412 3385980–。doi：。/。Bernhard Kragl 和 Shaz Qadeer。。分层并发程序。载于《计算机辅助验证》，Hana Chockler 和 Georg Weissenbacher（编）。施普林格国际出版社，Cham，79 102 10 1007 978 3 319 96145 3_5–。doi：。/ - - - -

[35] Salvatore La Torre、Anca Muscholl 和 Igor Walukiewicz。。参数化异步共享内存系统的安全性几乎总是可判定的。载于第26届国际并发理论会议（CONCUR 2015）（《莱布尼茨国际信息学会议论文集》（LIPIcs），第42卷），Luca Aceto 和 David de Frutos Escrig（编）。Dagstuhl城堡—莱布尼茨信息学中心，德国 Dagstuhl，72 84 10 4230 2015 72–。doi：。/

[36] LIPIcs.CONCUR.。Leonid Libkin。。《有限模型论基础》。施普林格柏林海德堡，柏林，海德堡。doi: 10 1007 978 3 662 07003 1。/ - - - - Anthony W. Lin、Truong Khanh Nguyen、Philipp Rümmer 和 Jun Sun。2016年。正则对称模式。载于《验证、模型检测与抽象解释》，Barbara Jobstmann 和 K. Rustan M. Leino（编）。Springer，柏林，海德堡，455 475 10 1007 978 3 662 49122 5_22–。

[38] doi：。/ - - - - Kedar S. Namjoshi 和 Richard J. Trefler。。局部对称性与组合式验证。载于《验证、模型检测与抽象解释》，Viktor Kuncak 和 Andrey Rybalchenko（编）。第7148卷。施普林格柏林海德堡，柏林，海德堡，348 362 10 1007 978 3 642 27940 9_23–。doi：。/ - - - -

[39] Kedar S. Namjoshi 和 Richard J. Trefler。。参数化组合式模型检测。载于《系统构造与分析工具和算法》，Marsha Chechik 和 Jean-François Raskin（编）。Springer，柏林，海德堡，589 606 10 1007 978 3 662 49674 9_39–。doi：。/ - - - - Kedar S. Namjoshi 和 Richard J. Trefler。。局部μ演算的对称性约简。载于《系统构造与分析工具和算法》，Dirk Beyer 和 Marieke Huisman（编）。施普林格国际出版社，Cham，379 395 10 1007 978 3 319 89963 3_22–。doi：。/ - - - - Frank Neven、Thomas Schwentick 和 Victor Vianu。。无限字母表字符串的有限状态机。《ACM计算逻辑汇刊》5卷3期（2004年7月），403–435。doi:10.1145/1013560.1013562[42] Oded Padon、Giuliano Losa、Mooly Sagiv 和 Sharon Shoham。2017年。Paxos化为EPR：关于分布式协议的可判定推理。《ACM

编程语言 1, 2017 1 31 10 1145 3140568 2001
[43] OOPSLA（10月），–。doi：。/Amir Pnueli、Sitvanit Ruah 和 Lenore Zuck。。利用不可见不变量进行自动演绎验证。载于系统构造与分析工具和算法，Tiziana Margaria 和 Wang Yi（编）。Springer，柏林，海德堡，
82 97 10 1007 3 540 45319 9_7 –。doi：。/---Klaus v. Gleissenthall、
[44] Rami Gökhan Kıcı、Alexander Bakst、Deian Stefan 和 Ranjit Jhala。2019年。伪同步：异步分布式程序的同步验证。文章《伪同步：异步分布式程序的同步验证》的复现包3，POPL（2019年1月），59:1–59:30。doi:10.1145/3290372

第 2 节详述

词汇表与极限。固定高度 $h \in \mathbb{N}_{+}$。令 $\mathrm{V}_{\mathrm{Node}} := \{u,\ d\} \cup \{p_i \mid i \in \{-1,\ \ldots\ldots,\ h-1\}\}$，其中

- u 和 d 是分别表示“向上”和“向下”的一元函数符号；对于一个矩形节点 e，我们定义 $u(e) = d(e) = e$。• 对于每个 $i \in \{-1,\ \ldots\ldots,\ h-1\}$，$p_i$ 是一个二元谓词符号，用于测试两个节点的最近公共祖先（LCA）的深度是否为 i；对于属于两棵不同森林的两个节点，我们将它们的 LCA 的深度视为 -1。

对于任意 $k,\ l \in \mathbb{N}_{+}$，令 $\mathrm{V}_{\mathrm{Node}}$-结构 $\mathbb{T}_{k,\ l}$ 为由 l 棵高度为 h 的树构成的森林，其中矩形节点的分支因子为 k，如图 2 所示（不同之处在于树的高度为 h，而非 5）。令 $\mathbb{T} = \{\mathbb{T}_{k,\ l}\}_{(k,l)\in I}$，其中 $\mathrm{I} := \mathbb{N}_{+} \times \mathbb{N}_{+}$。

可以验证，命题7.1 的条件得到满足，因此 $\mathcal{T}$ 具有一个极限 $\mathbb{T}_{\mathrm{forest}}$，该极限可以构造为 Fraïssé 极限。注意，词汇表 $\mathcal{V}_{\mathrm{Node}}$ 的选择会影响 Fraïssé 极限的存在性，尽管函数符号 u 和 d足以描述一片森林，并且按照定义 7.1 的意义，即使使用 $\mathcal{V}_{\mathrm{Node}} = \{u, d\}$，极限仍然存在。如果没有二元谓词符号（即使将其替换为表示每个节点深度的一元谓词），$\mathrm{CL}(\mathcal{T})$ 也不会是 Fraïssé 类，因为 *amalgamation* 性质 将不成立。

良拟序。 我们首先证明两个引理。第一个引理的证明受 [11, 引理 1.15] 的证明启发。

引理 A.1。 设 Q 是一个有限关系词汇表，且 $\mathscr{C} \subset_2 \mathscr{F}^{\mathbb{T}}_{\mathrm{atom}}(\mathcal{Q})$ 是一个 $\mathbb{T}$-PA 配置集合。覆盖关系 $\preceq$在 $\mathscr{C}$ 上是良拟序，当且仅当存在一个良拟序集合 $(X, \preccurlyeq)$ 以及一个映射 $f\colon \mathscr{C} \to X$，该映射反映拟序：

$$\forall C, C' \in \mathscr{C}.\, f(C) \preccurlyeq f(C') \implies C \preceq C'.$$

证明。 $\Rightarrow$ 方向显然。对于 $\Leftarrow$ 方向，假设存在一个良拟序集 $(X, \preccurlyeq)$，以及一个映射 $f\colon \mathscr{C} \to X$，它反映该拟序。设 $\mathrm{C}_1, \mathrm{C}2, \ldots$ 是 $\mathscr{C}$ 上的一个无限序列。若这是一个反链，那么 $f(\mathrm{C}_1)\ f(\mathrm{C}_2)\ldots$ 也是反链，这与 $\preccurlyeq$ 是良拟序相矛盾。接下来只需证明它不可能是严格下降链。假设 $\mathrm{C} \prec \mathrm{C}'$，即 $\mathrm{C} \preceq \mathrm{C}'$，但 $\mathrm{C}' \not\preceq \mathrm{C}$。取自同构 π，使得对于任意 $q(\mathrm{i}1, \ldots, \mathrm{i}_n) \in \mathrm{C}$，都有 $q(\pi(\mathrm{i}_1) \ldots, \pi(\mathrm{i}_n)) \in \mathrm{C}$。于是 π 诱导出从 C 到 C′ 的单射。若它是双射，则自同构 π^{-1} 见证了 $\mathrm{C}' \preceq \mathrm{C}$，这就产生了矛盾。因此，我们有 $\mathrm{C} \prec \mathrm{C}' \Rightarrow |\mathrm{C}| < |\mathrm{C}'|$，这说明 $\mathrm{C}_1, \mathrm{C}2, \ldots$ 不可能构成严格下降链。□

引理 A.2。 设 Q 是一个有限关系词汇表，且 $\mathscr{C} \subset_2 \mathscr{F}^{\mathbb{T}}_{\mathrm{atom}}(\mathcal{Q})$ 是一个 $\mathbb{T}$-PA 配置集合。设 $\mathscr{C}'$ 是将每个配置中的所有零元符号移除后的 $\mathscr{C}$。如果覆盖关系 wqo $\mathscr{C}'$，那么它也 wqo $\mathscr{C}$。

证明。假设覆盖关系 wqo $\mathscr{C}'$。令 Q_0 为 Q 中的所有零元符号。则 $2^{\mathcal{Q}_0}$ 在子集关系下是良拟序。乘积 $\mathscr{C}' \times 2^{\mathcal{Q}_0}$ 在积序下是良拟序。

定义映射 $f\colon \mathscr{C} \to \mathscr{C}' \times 2^{\mathcal{Q}_0}$，方法是将零元符号与其他原子公式分开。于是 f保持拟序。由引理 A.1， $\mathscr{C}$ 在覆盖关系下是良拟序。□

我们现在可以证明，覆盖关系（定义7.5）对 $\mathbb{T}_{\mathrm{forest}}$-PA 的单子配置构成良拟序：

定理 A.3。 设 $\mathbb{T} := \mathbb{T}_{\mathrm{forest}}$，且 Q 是一个有限关系词汇表（与 $\mathrm{V}_{\mathrm{Node}}[\mathbb{T}]$ 不相交），其中符号的元数至多为 1。设 $\mathscr{C} \subset 2^{\mathscr{F}^{\mathbb{T}}_{\mathrm{atom}}(\mathcal{Q})}$ 是一个 $\mathbb{T}$-PA 配置集合。则覆盖关系是 $\mathscr{C}$上的良拟序。

证明。根据引理 A.2，我们可以假设 Q 中的符号都是一元的。根据引理 A.1，只需找到一个良拟序集合 $(X, \preccurlyeq)$ 以及一个映射 f: $\mathscr{C} \to X$，使其反映该拟序。令 $L := 2^{\mathcal{Q}} \times \{-1, \ldots, h-1\}$，其在恒等关系下为良拟序。令 X为 L-标记的树集合，令 $\preccurlyeq$为 L-标记的拓扑小 minor。根据 Kruskal 树定理，$(X, \preccurlyeq)$ 是一个良拟序集合。

对于任意配置 $\mathrm{C} \in \mathscr{C}$，将 $f(\mathrm{C})$ 定义为以下方式得到的 L标记树：

- 取包含 C 中所有节点及其根节点的 $\mathbb{T}_{\text{森林}}$ 的极小子森林，
- 用 $(\{q \in \mathrm{Q} \mid q(a) \in \mathrm{C}\}\ d)$ 标记该子森林中的每个节点，其中 d是 a的深度，并且
- 添加一个标记为 $(\varnothing,\ -1)$ 的额外节点，并将其连接到子森林中的每个根节点。

可以验证， f确实反映了该拟序。 □

B 证明

全文中，我们区分圆括号和（方）括号的用法：前者用于（具体或符号）函数应用，后者用于公式代换。例如，若 $t[u_1, \ldots, u_n]$ 是一个 $\mathrm{V}_{\mathrm{Node}}$-项，且 $(a_1,\ \ldots\ldots,\ a_n)$ 是一个节点元组，那么 $t(\vec{a})$ 是一个节点，而 $t[\vec{a}]$ 是一个公式。$\mu(a_1)$ 也是一个公式，因为 μ 是一个具体函数符号。

我们用 $+\!\!+$ 表示两个元组的串接。对于任意拓扑或通用结构 $\mathbb{T}$，以及任意由 $\Sigma \times \mathbb{T}$ 中的元素组成的单词 ρ，我们用 $\mathrm{nodes}(\rho)$ 表示出现在 ρ 中的节点集合。

B.1 第 4 节的证明 4

命题 B.1 (命题 4.1)。 对于任意 $\vec{a} \in \mathbb{T}^d$ 和 $\vec{b} \in \mathbb{T}'^d$，$\vec{}$从 $N(\vec{a})$ 到 $N(b)$ 且将 $\vec{a}$映射到 $\vec{b}$的同构至多有一个。

证明。设 V 为 $\mathbb{T}$ 和 $\mathbb{T}'$的共同词汇表。设 f $\vec{}$和 g是从 $N(\vec{a})$ 到 $N(b)$ 的同构，使得 $f(\vec{a}) = b$且$g(\vec{a}) = \vec{b}$。设 $c \in N(\vec{a})$。则存在某个 V-项 $t[\vec{u}]$ 使得$c = t(\vec{a})$。由此可得 $f(c) = f(t(\vec{a})) = t(f(\vec{a})) = t(\vec{b})$。类似地，我们有 $g(c) = t(\vec{b})$。因此， f和 g相同。□

命题 B.2（命题 4.2）。 设 $\mathbb{T}$ 和 $\mathbb{T}'$ 是词汇表 $\mathcal{V}$上的结构。对于任意 $\vec{a} \in \mathbb{T}^d$和 $\vec{b} \in \mathbb{T}'^d$，有 $\vec{a} \simeq \vec{b}$当且仅当对于每个无量词 $\mathcal{V}$-公式 $\varphi[x_1, \ldots, x_d]$，$\mathbb{T} \models \varphi[\vec{a}] \iff \mathbb{T}' \models \varphi[\vec{b}]$。

证明。 $\Rightarrow$ 假设 $\vec{a} \simeq \vec{b}$。那么存在一个同构 $\varphi\colon N(\vec{a}) \to N(\vec{b})$，它诱导出一个嵌入 $N(\vec{a}) \hookrightarrow \mathbb{T}'$。无量词公式在嵌入下是绝对的。

$\Leftarrow$: 假设对于每个无量词 V-公式 $\varphi[x_1, \ldots, x_d]$,$\mathbb{T} \models \varphi[\vec{a}] \Leftarrow\Rightarrow \mathbb{T}' \models \varphi[\vec{b}]$。定义关系 $f \in N(\vec{a}) \times N(\vec{b}) :=\{(t(\vec{a})\ t(\vec{b})) \mid t$是一个 V-项}。容易证明， f是（图像）

从 $\vec{a}$ 到 $\vec{b}$的局部同构。例如，若 t_1, t_2 是 V-项，且 $t_1(\vec{a}) = t_2(\vec{a})$，令 $\varphi := t_1[\vec{u}] = t_2[\vec{u}]$，则有 $t_1(b) = t_2(\vec{b})$，这表明定义良好。□

命题 B.3 (命题 4.3)。对于 $d \in \mathbb{N}_+$，如果商集 $\mathbb{T}^d/\simeq$ 是有限的，那么 $\simeq$ 的每个等价类都可以由一个无量词 $\mathcal{V}$-公式定义。

证明。将 $\mathbb{T}^d/\simeq$ 写成 $\{E_1, \ldots, E_n\}$。对于 $i = 1, \ldots, n$，令 F_i 为所有满足以下条件的无量词公式的集合：这些公式有 d 个自由变量，且由 E_i 中的元组满足。令 $i, j \in \{1, \ldots, n\}$ 两两不同。根据命题 4.2，存在一个无量词公式 φ，使得 $\varphi \in \mathrm{F}_i$ 且 $\varphi \notin \mathrm{F}_j$，或者 $\varphi \in \mathrm{F}_j$ 且 $\varphi \notin \mathrm{F}_i$。在前一种情况下，令 $\psi_{i,j} := \varphi$。在后一种情况下，令 $\psi_{i,j} := \neg\varphi$。那么，$E_i$ 由 $\bigwedge_{j \neq i} \psi_{i,j}$ 定义。□

对于关系词汇表，公式集合 $\mathrm{F}_1, \ldots, \mathrm{F}_n$ 也称为秩-0 d-类型 [36, 定义 3.14]。

B.2 第 5 节的证明

引理 B.4。设 $\mathbb{T}$ 是一个通用结构。设 $\sigma_1 : v_1 \cdots \sigma_n : v_n \in (\mathbb{T} \times \Sigma)^*$, U 是一个包含 $\vec{v}$的邻域，并且 β 是从 $\vec{v}$ 到 $\mathbb{T}$上另一个元组的局部同构。那么

$$\beta^* \mathrm{Global}_{\beta(U)}[\![\langle \sigma_1 : \beta(v_1)\rangle \cdots \langle \sigma_n : \beta(v_n)\rangle]\!] = \mathrm{Global}_U[\![\langle \sigma_1 : v_1\rangle \cdots \langle \sigma_n : v_n\rangle]\!].$$

证明。容易验证

$$\beta^* \mathrm{Global}_{\beta(U)}[\![\langle \sigma_1 : \beta(v_1)\rangle \cdots \langle \sigma_n : \beta(v_n)\rangle]\!] = \beta^* \mathrm{Global}_{\beta(U)}[\![\langle \sigma_1 : \beta(v_1)\rangle]\!] \circ \cdots \circ \beta^* \mathrm{Global}_{\beta(U)}\langle \sigma_n : \beta(v_n)\rangle.$$

另一方面，

$$\mathrm{Global}_U[\![\langle \sigma_1 : v_1\rangle \cdots \langle \sigma_n : v_n\rangle]\!] = \mathrm{Global}_U[\![\langle \sigma_1 : v_1\rangle]\!] \circ \cdots \circ \mathrm{Global}_U[\![\langle \sigma_n : v_n\rangle]\!].$$

因此，只需证明

$$\beta^* \mathrm{Global}_{\beta(U)}[\![\langle \sigma_1 : \beta(v_1)\rangle]\!] \subset \mathrm{Global}_U[\![\langle \sigma_1 : v_1\rangle]\!].$$

（关于包含关系的另一方向，注意到 β^{-1} 是从 $\beta(\vec{v})$ 到 $\vec{v}$ 的局部同构。）

令 $(\mu_1, \mu_1') \in \beta^*\text{全局}_{\beta(U)}[\![\sigma_1 : \beta(v_1)]\!]$。设 $\mu := (\beta^{-1})^* \mu_1$ 和 $\mu' := (\beta^{-1})^* \mu_1'$。则 $(\mu, \mu') \in \text{全局}_{\beta(U)}[\![\sigma_1 : \beta(v_1)]\!]$。根据定义，我们有

$$(\mu|_{N(\beta(v_1))}, \mu'|_{N(\beta(v_1))}) \in [\![\langle \sigma_1 : \beta(v_1)\rangle]\!], \tag{3}$$

$$\mu|_{\overline{N(\beta(v_1))}} = \mu'|_{\overline{N(\beta(v_1))}}. \tag{4}$$

令 γ 为 β在 $N(v_1) \to N(\beta(v_1))$ 在 v_1到 $\beta(v_1)$ 上的限制，它是一个局部同构。由 (3)，我们有

$$(\gamma^*(\mu|_{N(\beta(v_1))}), \gamma^*(\mu'|_{N(\beta(v_1))})) \in [\![\langle \sigma_1 : v_1\rangle]\!],$$

其中左端恰好为 $(\mu_1|_{N(v_1)}, \mu_1'|_{N(v_1)})$。另一方面，由 (4) 可得

$$\mu_1|_{\overline{N(v_1)}} = \mu_1'|_{\overline{N(v_1)}}.$$

因此，$(\mu_1, \mu_1') \in \text{全局}_U[\![\sigma_1 : v_1]\!]$。□

定理 B.5 (定理5.1)。规则 S-PARAM 对于任意元组 $\vec{u}$、$\vec{v}$、$\vec{w}$、$\vec{u}'$、$\vec{v}'$以及 $\vec{w}'$，均是可靠的，其适用范围为 $\mathbb{T}$。

证明。假设 $\left\{\phi[\overrightarrow{u}\,\mu()]\right\}\sigma_1 : v_1 \cdots \sigma_n : v_n \left\{\psi[\overrightarrow{w}\,\mu()]\right\}$ 相对于 $[\![-]\!]$（通用结构上带索引命令的语义函数）是有效的。设 $U = N(\vec{u} + \vec{v} + \vec{w})$ 和 β 是来自 U 的局部同构。我们要证明 $\left\{\phi[\overrightarrow{\beta\,u}\,\mu(())]\right\}$ $\langle \sigma_1 : \beta(v_1)\rangle \cdots \langle \sigma_n : \beta(v_n)\rangle$ $\left\{\psi[\overrightarrow{\mu(\beta(w))}]\right\}$ is also valid.

注意，$N(\beta(\vec{u}) + \beta(\vec{v}) + \beta(\vec{w})) = N(\beta(\vec{u} + \vec{v} + \vec{w})) = \beta(U)$。令 $(\mu, \mu') \in \text{全局}_{\beta(U)}[\![\sigma_1 : \beta(v_1) \cdots \sigma_n : \beta(v_n)]\!]$。假设 $\mu \models \phi[\overrightarrow{\beta\,u}\,\mu(())]$。那么 $(\beta^*\mu, \beta^*\mu') \in \text{全局}_U[\![\sigma_1 : v_1 \cdots \sigma_n : v_n]\!]$ 由引理 B.4 得出，并且 $\beta^*\mu \models \phi[\overrightarrow{u}\,\mu()]$。因此，$\beta^*\mu' \models \psi[\overrightarrow{w}\,\mu()]$，从而 $\mu' \models \psi[\overrightarrow{\beta\,w}\,\mu(())]$。□

定理 B.6 (定理 5.2)。证明空间语言在等价关系下封闭 $\simeq$。

证明。设 H 是一个 $\mathbb{T}$-证明空间。假设 $\sigma_1 : v_1 \cdots \sigma_n : v_n \in \mathrm{L}(\mathrm{H})$。那么 $\{\top\}\,\sigma_1 : v_1 \cdots \sigma_n : v_n\,\{\bot\} \in \mathrm{H}$。假设 $\rho \in (\Sigma \times \mathbb{T})^*$ 且 $\rho \simeq \sigma_1 : v_1 \cdots \sigma_n : v_n$。那么存在一个元组 $\vec{v}' \in \mathbb{T}^n$，使得 $\vec{v}' \simeq \vec{v}$ 且 $\rho = \sigma_1 : v_1' \cdots \sigma_n : v_n'$。根据 S-param，我们有 $\{\top\}\,\rho\,\{\bot\} \in \mathrm{H}$。因此，$\rho \in \mathrm{L}(\mathrm{H})$。□

引理 B.7。设 $\{P_i, [\![-]\!]_i\}_{i \in I}$ 是定义在拓扑 $\{\mathbb{T}_i\}_{i \in I}$ 上的、具有通用结构 $\mathbb{T}$ 的参数化程序。对于任意 $i \in \mathrm{I}$，关于 $(\Sigma \times \mathbb{T}_i)^*$ 的任意霍尔三元组的有效性，相对于 $[\![-]\!]_i$ 和 $[\![-]\!]$是相同的。

证明。设 $i \in \mathrm{I}$。对于任意 $\sigma : v \in \Sigma \times \mathbb{T}_i$，我们有 $[\![\sigma : v]\!] = [\![\sigma : v]\!]_i$。因此，对于任意 $\rho \in (\Sigma \times \mathbb{T}_i)^*$ $V =: N$。□

于任意 I 和 ()，对于 (节点(ρ))，我们有 $\mathrm{Global}_V[\![\rho]\!] = \mathrm{Global}_V[\![\rho]\!]_i$。

定理 B.8 （定理5.3）。若存在通用结构 $\mathbb{T}$ 和有效的 $\mathbb{T}$-证明空间 H，使得 $\bigcup_{i \in I} \mathrm{L}_{\mathrm{err}}(P_i) \subset \mathrm{L}(\mathrm{H})$，则参数化程序 $\mathrm{P} = \{P_i, [\![-]\!]_i\}_{i \in I}$ 是安全的。

证明。设 $\mathrm{P} = \{P_i, [\![-]\!]_i\}_{i \in I}$ 是一个参数化程序。假设存在一个通用结构 $\mathbb{T}$ 和一个有效的 $\mathbb{T}$-证明空间 H，使得 $\bigcup_{i \in \mathrm{I}} \mathrm{L}_{\mathrm{err}}(P_i) \subset \mathrm{L}(\mathrm{H})$。令 $i \in \mathrm{I}$ 且 $\rho \in \mathrm{L}_{\mathrm{err}}(P_i)$。那么 $\rho \in \mathrm{L}(\mathrm{H})$。根据参数化证明空间的可靠性和引理 B.7，霍尔三元组 $\{\top\}\,\rho\,\{\bot\}$ 相对于 $[\![-]\!]_i$ 有效，因此 ρ 不可行。由此，P 中的每个成员都是安全的，所以 P 是安全的。□

B.3 第 6 节的证明6

命题 B.9（命题 6.2）。给定一个 $\mathrm{V}_{\mathrm{Node}}$-结构 $\mathbb{T}$，对于任意 QF-Ashcroft 公式 $\varphi[u_1, \ldots, u_n]$和 $\vec{a} \in \mathbb{T}^n$，存在唯一的（在等价意义下）扩展断言 $\phi[\mu(b), \mathrm{l}(b)]$，，其中$b$是来自 $N(\vec{a})$ 的一个元组；使得对于任意从 $\vec{a}$ 出发的局部同构 β，扩展断言 $\phi[\overrightarrow{\beta\,b}\,\mu(()) \longrightarrow \mathrm{l}(\beta(b))]$等价于 $\varphi[\beta(\vec{a})]$。

证明。设 $\varphi[u_1, \ldots, u_n]$ 是一个 QF-Ashcroft 公式，且 $\vec{a} \in \mathbb{T}^n$。可以找到 V-项 $t_1[\vec{u}] \ldots t_m[\vec{u}]$，使其在（逻辑上）等价于 $\varphi[\vec{a}]$ 以下形式的公式的布尔组合

(1) $\phi_{\mathrm{Node}}[\vec{u}]$, (2) $\phi_{\mathrm{Data}}[\mu(t_1[\vec{u}]) \ldots, \mu(t_m[\vec{u}])]$, 和 (3) $\phi_{\mathrm{Loc}}[\mathrm{l}(t_1[\vec{u}]) \ldots, \mathrm{l}(t_m[\vec{u}])]$。

其中 ϕ_{Node} 是一个无量词的 V_{Node}-公式， ϕ_{Data} 是一个 V_{Data}-公式，且 ϕ_{Loc} 是一个 V_{Loc}-公式。因此，我们可以写成

$$\begin{aligned}\varphi[\vec{u}] \equiv \bigvee_{r=1}^{k} (&\phi^{(r)}_{\text{Node}}[\vec{u}] \\ &\wedge \phi^{(r)}_{\text{Data}}[\mu(t_1[\vec{u}]), \ldots, \mu(t_m[\vec{u}])] \\ &\wedge \phi^{(r)}_{\text{Loc}}[\mathrm{l}(t_1[\vec{u}]), \ldots, \mathrm{l}(t_m[\vec{u}])]).\end{aligned}$$

令 $R := \{r \in \{1,\ldots,k\} \mid \phi_{(r)\text{节点}}[\vec{a}] \equiv \top\}$。定义

$$\phi[x_1,\ldots,x_m,l_1,\ldots,l_m] := \bigvee_{r\in R} \phi^{(r)}_{\text{Data}}[\vec{x}] \wedge \phi^{(r)}_{\text{Loc}}[\vec{l}],$$

$$\vec{b} := (t_1(\vec{a}),\ldots,t_m(\vec{a})) \in \mathbb{T}^m.$$

$\longrightarrow \longrightarrow$那么， $\phi[\mu(b), \mathrm{l}(b)]$ 是一个与 $\varphi[\vec{a}]$ 等价的扩展断言。唯一性很容易得出。

令 β 为从 $\vec{a}$ 出发的局部同构。由于局部同构保持无量词 V_{Node}-公式的真值，我们有

$$\begin{aligned}\varphi[\beta(\vec{a})] \equiv \bigvee_{r\in R} (&\phi^{(r)}_{\text{Data}}[\mu(t_1(\beta(\vec{a}))), \ldots, \mu(t_m(\beta(\vec{a})))] \\ &\wedge \phi^{(r)}_{\text{Loc}}[\mathrm{l}(t_1(\beta(\vec{a}))), \ldots, \mathrm{l}(t_m(\beta(\vec{a})))]).\end{aligned}$$

由于 $t_i(\beta(\vec{a})) = \beta(b_i)$，这等价于 $\phi[\overrightarrow{\beta\, b}\mu(\,) \longrightarrow \mathrm{l}(\beta(b))]$。□

引理 B.10 (引理6.3)。设 P是一个定义在有限拓扑上的程序 $\mathbb{T}_i \subset \mathbb{T}$。对于任意 Ashcroft 断言 Φ， $H_\Phi^{\mathbb{T}_i}$中的扩展 Hoare 三元组均有效，只要 Φ 是 P的 Ashcroft 不变式。

证明。若以下扩展 Hoare 三元组均有效，则 $H_\Phi^{\mathbb{T}_i}$中的扩展 Hoare 三元组有效：对于所有带索引命令 σ： $a_0 \in \Sigma \times \mathbb{T}_i$，

$$\begin{aligned}&\left\{\wedge_{a\in\mathbb{T}_i} \mathrm{l}(a) = l_{\text{init}}\right\} \in \left\{\wedge_{\vec{a}\in\mathbb{T}_i^k} \varphi[\vec{a}]\right\} \\ &\left\{\wedge_{\vec{a}\in\mathbb{T}_i^k} \varphi[\vec{a}]\right\} \langle\sigma : a_0\rangle \left\{\wedge_{\vec{a}\in\mathbb{T}_i^k} \varphi[\vec{a}]\right\} \\ &\left\{\wedge_{\vec{a}\in\mathbb{T}_i^k} \varphi[\vec{a}]\right\} \in \left\{\wedge_{a\in\mathbb{T}_i} \mathrm{l}(a) \neq l_{\text{err}}\right\}\end{aligned}$$

这些只是通过将每个全称量化替换为合取得到的 Φ 的初始化、延续和安全条件的重述。□

引理 B.11（引理 6.4）。设 $\mathrm{P} = \{\mathrm{P}_i\}_{i\in I}$ 是一个定义在拓扑 $\{\mathbb{T}_i\}_{i\in I}$ 上、具有通用结构 $\mathbb{T}$ 的参数化程序。假设存在一个有效的有限生成扩展 $\mathbb{T}$-证明空间 H，使得对于每个 $i \in \mathrm{I}$、 $\rho \in \mathrm{L}_{\text{err}}(P_i)$ 和 $b \in \mathbb{T}_i$，都存在 $\mathbb{T}_i$ 的某个有限子集 U，满足 $\{\wedge_{a\in U} \mathrm{l}(a) = l_{\text{init}}\}\ \rho\{\mathrm{l}(b) \neq l_{\text{err}}\} \in \mathrm{H}$。则可通过一个有限生成的 $\mathbb{T}$-证明空间证明 P 是安全的。

证明概要。设 H 是 H 的一个有限基。设 H' 包含以下霍尔三元组：对于 H 中的每个扩展 Hoare 三元组 $\left\{\phi[\overrightarrow{a}\mu(\,) \longrightarrow \mathrm{l}(a)]\right\}\ \rho\ \left\{\psi[\overrightarrow{c}\mu(\,) \longrightarrow \mathrm{l}(c)]\right\}$，以及在 $\{a_i\}_i \cup$ 个节点上满足 $\lambda \xrightarrow{\rho} \lambda'$ 的每对位置映射 λ 和 λ' $\rho \cup \{c_j\}_j$，

$$\left\{\phi[\overrightarrow{\mu(a)}, \overrightarrow{\lambda(a)}]\right\}\ \rho\ \left\{\psi[\overrightarrow{\mu(c)}, \overrightarrow{\lambda'(c)}]\right\},$$

也就是说，分别将 λ 和 λ' 代入前置条件和后置条件中的 l，并进行化简；可以证明，这会得到一个断言中的公式。接着，可以证明 H' 是一个由有效霍尔三元组构成的有限集。我们的目标是证明，P 可以通过由 H，生成的 $\mathbb{T}$-证明空间 H′ 证明为安全；这依赖于以下可通过对推导进行归纳证明的断言：

命题 B.12. 对于 H 中任意扩展 Hoare 三元组 $\left\{\phi[\overrightarrow{a}\mu(\,) \longrightarrow \mathrm{l}(a)]\right\}\ \rho\left\{\psi[\overrightarrow{c}\mu(\,) \longrightarrow \mathrm{l}(c)]\right\}$，对于 $\{a_i\}_i \cup$节点$(\rho) \cup \{c_j\}_j$上的任意一对位置映射 λ和 λ'，只要满足 $\lambda \xrightarrow{\rho} \lambda'$，霍尔三元组$\left\{\phi[\overrightarrow{a}\mu(\,)\ \overrightarrow{\lambda\ a}(\,)]\right\}\ \rho\ \left\{\psi[\overrightarrow{c}\mu(\,)\ \overrightarrow{\lambda'\ c}(\,)]\right\}$就属于 H′。

令 $i \in \mathrm{I}$ 和 $\rho \in \mathrm{L}_{\text{err}}(P_i)$。我们要证明 $\{\top\}\ \rho\{\bot\}$ 属于 H′。令 λ 为定义在 nodes(ρ) 上的一个初始位置映射，并令$\lambda \xrightarrow{\rho} \lambda'$。由于 ρ是一个错误运行，因此存在某个 $b \in \mathbb{T}_i$，使得 $\lambda'(b) = l_{\text{err}}$。于是，根据假设，存在 $\mathbb{T}_i$ 的一个有限子集 U，使得 $\{\wedge_{a\in U}\mathrm{l}(a) = l_{\text{init}}\}\ \rho\{\mathrm{l}(b) \neq l_{\text{err}}\} \in \mathrm{H}$。根据命题 B.12，可得 $\{\wedge_{a\in U}\lambda(a) = l_{\text{init}}\}\ \rho\{\lambda(b) \neq l_{\text{err}}\} \in \mathrm{H}'$。注意，$(\wedge_{a\in U}\lambda(a) = l_{\text{init}}) \equiv \top$ 且 $(\lambda(b) \neq l_{\text{err}}) \equiv \bot$。□

引理 B.13. 设 $\{\mathbb{T}_i\}_{i\in I}$是一个拓扑族，其通用结构为 $\mathbb{T}$。假设 $\{\mathbb{T}_i\}_{i\in I}$有一个秩为 $k+1$ 的基 $\{\mathbb{T}_j\}_{j\in\mathcal{J}}$，其中每个 $\mathbb{T}_j$ 都是有限的。令 H 为由 $\bigcup_{j\in\mathcal{J}} H_\Phi^{\mathbb{T}_j}$ 生成的扩展 $\mathbb{T}$-证明空间。令 $i \in \mathrm{I}$。

- 对于任意 $\vec{a} \in \mathbb{T}_i^k$，存在 $\mathbb{T}_i$ 的有限子集 U，使得
$$\{\wedge_{a\in U} \mathrm{l}(a) = l_{\text{init}}\} \in \{\varphi[\vec{a}]\} \in \mathcal{H}.$$
- 对于任意 σ： $a_0 \in \Sigma\times\mathbb{T}_i$ 和 $\vec{a} \in \mathbb{T}_i^k$，存在 $\mathbb{T}_i^k$ 的有限子集U，使得
$$\{\wedge_{\vec{a}\in U} \varphi[\vec{a}]\} \langle\sigma : a_0\rangle \{\varphi[\vec{a}]\} \in \mathcal{H}.$$
- 对于任意 $b \in \mathbb{T}_i$，存在 $\mathbb{T}_i^k$ 的有限子集 U，使得
$$\{\wedge_{\vec{a}\in U} \varphi[\vec{a}]\} \in \{\mathrm{l}(b) \neq l_{\text{err}}\} \in \mathcal{H}.$$

证明。我们证明第二个要点。其余两个类似。设 σ： $a_0 \in \Sigma \times \mathbb{T}_i$ 和 $\vec{a} \in \mathbb{T}_i^k$。那么存在某个 $j \in \mathrm{J}$ 以及嵌入 f: $\mathbb{T}_j \hookrightarrow \mathbb{T}_i$，使得 a_0， $\vec{a} \in f(\mathbb{T}_j)^{k+1}$。令 β: $\mathbb{T}_j \to f(\mathbb{T}_j)$ 为将陪域限制到其像上的 f。那么 β 是一个同构，并且

$$\begin{aligned}&\left\{\wedge_{\vec{b}\in\mathbb{T}_j^k} \varphi[\vec{b}]\right\} \langle\sigma : \beta^{-1}(a_0)\rangle \left\{\varphi[\beta^{-1}(\vec{a})]\right\} \\ \xRightarrow{\text{S-Param}}\ &\left\{\wedge_{\vec{b}\in\mathbb{T}_j^k} \varphi[\beta(\vec{b})]\right\} \langle\sigma : a_0\rangle \{\varphi[\vec{a}]\} \\ \iff\ &\left\{\wedge_{\vec{a}\in f(\mathbb{T}_j)^k} \varphi[\vec{a}]\right\} \langle\sigma : a_0\rangle \{\varphi[\vec{a}]\}.\end{aligned}$$

因此，我们可以取 $U = f(\mathbb{T}_j)^k$。□

定理 B.14 (定理 6.5)。设 $\mathcal{P} = \{\mathrm{P}_i\}_{i\in I}$ 是一个参数化程序，其定义在一族拓扑 $\mathcal{T} = \{\mathbb{T}_i\}_{i\in I}$ 上，且其通用结构为 $\mathbb{T}$。若存在一个宽度为 Φ 的 Ashcroft 不变式 k，适用于 $\mathcal{P}$，且 $\mathcal{T}$具有一个秩为 $k+1$、由有限拓扑组成的有限基，那么就存在一个有限生成的 $\mathbb{T}$-证明空间，可证明 $\mathcal{P}$ 是安全的。

证明。假设 Φ 是 P 的宽度为 k 的 Ashcroft 不变式，且 T 有有限基 $\{\mathbb{T}_j\}_{j\in\mathcal{J}}$，其秩为 $k+1$，其中每个 $\mathbb{T}_j$ 都是有限的。令 H 为由该有限基生成的扩展 $\mathbb{T}$-证明空间

$\bigcup_{j\in\mathcal{J}} H_\Phi^{\mathbb{T}_j}$，根据引理 6.3，它是有效的。根据引理 6.4，只需证明：对于任意 $i\in\mathcal{I}$、$\rho\in\mathcal{L}_{\text{err}}(P_i)$，以及 $b\in\mathbb{T}_i$，存在某个有限子集 U，它属于 $\mathbb{T}_i$，使得

$$\{\bigwedge_{a\in U} 1(a)=l_{\text{init}}\}\ \rho\ \{1(b)\neq l_{\text{err}}\}\in\mathcal{H}.$$

这可由引理 B.13 以及合取规则和顺序规则推出。□

定理 B.15（定理 6.6）。对于任意参数化程序 P 和 P′，若 P $\subset$ P′，且 P′ 中的每个程序都是 P 中某个程序的子程序，则 P 安全当且仅当 P′ 安全；此外，若 $\mathbb{T}$ 是 P 和 P′的通用结构，则任意 $\mathbb{T}$-证明空间 H 证明 P 的安全性当且仅当它证明 P′的安全性。

证明。设 P 和 P′ 是参数化程序，满足 P $\subset$P′，且 P′ 中的每个程序都是 P 中某个程序的子程序。由于 P $\subset$ P′，若 P′ 安全，则 P 也安全。假设 P′ 不安全，且 ρ 是程序 P′ $\in$ P′的一次可行错误运行。那么存在一个程序 P $\in$ P，它是 P′的超程序。注意，ρ 也是 P的一次可行错误运行，因此 P 也不安全。

令 $\mathbb{T}$ 是 P 和 P′ 的通用结构，并且 H 是一个 $\mathbb{T}$-证明空间。注意，P 和 P′ 的错误运行集合相同。因此，L(H) 包含前者当且仅当它包含后者。□

对于任意一阶公式 φ，我们将 φ 的项高度定义为其中出现的项的最大高度。变量的项高度为 0。

引理 B.16。令 V 为有限词汇表，且 $\mathbb{T}$ 为一个 V-结构。若 $\vec{a}\in\mathbb{T}^k$ 生成了一个大小为 $n<\omega$ 的子结构，则存在一个项高度至多为 n 的无量词公式 $\chi[u_1,\dots,u_k]$，使得对于任意 V-结构 $\mathbb{T}'$ 和元组 $\vec{b}\in\mathbb{T}'^k$，都有 $\mathbb{T}'\models\chi(\vec{b})\iff\vec{a}\simeq\vec{b}$。

证明。令 $\mathbb{S}$ 为由 $\vec{a}$ 生成的子结构。对于每个 $b\in\mathbb{S}$，令 $t_b[\vec{u}]$ 为一个高度最小的 V- 项，使得 $t_b(\vec{a})=b$。令 C 由以下公式构成：

- $t_{a_i}=u_i$对每个 $i\in\{1,\dots,k\}$；
- $t_{b_1}\neq t_{b_2}$对所有互不相同的 $b_1,b_2\in\mathbb{S}$；
- 对所有关系符号 $R\in$V，并且 $b\in\mathbb{S}^r$当 $r=\text{ar}(R)$ 时，若$R(\vec{b})$为真，则 $R(t_{b_1},\dots,t_{b_r})$，否则为 $\neg R(t_{b_1},\dots,t_{b_r})$；并且
- $f(t_{b_1},\dots,t_{b_r})=t_c$对所有函数符号 $f\in$V，$\vec{b}\in\mathbb{S}^r$其中 $r=\text{ar}(f)$，且 $c=f(b_1,\dots,b_r)$。

由于 $\mathbb{S}$ 和 V 都是有限的，因此集合 C 是有限的。令 χ为 C 中所有公式的合取。由于 $\mathbb{S}$的大小为 n，每个 t_b的项高度至多为 $n-1$，由此可得 χ的项高度至多为 n。不难验证，这样选取 χ满足要求。□

命题 B.17（命题 6.7）。对于任意 $k\in\mathbb{N}_+$ 以及共同词汇表 V 上的拓扑族 T，若对任意 $\mathbb{T}\in$T 都有 $\text{sub}_k(\mathbb{T})\subset$T，则 $\bigcup_{\mathbb{T}\in\mathcal{T}}\text{sub}_k(\mathbb{T})$ 是 T 的一个秩为 k 的基。此外，若 V 有限，且存在某个 $n\in\mathbb{N}$，使得 $\bigcup_{\mathbb{T}\in\mathcal{T}}\text{sub}_k(\mathbb{T})$ 中每个结构的基数都不超过 n，那么这个基在同构意义下是有限的。

证明。设 $k\in\mathbb{N}_+$ 和 T 是共同词汇表上的一个拓扑族。假设对任意 $\mathbb{T}\in$T 都有 $\text{sub}_k(\mathbb{T})\subset$T。令 S $:=\bigcup_{\mathbb{T}\in\mathcal{T}}\text{sub}_k(\mathbb{T})$。则 S $\subset$T。

设 $\mathbb{T}\in$T 和 $\vec{a}\in\mathbb{T}^k$。设 $\mathbb{S}$ 为由 $\vec{a}$ 生成的 $\mathbb{T}$ 的子结构。注意，$\mathbb{S}\in$S 和 $\vec{a}\in\mathbb{S}^k$。因此，S 是 T 的一个基，其秩为 k。

假设 V 是有限的，并且存在某个 $n\in\mathbb{N}$，使得 $\bigcup_{\mathbb{T}\in\mathcal{T}}\text{sub}_k(\mathbb{T})$ 中每个结构的基数都不超过 n。只需证明，在 $\simeq$-等价意义下，T 中的 k-元组只有有限多个。

对于每个 $\mathbb{T}\in$T 和 $\vec{a}\in\mathbb{T}^k$，令 $\chi_{\mathbb{T},\vec{a}}$ 为根据引理 B.16 针对 $\mathbb{T}$ 和 $\vec{a}$ 得到的无量词公式。由于 V 是有限的，在逻辑等价 $\equiv$ 意义下，项高度不超过 n 的无量词公式只有有限多个。因此，商集 $\{\chi_{\mathbb{T},\vec{a}}\mid\mathbb{T}\in\text{T}\ \ \vec{a}\in\mathbb{T}^k\}/\equiv$ 是有限的。

对于任意 $\mathbb{T},\mathbb{T}'\in$T、$\vec{a}\in\mathbb{T}^k$ 和 $\vec{b}\in\mathbb{T}'^k$，若 $\chi_{\mathbb{T},\vec{a}}\equiv\chi_{\mathbb{T}',\vec{b}}$，则 $\mathbb{T}'\models\chi_{\mathbb{T},\vec{a}}(\vec{b})$，因此 $\vec{a}\simeq\vec{b}$。所以，$\{(\mathbb{T},\vec{a})\mid\mathbb{T}\in\text{T}\ \ \vec{a}\in\mathbb{T}^k\}/\simeq$ 也是有限的。□

定理 B.18 (定理 6.8)。任意具有通用结构 $\mathbb{T}$ 的安全参数化布尔程序 $\mathcal{P}$ 都有一个 $\mathbb{T}$-证明空间，该空间能够证明其安全性，且仅取决于 $\mathbb{T}$ 以及命令集及其语义。此外，若 $\mathbb{T}^2/\simeq$ 是有限的，则该证明空间是有限生成的。

记 P $=\{\text{P}_i\}_{i\in\mathcal{I}}$，并记 $\text{P}_i=P_i,[\![-]\!]_i$。令 $\mathbb{T}_i$为 P_i的拓扑。我们首先形式化如下陈述的含义：“每个带索引命令的（局部）语义都是确定性的，且只读取和写入有限个变量”：对于每个 $i\in$I 和 $a\in\mathbb{T}_i$，都存在一个有限集 $\text{rw}_i(a)\subset N(a)$，以及一个函数 $[\![-]\!]_{\text{rw}_i}$，它将每个 $\sigma:\ a\in\Sigma\times\mathbb{T}_i$ 映射为定义在赋值集合 $\text{rw}_i(a)\to\mathbb{D}$ 上的偏函数，使得

- 对于任意 $i,\ j\in$I、$a\in\mathbb{T}_i$以及 $b\in\mathbb{T}_j$从 a到 b的局部同构 β，我们有 $\beta(\text{rw}_i(a))=\text{rw}_i(b)$ 且 $[\![\sigma:a]\!]_{\text{rw}_i}=\beta^*[\![\sigma:a]\!]_{\text{rw}_i}$ 对每个 $\sigma\in\Sigma$均成立；并且
- $[\![-]\!]_i=\text{Global}_{N(a)}[\![-]\!]_{\text{rw}_i}$。

我们进一步假设，每个 $\text{rw}_i(a)$ 都由一个 V_{Node}-项列表表示。

我们先开始证明。

证明。对于任意 $a\in\mathbb{T}$，如果存在某个 $i\in$I、$a'\in\mathbb{T}_i$以及一个从 a到 a'的局部同构 β，则令 $\text{rw}(a):=\beta^{-1}(\text{rw}_i(a'))$ 且 $[\![\sigma:a]\!]_{\text{rw}}:=\beta^*[\![\sigma:a']\!]_{\text{rw}}$ 对每个 $\sigma\in\Sigma$均成立。

令 H 为由以下霍尔三元组生成的 $\mathbb{T}$证明空间：对于任意 $\sigma:\ a\in\Sigma\times\mathbb{T}$，使得 $a\simeq a'$ 对于某个 $a'\in\mathbb{T}_i$和 $i\in$I，

(1) 对于每个 $b\in\text{rw}(a)$、$x\in\mathbb{D}$ 和 $\mu\in[\![\sigma:\ a]\!]_{\text{rw}}^{-1}(\{\mu':\text{rw}(a)\to\mathbb{D}\mid\mu'(b)=x\})$，

$$\{\textstyle\bigwedge_{c\in\text{rw}(a)}\mathfrak{p}(c)=\mu(c)\}\ \langle\sigma:a\rangle\ \{\mathfrak{p}(b)=x\}$$

(2) 对于每个 $b\in\mathbb{T}\setminus\text{rw}(a)$，

$$\{\mathfrak{p}(b)=x\}\ \langle\sigma:a\rangle\ \{\mathfrak{p}(b)=x\}$$

(3) 对于每个 $\mu\in[\![\sigma:\ a]\!]_{\text{rw}}^{-1}(\varnothing)$，

$$\{\textstyle\bigwedge_{c\in\text{rw}(a)}\mathfrak{p}(c)=\mu(c)\}\ \langle\sigma:a\rangle\ \{\bot\}$$

(4) 如果 $[\![\sigma\colon\ a]\!]^{\neg\downarrow}_{\mathrm{rw}}(\varnothing) = \varnothing$,

$$\{\bot\}\ \langle\sigma : a\rangle\ \{\bot\}$$

显然，这些三元组都是有效的。另一方面，可以证明，对于任意运行 $\rho \in (\Sigma \times \mathbb{T}_i)^*$，如果将所有变量初始化为 0，该运行不可行，那么存在一个有限集 $U \subset \mathbb{T}_i$，使得

$$\{\textstyle\bigwedge_{c\in U} \mu(c) = 0\}\ \rho\ \{\bot\} \in \mathcal{H}.$$

根据 S-Param 规则，对于 (1) 和 (2)，只需从每个 $\simeq$-等价类的 $\mathbb{T}^2$ 中选取一个代表 $(a,\ b)$；对于 (3) 和 (4)，只需从每个 $\simeq$-等价类的 $\mathbb{T}$ 中选取一个代表 a。□

B.4 第 7 节的证明7

命题 B.19 （命题 7.1）。 设 $\mathcal{T}$ 是共同词汇表上的有限逻辑结构类，且 CL($\mathcal{T}$) 是 $\mathcal{T}$中成员的子结构。若 CL($\mathcal{T}$) 是一个可数 Fraïssé 类，那么它的 Fraïssé 极限就是定义 7.1所述的 $\mathcal{T}$的一个极限。

证明。假设 CL(T) 是一个可数 Fraïssé 类。令 $\mathbb{T}_\infty$ 为可数 Fraïssé 极限。那么，CL(T) 可以视为 $\mathbb{T}_\infty$ 的有限生成子结构类。因此，T $\subset$ CL(T) 中的任意结构都是 $\mathbb{T}_\infty$ 的嵌入子结构，而 $\mathbb{T}_\infty$ 的任意有限生成子结构都同构于 CL(T) 中的某个结构，即 T 中某个结构的子结构。□

定理 B.20 (定理 7.2)。 对于任意参数化程序$\mathcal{P}$ 以及拓扑 $_\infty$ 上的极限程序 $\mathbf{P}_\infty$（针对 $\mathcal{P}$），有： $\mathcal{P}$ 是安全的，当且仅当$\mathbf{P}_\infty$ 是安全的。此外，如果 $\mathbb{T}$ 是 $\mathcal{P}$ 的通用结构，且 $\mathbb{T}_\infty$ 可视为其子结构，那么任意 $\mathbb{T}$-证明空间 $\mathcal{H}$ 证明 $\mathcal{P}$ 的安全性，当且仅当它证明$\mathbf{P}_\infty$的安全性。

证明。令 P 为一个参数化程序，且 $\mathbf{P}_\infty$ 为 P 在 $\mathbb{T}_\infty$ 上的极限程序。P 中的任意错误运行都与 $\mathbf{P}_\infty$ 中的某个错误运行 $\simeq$-等价，且二者的可行性相同，反之亦然。因此，P 是安全的，当且仅当 $\mathbf{P}_\infty$ 是安全的。

假设 $\mathbb{T}$ 是 P 的一个通用结构，其中嵌入了 $\mathbb{T}_\infty$。设 H 是一个 $\mathbb{T}$-证明空间。注意，语言 L $\bigcup_{\in}$ $_{\mathrm{err}}$ P $_{\mathrm{err}}$ P_∞ $\simeq$ $_{\mathrm{err}}$ $[-]^{\simeq}$ (P) :$\mathcal{P}$ L ( ) 和 L ( ) 具有相同的 -闭包。令 表示 $\simeq$-闭包算子。那么

$$\begin{aligned}\mathcal{L}_{\mathrm{err}}(\mathcal{P}) \subset \mathcal{L}(\mathcal{H}) &\iff [\mathcal{L}_{\mathrm{err}}(\mathcal{P})]^{\simeq} \subset \mathcal{L}(\mathcal{H})\\ &\iff [\mathcal{L}_{\mathrm{err}}(P_\infty)]^{\simeq} \subset \mathcal{L}(\mathcal{H})\\ &\iff \mathcal{L}_{\mathrm{err}}(P_\infty) \subset \mathcal{L}(\mathcal{H}),\end{aligned}$$

其中第一个和最后一个等价关系由定理 5.2 得出。 □

定理 B.21 (定理 7.3)。 $\mathbb{T}$-PAL 在交和补运算下封闭，因此一元的 $\mathbb{T}$-PAL 也如此。任何 $\mathbb{T}$-PAL 都是局部对称的，即在等价关系 $\simeq$ 下封闭。

证明。交和补运算的构造与 $\mathbb{N}$-PA 的构造相同，参见命题 6.5]。

设 A:=Q $\Sigma \times \mathbb{T}$, δ, φ_{start}, F 和 A':=Q′, $\Sigma \times \mathbb{T}$, δ', $\varphi'_{\mathrm{start}}$, F, 是 $\mathbb{T}$-PA。假设不失一般性，Q 和 Q′ 不相交。自动机 Q $\cup$Q′, $\Sigma \times \mathbb{T}$, $\delta \cup \delta'$, $\varphi_{\mathrm{start}} \wedge \varphi'_{\mathrm{start}}$, $F \cup F$, 识别语言 L(A) $\cap$L(A')。

令 $\overline{\mathrm{Q}} := \{\overline{q} \mid q \in \mathrm{Q}\}$ 为 Q 的一个副本，并保留元数。对于任意公式 $\varphi \in \mathrm{F}_+(\mathrm{V},\ \mathrm{Q})$，令 φ 为通过以下方式得到的公式对 φ进行语法否定，将 $\neg$ 向内推，并将每个$\neg q$ 替换为 q。为识别语言 L(A)，我们构造自动机 A:=Q $\Sigma \times \mathbb{T}$, δ, φ_{start}, F, 其中接受符号$F := \{Q \mid \overline{Q} \in \overline{\mathrm{Q}} \setminus F\}$，转移函数定义为$\delta(q,\ \sigma)$:= $\delta(q,\ \sigma)$, 对每个 $q \in$Q 和 $\sigma \in \Sigma$ 均如此。

对于局部对称性，注意到对于任意 C $\xrightarrow{\sigma\ a}$:C′ 以及来自 a 的局部同构 β，我们有 β(C) $\sigma:\beta(a)$⟶ β(C′)，其中对于任意配置 C， β(C) 是通过将 $a \in \mathbb{T}$ 替换为 $\beta(a)$ 从 C 得到的。□

定理 B.22 (定理 7.4)。 设 $\mathbb{T}$ 是词汇表 $\mathcal{V}_{\mathrm{Node}}$上的一个拓扑。设 $\langle P,\ [\![-]\!]\rangle$ 是定义在 $\mathbb{T}$上的局部对称并发程序。如果 $\mathbb{T}/\simeq$ 是有限的，那么 $\mathcal{L}_{\mathrm{err}}(P)$ 是一个一元的 $\mathbb{T}$-PAL。

证明。该构造是 [21, 命题 6.3], 的推广，它仿照 P 的（反向）控制结构，但我们还确保每个出现的带索引命令 $\sigma\colon\ u$ 都满足 $\sigma \in P(u)$。

我们如下定义一个 $\mathbb{T}$-PA A_P:=Q $\Sigma \times \mathbb{T}$, δ, φ_{start}, F:

- Q := L $\cup \{I,\ err\}$, 其中每个 $l \in$ L 的元数为 1，而 I和 err是两个元数为 0 的特定符号。
- 转移定义如下，适用于任意 $\sigma \in \Sigma$ 和$l \in$L:

$$\delta(I, \sigma) := I \wedge \mathrm{src}(\sigma)(v_0) \wedge v_0 \in P^{-1}(\sigma),$$

$$\delta(l, \sigma) := \begin{cases} (v_1 = v_0 \wedge \mathrm{src}(\sigma)(v_0)) \vee (v_1 \neq v_0 \wedge l(v_1)), & \text{if } l = \mathrm{tgt}(\sigma) \\ v_1 \neq v_0 \wedge l(v_1), & \text{if } l \neq \mathrm{tgt}(\sigma) \end{cases}$$

$$\delta(err, \sigma) := \begin{cases} err, & \text{if } \mathrm{tgt}(\sigma) \neq l_{\mathrm{err}} \\ \top. & \text{if } \mathrm{tgt}(\sigma) = l_{\mathrm{err}} \end{cases}$$

由于 $P^{-1}(\sigma)$ 在局部对称性下封闭，因此它是 $\simeq$-等价类在 $\mathbb{T}$中的并。然后，根据假设$\mathbb{T}/\simeq$ 是有限的，以及命题 4.3， $P^{-1}(\sigma)$ 可由无量词公式 $\chi[v_0]$ 定义，我们将其简写为$v_0 \in P^{-1}(\sigma)$。

- $\overline{\varphi_{\text{开始}} := I \wedge \text{错误}}$。
- F:= $\{I, l_{\text{初始化}}\}$。

□

定理 B.23 (定理 7.5)。设 $\mathbb{T}$ 是词汇表$\mathcal{V}_{\mathrm{Node}}$上的一个拓扑。设 H 是由范式下的有限基生成的 $\mathbb{T}$-证明空间。若对每个 $\mathbb{T}^d/\simeq$ 均为有限，则$\mathcal{L}(\mathcal{H})$ 是一个$\mathbb{T}$-PAL。

证明。设 H 是 H 的范式有限基。该构造是对 [21, 命题 6 4], 的推广， H与其结构相对应： H 中的断言成为 $\mathbb{T}$-PA 的谓词，而霍尔三元组则成为转移。

我们定义一个 $\mathbb{T}$-PA A_H:=Q $\Sigma \times \mathbb{T}$, δ, φ_{start}, F如下：

- Q 由以下内容组成：对于出现在H中的每个断言 $\phi[\mu(a)]$，都有一个由 ϕ唯一确定的谓词符号 $[\phi]$ （在 [21]中称为规范名称），其元数与 ϕ的元数相同。请注意，由组合蕴涵识别出的合取项应拆分为不同的断言。

• 定义 δ 如下：对于每个霍尔三元组

$$\left\{\wedge_{i=1}^{n}\phi_i[\overrightarrow{\mathfrak{u}(a_i)}]\right\}\ \langle\sigma:b\rangle\ \left\{\psi[\overrightarrow{\mathfrak{u}(c)}]\right\}$$

在 H中，设

$$\delta([\psi],\sigma) := \bigwedge_{i=1}^{n}[\phi_i](\vec{t_i}) \wedge \chi(v_0,\ldots,v_{|\vec{c}|}),$$

其中，χ是一个无量词公式，用于定义 $b\vec{c}$的 $\simeq$-等价类（命题 4.3）；$\vec{t_i}$是由 V_{Node}-项组成的元组（t_i,1, . . . , t_i, m_i），满足对于每个 $i \in \{1,\ldots,n\}$和 $j \in \{1,\ldots,m_i\}$，$t_{i,j}$ $(b\vec{c}) = a_{i,j}$。如果$t_{i,j}$有多个选择，任选其一即可，因为所有与 $b\vec{c}$ 在 $\simeq$意义下等价的元组都会保持它们相等（命题 4.2）。如果 H 中有多个霍尔三元组具有相同的 $[\psi]$ 和 σ，我们就将它们的转移作析取合并。• $\varphi_{\text{start}} := [\bot]$。• $F := \varnothing$。

□

引理 B.24（引理 7.6）。设 $\mathbb{T}$ 是一个 V-结构，Q 是一个与 V[$\mathbb{T}$] 不相交的有限关系词汇表。假设配置集合 $2^{\mathcal{F}^{\mathbb{T}}_{\text{atom}(Q)}}$ 上存在一个预序 $\preceq$，使得对于任意 $\mathbb{T}$-谓词自动机 Q $\Sigma \times \mathbb{T}, \delta, \varphi_{\text{start}}, F$，有

（1） $\preceq$是可判定的。（2）（本质有限分支）给定任意配置 C，可以有效地计算出 C 的后继中的一个有限子集 $\mathscr{C}$，使得对于 C 的任意后继 C′，都存在某个 $\bar{}$C′ $\in \mathscr{C}$，满足 $\bar{}$C′ $\preceq$ C′。（3）（向下兼容性）对于任意配置 C 和 $\bar{}$C，若 $\bar{}$C $\preceq$ C，则如果 C 是接受的，那么 $\bar{}$C 也是接受的；并且对于任意满足 C $\rightarrow$ C′ 的配置 C′，都存在某个配置 $\bar{}$C′，使得 $\bar{}$C $\rightarrow$ $\bar{}$C′。

因此，存在一个用于判定 $\mathbb{T}$-PA 非空性的半算法；对于其可达配置在 $\preceq$ 下良拟序的 $\mathbb{T}$-PA 类，该半算法可转化为判定过程。

证明。半算法如下（假设 $\mathbb{T}$ 固定）。其中，过程调用 Repr-Succ(A, C) 会返回 C 的后继的一个有限子集，如本质有限分支条件所述：

1：过程 Check-Emptiness(A = Q $\Sigma, \delta, \varphi_{\text{start}}, F$) 2：Closed $\leftarrow \varnothing$3：Open $\leftarrow$ Dnf(φ_{start}) 4：当 Nonempty(Open) 时执行 5：C $\leftarrow$ Deqeue(Open) 6：如果 $\nexists$C′ $\in$ Closed. C′ $\preceq$ C 则 7：对所有 C′ $\in$ Repr-Succ(A,C) 执行 8：如果 C′ 是接受状态，则 9：返回 False 10：否则，如果 C′ $\notin$ Open 且 C′ $\notin$ Closed，则 11：Enqeue(Open, C′) 12：Closed $\leftarrow$ Closed $\cup${C} 13：返回 True

对于终止性，我们沿用 [32, 命题 5.3.15]证明中的论证。该算法的搜索空间构成一棵有限分支树。根据柯尼希引理，如果算法不终止，那么从根节点必然存在一条无限路径。假设 $\preceq$对可达配置构成良拟序。那么在这条无限路径上，存在一个配置 C 及其后代 C′，使得 C $\preceq$ C′。但是，当展开 C′ 时，C 必须已在 已关闭集合中，因此不应展开 C′。这一矛盾表明，如果 $\preceq$ 对可达配置构成良拟序，那么算法就会终止。□

定理 B.25(定理 7.7)。假设 $\mathbb{T}$ 满足以下条件：

(1) $\mathbb{T}$是齐次的。(2) $\mathbb{T}^d/\simeq$ 对每个 $d \in \mathbb{N}_+$都是有限的，并且存在一个算法，输入 d 后输出一组定义这些等价类的无量词公式（命题 4.3保证了这些等价类的存在）。(3) 任意无量词 $\mathcal{V}[\mathbb{T}]$-公式在 $\mathbb{T}$ 中的可满足性都是可判定的。

那么，覆盖关系满足引理 7.6中的条件 (1)–(3)。

证明。(1) 当 $\mathbb{T}$ 是齐次的时，如果局部对称关系是可判定的，那么覆盖关系也是可判定的。给定 $\vec{a}$，$\vec{b} \in \mathbb{T}^d$，要判定 $\vec{a} \simeq \vec{b}$，只需计算一组定义等价类的无量词公式 $\mathbb{T}^d/\simeq$，并检验 $\vec{a}$和 $\vec{b}$是否属于同一个等价类。

（2）下面是 Repr-Succ 的一种实现，其中

• Support(C) 返回 C 中按某种顺序列出的 $\mathbb{T}$ 元素顺序，

• Eq-Classes(d) 返回一个量词自由公式列表 这些公式定义了 $\mathbb{T}^d/\simeq$ 中的等价类，对于任意 $d \in \mathbb{N}_+$，以及

• Solve($p[\vec{u}]$) 会为任意 量词自由的 V[$\mathbb{T}$]-公式 p 返回一个满足赋值；若其不可满足，则返回 None，即不可满足。

尽管这里我们只假设可满足性的可判定性，可满足性，在第 7 节中，我们有一个全局假设：元素 $\mathbb{T}$ 可以有限表示（并且可递归枚举）；二者共同表明 Solve 是可计算的。

1: **过程** Repr-Succ($\langle Q\ \Sigma, \delta, \varphi_{\text{start}}, F\rangle, C$)
2: $\mathscr{C} \leftarrow${}
3: $\vec{a} \leftarrow$Support(C)
4: $d \leftarrow |\vec{a}|$
5: $X \leftarrow$ Eq-Classes($\mathbb{T}, d+1$)
6: **对所有** $\chi[u_1,\ldots,u_{d+1}] \in X$ **执行**
7: $b \leftarrow$ Solve($\chi[u_1, \vec{a}]$)
8: **如果** $b \neq$ 无**则**
9: **对所有** $\sigma \in \Sigma$ **执行**
10: **对所有** C' 使得 $C \xrightarrow{\sigma\, b} : C'$ **执行**
11: $\mathscr{C} \leftarrow \mathscr{C} \cup${C′}
12: **返回** $\mathscr{C}$

（3）向下兼容性很容易验证。

□

注 2. 在反例引导的抽象精化循环中，我们也需要一个用于证明非空性的见证。可以修改上述 Repr-Succ 的实现，使其也能

返回导致每个 C′ ∈ $\mathscr{C}$ 的字母 σ : b。然后，可以相应地修改 Check-Emptiness 空性检查过程，使其返回一个见证而不是 False。

推论 B.26 (推论 7.9**)**。 固定一个满足定理 $\mathbb{T}$ 条件的结构。定理7.7。此外，假设对于任意与 V 不相交、且符号元数至多为 1 的有限关系词汇表 Q，在配置集合 $2^{\mathscr{F}^{\mathbb{T}}_{\mathrm{atom}(Q)}}$ 上，覆盖预序是良拟序。那么，任何具有极限程序的参数化布尔程序的安全性都是可判定的，其中极限程序定义在 $\mathbb{T}$ 上。

证明。 令 $\mathcal{P}$ 为一个具有极限程序 $\mathbf{P}_\infty$（定义在 $\mathbb{T}$ 上）的参数化布尔程序。令 H 为定理 -证明空间 $\mathcal{H}$ 中构造出的一个有限基，该空间由定理 6.8构造。于是 $\mathcal{P}$ 安全，当且仅当 $\mathcal{H}$证明其安全性；而这又当且仅当$\mathcal{H}$ 证明 $\mathbf{P}_\infty$的安全性，其中它是 $\mathcal{P}$ 定义在 $\mathbb{T}$ 上的极限程序（定理 7.2，稍作修改以考虑运行邻域中节点的初始化）。

请注意，定理 7.7 中的条件允许进行以下操作：

- 枚举 $\mathbb{T}^d$ 的 $\simeq$-等价类（任意 $d \in \mathbb{N}_+$），每个等价类均由一个无量词公式表示。
- 找到满足某个无量词公式的具体元组（使用定理 Solve 证明中的 7.7 过程）。
- 找到一个无量词公式，用于定义给定元组的 $\simeq$-等价类。

我们的算法如下：
(1) 计算基 H。（定理 6.8）我们枚举 $\mathbb{T}$ 和 $\mathbb{T}^2$的 $\simeq$-等价类的代表元。假设存在一个可计算函数，能将极限结构中的每个节点映射到 P 的拓扑中的一个局部对称节点；又由于每个 $\mathrm{rw}_i(a)$ 都由一个 $\mathrm{V}_{\mathrm{Node}}$-项列表表示，因此函数 rw 和 $[\![-]\!]_{\mathrm{rw}}$ 均可计算。(2) 构造用于 L(H) 的 $\mathbb{T}$-PA A_H。（定理 7.5 ）注意， H 已经处于范式。我们将规范名称 $[x = 0]$ 加入接受符号集合 F，以处理初始化。(3) 构造用于 L(P_∞) 的 $\mathbb{T}$-PA A_{P_∞}。（定理 7.4）对于每个 $\sigma \in \Sigma$，定义 $P^{-1}(\sigma)$ 的公式都是可计算的：它是表示 $\mathbb{T}/\simeq$的无量词公式的某个子集的析取式。我们枚举每个等价类的代表元 a，并在 $\sigma \in P(a)$时加入对应于 a 的公式。(4) 检查 L(A_{P_∞}) ∩ L(A_H) 是否为空。（定理 7.3 和 7.7 ）若为空，则返回 Safe ； 否则，返回 Unsafe。根据假设，该语言由一元的 $\mathbb{T}$-PA 识别，因此空性检查会终止。

□