arXiv:2601.13500v1 [cs.GT] 20 Jan 2026

并发宽松策略模板*

Ashwani Anand[1],Christel Baier[2],Calvin Chau[2],Sascha Klüppelholz[2], Ali Mirzaei**、SatyaPrakash Nayak[1], 和 Anne-Kathrin Schmuck[1]

[1] 马克斯·普朗克软件系统研究所，凯撒斯劳滕，德国
{ashwani,sanayak,akschmuck}@mpi-sws.org[2] 德累斯顿工业大学，德国
{christel.baier,calvin.chau,sasha.klueppelholz}@tu-dresden.de

摘要 有限图上的双人博弈为建模反应式系统与其环境之间的策略交互提供了严格基础。尽管并发博弈语义能够自然地刻画许多信息物理系统（CPS）所特有的同步交互，但其在 CPS设计中的应用仍然有限。在轮流制博弈的宽容策略模板（PeSTels）概念基础上，我们引入并发（宽容）策略模板（ConSTels）——一种用于表示具有安全性、Büchi 和余 Büchi 目标的并发博弈中随机化获胜策略集合的新颖表示方法。ConSTels 能够紧凑地编码无穷多个策略，从而同时支持离线和在线适应。在离线场景中，我们利用组合性实现增量综合：将针对更简单目标的 ConSTels 组合为针对更复杂组合目标的无冲突模板。在在线场景中，我们展示了 ConSTels 如何促进运行时适应：根据观测到的对手行为调整动作概率，在保持正确性的同时优化性能。我们在一个原型工具中实现了 ConSTel 综合与适应，并通过实验展示了其潜力。

关键词： 并发博弈 · 宽松策略 · 策略适配。

1 引言

有限图上的双人博弈 [17] 为建模反应式系统与其环境之间的策略交互提供了强大的抽象。

* 作者按字母顺序排列。所有作者均获得德国研究联合会项目 389792660 TRR 248-CPEC 的支持。此外，A.-K. Schmuck 获得德国研究联合会项目 SCHM 3541/1-1 的支持。C. Baier、C. Chau 和 S. Klüppelholz 获得德国联邦研究、技术与航天部通过 SE-MECO Q1 项目（03ZU1210AG）提供的支持；该项目由德国研究联合会（DFG, Deutsche Forschungsgemeinschaft）资助，属于德国卓越战略的一部分——EXC 2050/2，项目编号 390696704——德累斯顿工业大学“人在回路触觉互联网中心”（CeTI）卓越集群。** 本研究是在 Ali Mirzaei 于德国 MPI-SWS 实习期间开展的（ali.mirzaei78@sharif.edu）。

因此，这类博弈已成功应用于软件 [20,18] 以及信息物理系统（或组件）的设计 [32,23,35] ，以确保交互符合严格的正确性要求。这些要求通常是时序性的——用于描述进程随时间推移的正确交互——并且可以用时序逻辑形式化，或者更一般地，表示为一种 ω-正则语言。

根据系统（或组件）之间交互方式的不同，所得博弈中玩家的交互语义也有所不同。异步交互会产生回合制 博弈 [30,15]，在这种博弈中，每个回合只有一名玩家可以从多个行动中进行选择。另一方面，同步交互会产生并发 博弈 [2,1,13]，在这种博弈中，每个回合两名玩家都可以同时且独立地从多个行动中进行选择。

事实上，反应式系统中的许多策略交互天然就是同步的——尤其是在信息物理系统设计的背景下。多个机器人会同时移动到状态空间中的不同区域，而不会等待彼此完成移动。类似地，在非结构化环境中，人机协作（例如清空洗碗机）只有在双方并肩工作（同步进行）时才真正能为人类提供帮助，而不会被迫轮流行动（如在制造流水线等结构化环境中）。最后，物理条件通常也会随系统动作同步变化，例如无人机着陆时，风况可能变得很恶劣，而 湍流并不会“等待”无人机着陆后才出现。

尽管技术系统与其环境之间存在这种自然的同步策略交互，几乎所有现有工作——尤其是在CPS的策略控制背景下——都侧重于使用轮流制博弈语义来抽象系统交互（参见 [23,35] 以了解概述）。传统方法通常将基于博弈的综合作为CPS设计中的黑盒，而近期的一项研究引入了宽容策略模板（PeSTels） [6,7,8] ，作为一种新的简洁数据结构，用于捕获轮流制图博弈中的无限多个获胜策略。这种宽容性使策略能够在离线情况下进行适配，例如用于增量式或分布式综合 [7,9,27] ，也能在在线情况下进行适配，例如处理执行器故障或实现CPS的动态多层控制[26] 。虽然文献 [29]将模板扩展到了随机奇偶博弈中的几乎必然获胜，但并未考虑并发设置。

并发博弈最早由Shapley [31]提出，此后一直受到研究 [16,2,1,13]。有关详细的历史介绍，请参见[17]。特别是，文献 [1]提供了求解并发ω-正则博弈的算法。研究表明，与轮流制博弈不同，并发博弈可能要求玩家对其策略选择进行随机化，才能获胜。这使得相应的求解算法明显比现有的轮流制博弈算法更加复杂。

尽管并发博弈在理论上已得到充分研究，但针对 ω-正则目标的工具支持仍然有限。Mocha 工具 [5]将并发系统视为反应式模块 [3] ，并支持交替时序逻辑（ATL），但似乎已不再提供。Gavs+ 平台 [14] 仅处理并发可达性博弈。Prism-games [24] 支持并发

随机博弈，但不支持 Büchi 或余 Büchi 条件。Praline [12] 可以检查是否存在纯（即不进行随机化的）策略纳什均衡。Mcmas[25]和 Eve [19] 也可以检查纯纳什均衡。对更丰富的性质类别（如 ω-正则目标）的支持有限，也进一步阻碍了将并发博弈用作 CPS 设计抽象的做法，尽管它们在语义上天然契合。

贡献。 基于 PeSTels 在信息物理系统设计中的成功，以及对其大多数交互都是同步的这一观察，本文针对安全性、Büchi 和余 Büchi 目标开发了并发策略模板 （ConSTels）。我们还介绍了原型工具 ConSTel，它可以自动为这些博弈计算 ConSTels。我们利用 ConSTels 相较于（经典）并发博弈中的获胜策略在策略离线 和在线 适配方面的优势。

（i）离线适配： 我们对增量综合进行了形式化，即将针对单个 Büchi 或余 Büchi 目标计算出的 ConSTels 组合起来，为组合目标生成新的 ConSTel；该组合目标的复杂度严格高于任一单独目标。若生成的 ConSTels 彼此不冲突，我们的方法便能够为这些多维目标计算获胜策略。尽管这种方法必然是不完备的，我们通过实验表明，它在若干改编自 SYNTCOMP基准测试套件 [21]的示例中均取得了成功（见图2）。这一点尤其值得关注，因为目前还没有其他工具能够支撑并发奇偶博弈。

(ii) 在线适应： 我们通过实验展示了这些模板在运行时优化方面的潜力。由于 ConSTels 能够涵盖无限多种随机化策略，我们可以在运行时调整特定状态下动作的概率分布。对于动作分布固定但未知的对手，ConSTels 能够在运行时推断该分布，并相应地调整策略。这有助于高效地满足目标（见图 3）。虽然该实验仅用于概念验证，但我们指出，它展现了 ConSTels 将矩阵博弈 [28] 集成到每个状态中的潜力，以解决战略性长期博弈中智能体之间动作概率的局部依赖关系。我们将这一探索留待未来工作。

尽管关于机器人规划 [33],中的并发博弈、多智能体学习 [11] 以及通过 ATL [4,5], 在多智能体系统[34] 中进行集体策略综合的研究有限，但据我们所知，我们提出了首个受 CPS 应用启发、针对并发博弈的宽松策略方法。

2 预备知识

本节回顾并发博弈的所有必要预备知识，内容摘自 [1]，以及策略模板的相关内容，摘自 [7]。

记号。 对于有限字母表 Σ，我们用 Σ^*、 Σ^+和 Σ^ω 分别表示 Σ 上所有有限词、非空有限词和无限词的集合。我们定义$\Sigma^\infty := \Sigma^* \cup \Sigma^\omega$ ，并且对于任意词 $\omega \in \Sigma^\infty$， ω_i 表示 ω中的第 i个符号，且 $\omega_{\leq i} := \omega_1\omega_2 \ldots \omega_i$。 有限集 A 上的概率分布 是一个函数

$p\colon A \to [0,1]$ 使得 $\sum_{a\in A} p(a) = 1$。 A 上的所有概率分布构成的集合记为 $\mathcal{D}(A)$。集合 A 上概率分布 p 的支撑集 定义为 $\mathsf{supp}(p) := \{a \in A \mid p(a) > 0\}$。对于 $B \subseteq A$，我们定义 $p(B) := \sum_{b\in B} p(b)$。

并发博弈图。双人并发博弈图是一个元组$G = (V, \Gamma_1, \Gamma_2, \delta)$，其中 V是有限状态集， Γ_1和 Γ_2分别是玩家1和玩家2的有限动作集，且 $\delta\colon V \times \Gamma_1 \times \Gamma_2 \to V$是一个转移函数。在每个状态 $v \in V$，我们用 $\Gamma_i(v) \subseteq \Gamma_i$ 表示玩家i 的所有可能动作；玩家1选择一个动作 $a \in \Gamma_1(v)$，同时玩家2独立地选择一个动作 $b \in \Gamma_2(v)$。随后，博弈转移到状态 $v' \in V$，使得 $\delta(v,a,b) = v'$。从状态 v_0开始的博弈过程是一个有限或无限状态序列 $\rho \in V^\infty$ ，满足 $\rho_0 = v_0$ ；并且对于每个 $0 \le i \le |\rho|$，都存在动作 $a_i \in \Gamma_1(\rho_i)$和 $b_i \in \Gamma_2(\rho_i)$，使得 $\delta(\rho_i, a_i, b_i) = \rho_{i+1}$。我们用 $\mathsf{inf}(\rho)$ 表示在 ρ中无限次出现的所有状态组成的集合。

策略。对于并发博弈图 G，玩家 $i \in \{1,2\}$ 的（随机化）策略是一个函数 $\pi_i\colon V^+ \to \mathcal{D}(\Gamma_i)$，满足对于所有 $\rho v \in V^+$，均有 $\mathsf{supp}(\pi_i(\rho v)) \subseteq \Gamma_i(v)$。我们将 G 上玩家 i 的所有策略收集在集合 Π_i 中。若对于每个对局序列 ρ，分布 $\pi(\rho)$ 都以概率 1 指定单个动作，则策略 π 是确定性的（或纯策略）；否则就是随机化的。若对于所有 ρv, $\rho' v \in V^+$，均有 $\pi(\rho v) = \pi(\rho' v)$，则策略 π 是无记忆的。无限对局序列 $\rho \in V^\omega$ 与玩家1的策略 π_1 相容，当且仅当对于每个 $i \in \mathbb{N}$，都存在动作 $a \in \Gamma_1(\rho_i)$ 和 $b \in \Gamma_2(\rho_i)$，使得 $\delta(\rho_i, a, b) = \rho_{i+1}$ 且 $a \in \mathsf{supp}(\pi_1(\rho_{\le i}))$，即 $\pi_1(\rho_{\le i})(a) > 0$。玩家2的策略定义类似。我们将与 π_i 相容的对局序列以及同时与 π_1 和 π_2 相容的对局序列分别称为 π_i-对局序列和 $\pi_1\pi_2$-对局序列，并将所有 π_i-对局序列收集在集合 $\mathsf{plays}(\pi)$ 中。

获胜条件。我们考虑无限对局上的安全性、Büchi 和余 Büchi 获胜条件。给定安全状态集合 $I \subseteq V$，若对于所有 $i \in \mathbb{N}$ 均成立$\rho(i) \in I$，则在并发安全博弈 $(G, \Box I)$ 中，$\rho \in V^\omega$ 上的无限对局 G 是获胜的。给定 Büchi 状态集合 $T \subseteq V$，若对于所有 $i \in \mathbb{N}$ 均存在 $j \ge i$ 使得 $\rho(j) \in T$，则在并发 Büchi 博弈 $(G, \Box\Diamond T)$ 中， $\rho \in V^\omega$ 上的无限对局 G 是获胜的。给定满足 $T = V \setminus T$ 的余 Büchi 状态集合 $T \subset V$，若存在 $i \in \mathbb{N}$ 使得对于所有 $j \ge i$ 均成立 $\rho(j) \notin T$，则在并发余 Büchi 博弈 $(G, \Diamond\Box T)$ 中，$\rho \in V^\omega$ 上的无限对局 G 是获胜的。我们用 (G, Φ)（其中 $\Phi \in \{\Box I, \Box\Diamond I, \Diamond\Box I\}$ 满足 $I \subseteq V$）表示任意并发博弈，并用 $\mathcal{L}(\Phi)$ 表示相应的获胜对局集合。

几乎必然获胜。 我们研究并发博弈中的几乎必然获胜。若策略 π 使玩家1从 v 在 (G, Φ) 中获胜的概率为 1，则称其为几乎必然获胜 （简称获胜）。其中，$\Pr_v^\pi(\mathcal{L}(\Phi)) = 1$表示从 v 开始的 π_i-博弈过程属于 X的概率。并发博弈 (G, Φ) 中玩家1的几乎必然获胜区域 （简称获胜区域）是集合 $\mathcal{W}_i(\Phi) \subseteq V$，其中包含玩家1能够通过某个策略获胜的所有状态。若策略 π 使玩家1从每个 $v \in \mathcal{W}_i(\Phi)$ 出发都获胜，则称其为 (G, Φ) 中玩家1的获胜策略 。

策略模板。 形式上，策略模板 Λ 是玩家1策略的一个（可能无限的）集合，即 $\Lambda \subseteq \Pi_1$。接着，如果 $\pi_1 \in \Lambda$，则称玩家1策略 π_1 遵循 模板 Λ。如果在博弈 $(G,\ \Phi)$ 中，所有遵循模板 Λ 的玩家1策略都

从 v 开始在 $(G,\ \Phi)$ 中获胜。此外，如果策略模板 Λ 从 $\mathcal{W}_1$ 中的每个状态出发都能获胜，则称其获胜。另外，如果每个在 $\mathcal{G}$ 中获胜的玩家1策略 π 也遵循 Λ，则称 Λ 对最大宽松。

3 概述与问题陈述

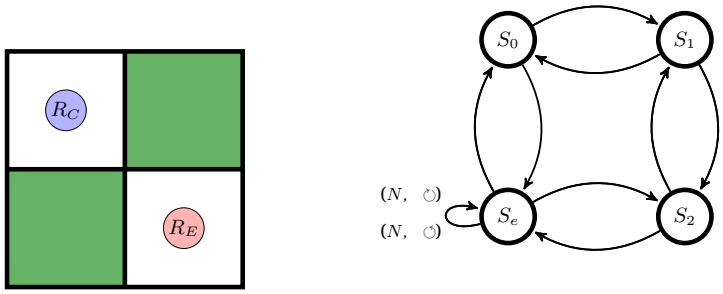
b正如上一节形式化所示，获胜策略模板凭借其作为策略集合的简单定义，可直接从轮流博弈推广到并发博弈。然而，与经典获胜策略相比，它们的主要优势在于采用了一种简单且局部的数据结构，能够简洁地描述这些策略集合。具体而言，对于轮流（奇偶）博弈，文献 [7]证明，获胜策略模板可以定义为以下三类简单的玩家1边条件的合取：(i) 不安全边：玩家1绝不应选择的边；(ii) 共生存边：玩家1在一次博弈过程中只能选择有限次的边；以及 (iii) 生存边组：边的集合，若某些生存边的源状态被无限次访问，玩家1就必须无限次地从该集合中选择至少一条边。具体来说，这些条件可分别在求解安全性、Büchi 和 co-Büchi 博弈（即计算获胜区域）的过程中，通过仔细记录信息提取出来。

本文的主要贡献是自动提取一种同样简洁的数据结构，用于表示并发安全性、Büchi 和余 Büchi 博弈中的一组宽容获胜策略。由于同步博弈语义、对随机化策略的需求，以及并发博弈中显著不同的求解算法，这些边界条件的形式化与计算方式与基于回合的情形有很大不同。

我们通过一个示例概述并发策略模板的一般构造，以说明其相较于经典策略的优势。此外，我们还利用该示例概述本文其余部分的结构；本文将对示例中预先介绍的所有概念进行形式化。

将战略互动建模为并发博弈。我们考虑两个机器人——一个受控机器人 R_C 和一个环境机器人 R_E ——它们位于图 1a 所示的非常简单的网格世界中，其中 R_C 必须独自进入绿色区域 $alone$，才能完成特定任务。两个机器人都可以沿顺时针或逆时针方向穿过网格移动。

这种策略互动可以建模为 R_C玩家1与 R_E 玩家2在图1b所示的博弈图 G上进行的并发博弈。该博弈有四个状态，用于区分工作空间中机器人位置的所有相关配置。初始状态 S_0 表示两个机器人分别位于网格对角相对的白色角落。状态 S_1 （相应地，S_2）表示两个机器人共同位于绿色单元格（相应地，白色单元格）中的配置。最后，逃逸状态 S_e 表示 R_C 位于绿色单元格而 R_E 位于白色单元格的配置。在每个状态下，两个玩家都可以选择顺时针（↻）或逆时针（↺）移动到相邻单元格。为简化所得博弈，我们在此仅作示意性假设，在



(a) 网格世界 (b) 并发博弈图

图1：示例说明。左：两个机器人， R_C 和 R_E，在网格世界中移动。 R_C 必须单独进入绿色区域 才能完成任务。两个机器人都可以顺时针（$\circlearrowleft$）或逆时针（$\circlearrowleft$）移动。右：并发博弈图，刻画了 R_C 和 R_E的同步策略交互。

逃逸状态 S_e下，玩家1（机器人 R_C）可以选择停留（N）在该单元格中；无论玩家2采取何种动作，机器人都会保持在 S_e中。

安全模板。 首先考虑一个非常简单的安全博弈：两台机器人都从 S_e 开始，而 R_C 必须确保博弈过程始终停留在 S_e 中。此时，获胜区域为 $\mathcal{W}_1(\square S_e) = \{S_e\}$，并且 R_C 要想从状态 $\mathcal{W}_1$ 获胜，就绝不能选择集合 $\{\circlearrowleft, \circlearrowleft\}$ 中的动作。我们用一个安全模板对此进行建模：该模板允许所有不会对安全函数 $\mathsf{S}(S_e) = \{\circlearrowleft, \circlearrowleft\}$ 指定的动作赋予正概率的策略。详情请参见第 4.2 节。

活跃组模板。 现在考虑布奇博弈（$G, \square\Diamond\{S_e\}$）。请注意，在这个博弈中，机器人 R_C 无法通过在每个状态都确定性地选择行动，从初始配置S_0 获胜。对于 R_C 在状态 $\{S_0, S_2\}$中的每一种确定性选择， R_E 都可以选择一个阻止到达S_e的动作。因此， R_C 必须在状态$\{S_0, S_2\}$ 中以正概率选择两个动作，以确保无论 R_E采用何种策略，博弈都能以概率 1 到达 S_e。然而，在这种情况下，我们有 $\mathcal{W}_1(\square\Diamond S_e) = V$；在这个博弈中， R_C 的获胜策略模板会要求在每个状态 $\{S_0, S_2\}$ 中都以正概率选择两个行动 $\{\circlearrowleft, \circlearrowleft\}$，而在状态 $\{S_e, S_1\}$ 中，该策略模板则不会限制 R_C 的行动，从而保持其宽松性。

形式化地，我们将这一策略模板编码为活跃组模板 $\Lambda_1 = \Lambda_{\text{LIVE}}(\mathsf{H}, \mathsf{P})$，其中 H 是一个函数，将每个状态映射到行动集合的集合，并要求为每个行动集合分配正概率；P 则允许从不同状态设置多个活性约束，以确保宽容性。在本例中，基于上述直观理解，我们有 $\mathsf{H}(S_0) = \mathsf{H}(S_2) = \{\{\circlearrowleft\}, \{\circlearrowleft\}\}$，以及 $\mathsf{H}(S_e) = \{\Gamma_1(S_e)\}$、$\mathsf{H}(S_1) = \{\Gamma_1(S_1)\}$。然而，实际上我们不需要同时满足来自 S_1 和 S_2 的活性约束，而只需满足其中一个状态的活性约束（以及状态 S_1 的活性约束），即可保证到达状态 S_e。这是通过将 S_1 和 S_2 合并到同一个活跃组中实现的，

从而得到 P $= \{\{S_0, S_2\}, \{S_1\}\}$。所得的活跃组模板只要求后续策略为 H($S_1$) 或 H($S_2$) 之一分配正概率，即可成为 R_C 的获胜策略。在一般情况下，当 $\mathcal{W}_1(\square\Diamond S_e) \subsetneq V$ 时，我们还需要一个安全模板，以防止策略离开 $\mathcal{W}_1$，因为 Büchi 博弈的活跃组模板只定义在获胜状态上。我们将在第 4.3 节中形式化讨论活跃组模板。

共活模板。 现在，我们考虑一个余 Büchi 博弈 $(G, \Diamond\square\{S_1, S_2, S_e\})$。我们观察到， R_C 无法确保从状态$\{S_1, S_2\}$出发时不访问初始状态 S_0 。因此， R_C 要赢得这场博弈，唯一的办法是确保 (i) 始终最终到达 S_e ，并且 (ii) 从某个时刻开始始终停留在逃逸状态 S_e。因此，宽容模板由两部分组成。为实现 (i)，需要使用为 Büchi 博弈 $(G, \square\Diamond\{S_e\})$ 计算得出的活组模板 Λ_{LIVE}(H,P)。为实现 (ii)，在状态 S_e 内，从某个时刻开始，获胜策略必须将动作 $\{\circlearrowleft, \circlearrowleft\}$ 的概率设为 0。这可以通过共活函数编码为共活性模板：该函数将状态映射到一组动作，这些动作的累积概率必须受到时间上的约束，从而最终不再被采取。在此例中，C(S_e) $= \{\{\circlearrowleft\}, \{\circlearrowleft\}\}$，而对于每个其他状态 s，C(s) $= \emptyset$。同样，如果获胜区域未覆盖所有状态，则还需要一个额外的安全模板。此外，一般而言，余 Büchi 博弈可能需要更复杂地组合共活模板和活组模板，从而导致提取算法更为复杂。我们将在第 4.4 节正式讨论共活模板。

离线适应性。 ConSTels 的宽松性所带来的一个主要优势，是其诱导出的离线适应性。例如，考虑一个同时包含上述 Büchi 和 co-Büchi 制胜条件的博弈，即$\Phi = \square\Diamond\{S_e\} \wedge \Diamond\square\{S_1, S_2, S_e\}$。在这种情况下，我们可以通过取两个策略模板约束的交集来组合它们；在本例中，该交集恰好与 co-Büchi 目标的模板一致。一般而言，如果模板之间不存在冲突，例如某个状态中唯一剩余的活跃动作集包含一个 co-live 动作，则这种组合将为组合目标导出一个获胜策略模板。我们将在第 5 节至第 6 节中将这一概念形式化。我们在第 7 节中通过实证表明，这类冲突在实践中很少见，因此在大量情况下都可以进行组合。

在线适应性。 由于策略模板具有宽松的结构，因此可以利用运行时可用的额外信息来优化机器人的行为。为此，假设 R_E 在博弈中采用一种随机策略，该策略对动作使用固定的概率分布。虽然 R_C不知道这一概率分布，但 R_C 可以在运行时观察 R_E 选择的动作。因此， R_C 可以“估计” R_E的动作分布，并在线调整自身的动作分布，从而在遵循策略模板的同时优化动作选择，以高效地到达 S_e 。我们已在图 1a 所示的机器人示例中，通过原型工具实现了这一想法，实验结果如图 3 所示。可以看到，调整策略后， R_C 到达 S_e 的速度快得多。

4 获胜策略模板的计算

本节介绍本文的主要贡献，即对并发安全模板（第 4.2 节）、活组模板（第 4.3 节）和共活模板（第 4.4 节）进行形式化并加以构造。与为轮流制博弈构造这些模板的方法类似，我们分别从并发安全博弈、Büchi 博弈和 co-Büchi 博弈的获胜区域的符号计算中提取这些模板。为形式化这一提取过程，我们在第 4.1 节补充介绍了预备知识。

4.1 预备知识

鉴于篇幅限制，关于 [22] -演算的介绍，请读者参阅 [22]；关于其在模板计算中的具体应用，请参阅 [6]。为确保内容具有最低限度的自包含性，本节回顾用于定义 [1]中并发安全性、Büchi 和 co-Büchi 博弈的符号固定点算法所需的集合变换器。

给定一个并发博弈 (G, Φ)、一个状态 $v \in V$ 以及玩家 $i \in \{1,2\}$的转移分布$d_i \in \mathcal{D}(\Gamma_i(v))$，从 v 到达集合$X \subseteq V$的单轮概率定义为

$$\Pr_v^{d_1,d_2}(X) := \sum_{a\in\Gamma_1(v)} \sum_{b\in\Gamma_2(v)} \sum_{\{v'\in X \mid \delta(v,a,b)=v'\}} d_1(a)d_2(b).$$

接下来，我们定义三种类型的前驱算子:

– $\mathsf{pre}_1 : 2^V \to 2^V$ 使得 $v \in \mathsf{pre}_1(X)$ 若

$$\exists d_1 \in \mathcal{D}(\Gamma_1(v)).\forall d_2 \in \mathcal{D}(\Gamma_2(v)).\, \Pr_v^{d_1,d_2}(X) = 1.$$

– $\mathsf{Apre}_1 : 2^V \times 2^V \to 2^V$ s.t. $v \in \mathsf{Apre}_1(Y, X)$ if

$$\exists d_1 \in \mathcal{D}(\Gamma_1(v)).\forall d_2 \in \mathcal{D}(\Gamma_2(v)).\, \left[\Pr_v^{d_1,d_2}(Y) = 1 \wedge \Pr_v^{d_1,d_2}(X) > 0\right]$$

– AF 前驱$_1$: $2^V \times 2^V \times 2^V \to 2^V$使得 $v \in \mathsf{AFpre}_1(Z, Y, X)$ 若存在某个 $\beta > 0$,

$$\exists d_1 \in \mathcal{D}(\Gamma_1(v)).\forall d_2 \in \mathcal{D}(\Gamma_2(v)).\, \begin{pmatrix} \Pr_v^{d_1,d_2}(Z) = 1 \\ \wedge \Pr_v^{d_1,d_2}(X) \geq \beta \Pr_v^{d_1,d_2}(\neg Y) \geq 0 \end{pmatrix}$$

前驱算子 $\mathsf{pre}_1(X)$ 计算出这样一组状态：从这些状态出发，玩家1可以确保以概率 1 到达 X。前驱算子 $\mathsf{Apre}_1(Y, X)$ 计算出这样一组状态：从这些状态出发，玩家1可以确保几乎必然停留在Y，并以正概率到达 X。最后，前驱算子 $\mathsf{AFpre}_1(Z, Y, X)$ 计算出这样一组状态：从这些状态出发，玩家1可以确保几乎必然停留在 Z；如果它可以离开 Y的概率为正，那么它也应以某个正概率到达 X。

此外，我们还使用了 [1, Sec.6], 中的两个动作集函数，即 $A_Y^v : 2^{\Gamma_2(v)} \to 2^{\Gamma_1(v)}$ 和 $B_X^v : 2^{\Gamma_1(v)} \to 2^{\Gamma_2(v)}$，其定义满足

$$\mathsf{A}_Y^v(\gamma_2) := \{a \in \Gamma_1(v) \mid \forall b \in \Gamma_2(v)\ .\ \delta(v,a,b) \notin Y \Rightarrow b \in \gamma_2\}, \tag{1}$$

$$\mathsf{B}_X^v(\gamma_1) := \{b \in \Gamma_2(v) \mid \exists a \in \gamma_1 \cdot \delta(v,a,b) \in X\} \tag{2}$$

直观而言，集合 $\mathrm{A}_Y^v(\gamma_2)$ 表示状态 v 下玩家1的动作集合；在玩家2不采取 γ_2 中动作的前提下，这些动作保证能够到达 $Y \subseteq V$ 中的某个状态。集合 $\mathrm{B}_X^v(\gamma_1)$ 表示状态 v 下玩家2的动作集合；对于这些动作，玩家1能够确保以正概率到达 $X \subseteq V$ 中的某个状态。由 [1,Sec.6] 可知，AF 前驱可以通过等价关系 $v \in \mathrm{AFpre}_1(Z, Y, X) \iff \nu\gamma.(\mathrm{A}_Z^v(\emptyset) \cap \mathrm{A}_Y^v(\mathrm{B}_X^v(\gamma))) \neq \emptyset$ 计算得出。

4.2 并发安全模板

本节介绍并发博弈的（最大宽松）安全模板，这与轮流制博弈的安全模板直接类似。

模板形式化。如第 3 节所示，与轮流制博弈中的对应模板类似，并发安全模板通过函数 $\mathsf{S}: V \to 2^{\Gamma_1(V)}$ 定义，该函数为玩家1在每个状态指定一组不安全动作，并收集所有避开 S 中动作的策略。也就是说，其支撑集[1] 从不包含不安全动作，具体形式化如下。

定义 1。设 G 为一个并发博弈图。给定函数 $S: V \to 2^{\Gamma_1(V)}$，并发安全模板 $\Lambda_{UNSAFE}(S)$ 定义为

$$\Lambda_{UNSAFE}(S) := \{\pi_1 \mid \forall \rho \in \mathit{plays}(\pi_1) \cdot \forall i \geq 0 \cdot \mathit{supp}(\pi_1(\rho_{\leq i})) \cap S(\rho_i) = \emptyset\}. \quad (3)$$

本节余下部分将说明如何计算 S，使得 $\Lambda_{\mathrm{UNSAFE}}$ 在安全博弈 $(G, \Box I)$ 中几乎必然获胜，因此该模板是可靠的。

模板计算。文献 [1]表明，并发安全博弈 $(G, \Box I)$ 中玩家1的（几乎必然）获胜区域可以通过符号化不动点公式计算，该公式与回合制安全博弈的不动点公式相同，即 $\mathcal{W}_1(\Box I) = \nu X.(\mathrm{pre}_1(X) \cap I)$。我们使用 $\mathrm{AlmostSafe}(G, I)$ 表示实现该公式的算法。

现在回顾方程 (1) 中的动作集函数 $A_Y^v: 2^{\Gamma_2(v)} \to 2^{\Gamma_1(v)}$，该函数刻画了状态 v 下玩家1的动作集合：在玩家2避开γ_2中的动作这一假设下，这些动作保证能够到达 $Y \subseteq V$中的某个状态。由于 γ_2 是玩家2为避免继续留在安全集合 Y 中唯一可以执行的动作，因此我们可以在 (1) 中设定 $\gamma_2 = \emptyset$ 和 $Y = \mathcal{W}_1(\Box I)$，从而得到玩家1的动作集合；这些动作确保无论玩家2采取何种动作，博弈过程都保持在 $\mathcal{W}_1(\Box I)$ 内。由此，我们得到如下构造。

定理 1。设 $(G, \Box I)$ 是一个并发安全博弈，其 $\mathcal{W}_1 = \mathcal{W}_1(\Box I)$ 且

$$S(v) := \begin{cases} \Gamma_1(v) \setminus A_{\mathcal{W}_1}^v(\emptyset) & \text{if } v \in \mathcal{W}_1, \\ \emptyset & \text{if } v \notin \mathcal{W}_1 \end{cases}. \quad (4)$$

那么 $\Lambda_{UNSAFE}(S)$ 在$(G, \Box I)$ 中几乎必然获胜，且是最大宽松的。

我们将 S 称为安全函数，并将其计算过程称为 $\mathrm{SafetyTemp}(G, I)$。

[1] 并发安全博弈允许采用确定性策略。为便于将安全模板与其他模板组合，我们仍基于随机化策略定义安全模板。

4.3 并发活跃组模板

本节以与轮流制博弈中的活性模板直接类比的方式，介绍并发博弈的活性模板。在轮流制博弈中，Anand 等人 [7]通过活组形式化了活性的模板，每个活组都包含一组 H 玩家1的边。只要玩家1的策略能够确保：若在一次博弈过程中某个源状态 src(H) 无限次出现，则该组 H 中至少有一条边会被无限次走过，就能保证始终再次朝着目标状态取得进展。轮流制博弈中活组的计算遵循布奇博弈经典不动点算法的步骤。直观而言，在不动点算法的每次迭代 i 中，玩家1为向前几次迭代中加入的状态取得进展而必须“主动”选择的所有边，构成一个活组 H。

虽然概念上类似，但在并发博弈中实现同一思路所需的数据结构更为复杂，而且从 [1] 中相应的并发 Büchi 博弈不动点算法中提取该结构也更为繁琐。

模板形式化。形式化并发生存模板需要采用不同数据结构，主要有两个原因。首先，我们注意到，由于并发博弈采用基于动作的语义，我们无法通过边集 H 间接定义源状态集合 src(H)。因此，我们分别维护源状态列表 P 和活跃动作 H。第二个更为复杂的区别在于 H 的定义。由于并发博弈采用同步语义，只有随机化策略为玩家1所有可能在玩家2选择“恰当”动作时促成进展的动作都赋予正概率，才能确保进展。因此，在每个状态 $v \in V$ 中，我们需要按玩家2的每个“促成进展”动作，分别记录玩家1的动作集合，由此得到函数 H：$V \to 2^{2^{\Gamma_1(V)}}$。

直观地说，并发活性模板必须确保：对于每个状态 v 以及每个可能取得进展的对手动作 $b \in \Gamma_2(v)$，都为所有能够促成该进展的动作 $\gamma^b \subseteq \Gamma_1(v)$赋予正概率。然而，仅仅让玩家1为这些子集赋予正概率还不够。还必须确保采取该动作的概率不会随时间趋近于零，从而能够无限多次访问目标集。下面对此进行形式化。

定义 2。设 G 为一个并发博弈。给定一个函数 H： $V \to 2^{2^{\Gamma_1(V)}}$和一个划分 $P \subseteq 2^V$，并发活组模板 Λ_{LIVE}（H, P） 定义为： $\pi_1 \in \Lambda_{LIVE}$（H,P） 当且仅当对于所有 $\rho \in$ 博弈过程（π_1） 以及所有 $U \in P$，都有

$$(U \cap \mathit{inf}(\rho) \neq \emptyset) \Rightarrow \left(\sum_{\{i \mid \rho_i \in U\}} \mathit{minProb}_{\pi_1(\rho_{\leq i})}(H(\rho_i)) = \infty\right), \tag{5}$$

$$where\ \mathit{minProb}_p(X) := \min_{\gamma \in X} p(\gamma). \tag{6}$$

直观地说，H(ρ_i) 是玩家1行动集合的集合，而 (5) 确保策略只有在其为 H(ρ_i) 中所有活跃行动集合所分配的最小概率之和趋于无穷时，才能遵循 Λ_{LIVE}(H，P)。这确保相应的动作确实会在期望意义下被无限次执行（如果相应的活跃

状态被无限次观测到）。本节剩余部分将展示如何计算 H 和 P，使得 Λ_{LIVE}(H，P) 从 Büchi 博弈 (G，$\Box I$) 的获胜区域中的所有状态出发都是（几乎必然）获胜的，因此是可靠的。

模板计算。 接下来，可以通过符号不动点公式 $\mathcal{W}_1(\Box\Diamond I) = \nu Y.\mu X.((\neg I \cap \text{Apre}_1(Y, X))) \cup (I \cap \text{pre}_1(Y))))$ 计算并发 Büchi 博弈 (G，$\Box\Diamond I$) 中玩家1的（几乎必然）获胜区域，而 AlmostBüchi (G，I) 表示实现该公式的算法。

与策略提取中的常见做法一样，我们考虑 AlmostBüchi (G，I) 的最后一次迭代（它计算 $Y = \mathcal{W}_1$），并定义满足 $X_1 = I \cap \text{pre}_1(\mathcal{W}_1)$ 且 $X_{i+1} = (\neg I \cap \text{Apre}_1(\mathcal{W}_1, X_i)) \cup X_1$ 的状态集合$X_0 \subseteq X_1 \subseteq \ldots X_k = X_{k+1} = \mathcal{W}_1$ s.t. $i \geq 1$。我们进一步定义 $I_i := X_i \setminus X_{i-1}$，其中 $i \geq 1$。现在，我们直观地定义 H 和 P，使得每当某个 $v \in I_i$ 在博弈过程 $play_\rho$中被无限多次看到时，玩家1就应以一种能够无限多次到达 I_{i-1} 的方式分配概率。形式上，根据上述定义，我们定义 $\mathsf{P} := \{I_1, I_2, \ldots I_k\}$，且对于 $v \in I_i$，有

$$\mathsf{H}(v) := \{\mathsf{H}^b(v)\}_{b \in \Gamma_2(v)} \quad \text{s.t. } \mathsf{H}^b(v) := \{a \in \Gamma_1(v) \setminus \mathsf{S}(v) \mid \delta(v, a, b) \in X_{i-1}\},$$

其中，S(v) 的定义同(3)，但相对于 $\mathcal{W}_1 = \mathcal{W}_1(\Box\Diamond I)$。我们将(H，P)称为活跃组函数，并通过 LivenTemp(G，I)来表示其概要计算过程。

可靠性。 继 [7], 之后，我们证明，为并发 Büchi 博弈计算的并发生存模板确实是可靠的。直观地说，此类博弈中的每个获胜策略都必须：(i) 确保博弈过程始终停留在获胜区域内；(ii) 始终朝着 Büchi 状态推进。其中，(i) 可由并发安全模板确保，而(ii) 则由并发生存模板确保。下面的定理对此进行了形式化（完整算法见附录 A）。

定理 2。 设$(G, \Box\Diamond I)$ 是一个并发 Büchi 博弈，其获胜区域为$\mathcal{W}_1$，安全函数为 $S = \mathit{SafetyTemp}(G, \mathcal{W}_1)$，活跃组函数为$(H, P) = \mathit{LivenTemp}(G, I)$，则 $\Lambda = \Lambda_{\mathit{UNSAFE}}(S) \cap \Lambda_{\mathit{LIVE}}(H, P)$ 是获胜策略模板，适用于$(G, \Box\Diamond I)$。

我们将在推论 1 中证明，模板 $\Lambda = \Lambda_{\text{UNSAFE}}$(S) $\cap\ \Lambda_{\text{LIVE}}$(H，P) 是完备的；也就是说，如果玩家1在博弈$\mathcal{G} = (G, \Box\Diamond I)$中存在获胜策略，那么就存在一种遵循模板 Λ 的策略。然而，该模板并非最大宽松的，也就是说，玩家1可能存在不遵循该模板的获胜策略（示例见 D 节）。

4.4 并发共活模板

本节将并发博弈中的共活模板引入，与轮流制博弈中的共活模板直接类比。虽然其定义背后的直觉也如预期般可以直接沿用，但遗憾的是，从并发余 Büchi 博弈获胜区域的不动点计算中提取可靠的 共活模板，并非那么直接。在更详细地讨论这一挑战之前，我们先形式化定义共活模板。

算法 1 coBüchiTemp$(G,\ I)$

输入： 一个博弈图 G，以及一个状态子集 I
输出： 一个安全函数 S、一个活性元组 (H，P) 和一个共活元组 (C)

1: $Z \leftarrow$ AlmostCo-Büchi$(G,\ I)$；S $\leftarrow$ SafetyTemp$(G,\ Z)$；
2: $X \leftarrow$ AlmostSafe$(G,\ I)$; C $\leftarrow$ SafetyTemp$(G,\ X)$;
3: **对于** $v \in X \cup (V \setminus Z)$ **执行** H$(v) \leftarrow \{\Gamma_1(v) \setminus$ S$(v)\}$;
4: **当** $True$ **执行**
5: $\quad Y' \leftarrow V$;
6: $\quad$ **重复**
7: $\quad\quad Y \leftarrow Y'$; $Y' \leftarrow (I \cap$ AF 前驱$_1(Z,\ Y,\ X)) \cup (\neg I \cap$ Apre$_1(Z,\ X))$;
8: $\quad$ **直到** $Y' == Y$
9: $\quad$ **如果** $X == Y$ **则中断**
10: $\quad$ **对于** $v \in Y \setminus X$ **执行**
11: $\quad\quad$ H$(v) \leftarrow \emptyset$;
12: $\quad\quad$ **对于** $b \in \Gamma_2(v)$ **执行** H$(v) \leftarrow$ H$(v) \cup \{\{a \in \Gamma_1(v) \setminus$ S$(v) \mid \delta(v,a,b) \in X\}\}$;
13: $\quad$ **如果** $Y \setminus X \neq \emptyset$ **则** P $\leftarrow$ P $\cup \{Y \setminus X\}$;
14: $\quad X \leftarrow Y$;
15: **返回** (S，(H，P)，C)

模板形式化。 如第 3 节所示，与轮流制博弈中的对应概念类似，并发共活模板通过共活函数 C：$V \to 2^{\Gamma_1(V)}$ 定义，该函数指定玩家1在 v 中只允许有限次采取的动作。为了确保所有遵循共活模板的策略 π_1 只采取有限次 此类动作，通过 π_1 与这些动作相关联的概率之和必须随时间保持有界。

定义3。 设 G 是一个并发博弈图。给定一个共活函数 C:$V \to 2^{\Gamma_1(V)}$，并发共生模板 $\Lambda_{\mathrm{COLIVE}}(C)$ 定义为： $\pi_1 \in \Lambda_{\mathrm{COLIVE}}(C)$ 当且仅当对所有 $\rho \in$博弈过程(π_1)，以下条件成立：

$$v \in \mathit{inf}(\rho) \Rightarrow \textstyle\sum_{\{i \mid \rho_i = v\}} \pi_1(\rho_{\le i})(C(\rho_i)) \neq \infty. \tag{7}$$

模板计算。 根据 [1]，，并发余 Büchi 博弈（$G, \Diamond\Box I$）中玩家1的（几乎必然）获胜区域可以通过以下符号化不动点公式计算得出

$$\mathcal{W}_1(\Diamond\Box I) = \nu Z.\mu X.\nu Y.\left((I \cap \mathsf{AFpre}_1(Z,Y,X)) \cup (\neg I \cap \mathsf{Apre}_1(Z,X))\right), \tag{8}$$

其中，AlmostCo-Büchi（G, I）表示实现该函数的算法。有趣的是，(8) 是一个三层嵌套的不动点公式，而在轮流制博弈中，计算 $\mathcal{W}_1(\Diamond\Box I)$只需要两层嵌套。

与 Büchi 情形一样，我们可以从 AlmostCo-Büchi 的最后一次迭代（$G,\ I$）（计算 $Z^* = \mathcal{W}_1$）中提取获胜策略模板，具体如算法 1 所述。和之前一样，我们首先计算获胜区域 $Z^* = \mathcal{W}_1$，并通过安全函数 S 确保博弈过程始终保持在该集合内（第 1 行）。

为了理解算法 1 的第 2 行，回想一下， Z^* 可以写成一列递增的集合 $X_1 \subset X_2 \subset X_3 \subset ... \subset X_k = X_{k+1} = Z^* = \mathcal{W}_1$。现在可以证明，根据第 4.1 节中 AF 前驱的定义，AFpre$_1(Z^*, Y, \emptyset)$可化简为 pre$_1(Y)$；而对于所有 $Y \subseteq V$，Apre$_1(Z^*, \emptyset) = \emptyset$则可直接得出。因此，我们可以得出结论：$X_1 = Y^* = \nu Y.[I \cap \text{pre}_1(Y)] = \mathcal{W}_1(\Box I) = \text{AlmostSafe}(G, I)$。也就是说，$X_1$ 是玩家1能够通过策略使博弈无限期保持在 I 中的状态。然而，由于 Z^*的定义，玩家1可以有限次离开 X_1 进入 Z^*，因为 AF 前驱确保博弈过程几乎必然能够再次被迫朝着 X_1 取得进展。因此，我们将所有离开 X_1 进入 Z^* 的动作收集到共活函数 C 中（算法 1 第 2 行）。

最后，为确保所讨论的进展朝向 X_1，我们还需要一个活组模板 (H，P)。该构造与 Büchi 情形非常相似，但使用了公式 (8) 中的特定不动点（算法 1 的第 4–19 行）。

总之，并发 co-Büchi 博弈的获胜策略模板由一个安全模板（以避免离开获胜区域）、一个 co-live 模板（以避免无限多次离开 X_1 ）以及一个活组模板（以确保最终到达 X_1 ）组成。以下定理对此进行了形式化。

定理 3。 设 $(G, \Diamond\Box I)$ 是一个并发 co-Büchi 博弈，其获胜区域为$\mathcal{W}_1$；如果 $(S, (H, P), C) = coB\ddot{u}chiTemp(G, I)$，那么 $\Lambda = \Lambda_{UNSAFE}(S) \cap \Lambda_{LIVE}(H, P) \cap \Lambda_{COLIVE}(C)$ 是 $(G, \Diamond\Box I)$的获胜策略模板。

虽然我们将在推论 1 中证明，模板 $\Lambda = \Lambda_{\text{UNSAFE}}$(S) $\cap \Lambda_{\text{LIVE}}$(H，P) $\cap \Lambda_{\text{COLIVE}}$ (C) 对博弈 $\mathcal{G} = (G, \Diamond\Box I)$ 是完备的，但该模板并非最大宽松（示例见 D 节）。

注 *1*。 我们注意到，轮流制 co-Büchi 博弈中的获胜策略模板可以仅通过安全与共活模板来表示。这是因为，为确保朝向安全不变集取得进展所需的活组模板，对于轮流制 co-Büchi 博弈可以转换为共活模板，因为只需在有限 时间范围内取得进展。遗憾的是，对于并发博弈而言，这一点并不成立，因为在这种情况下，活组模板与共活模板的并发版本并非对偶。

5 无冲突性与策略提取

本节讨论如何高效地提取一个遵循给定策略模板的策略。正如第 3 节所述，尽管在许多情况下，可以轻松构造出一个遵循给定策略模板的策略，但并非总是存在遵循给定策略模板的策略。例如，考虑策略模板 Λ_{UNSAFE}(S)，该模板禁止从状态 v 执行所有可用动作，即 S(v) $= \Gamma_1(v)$。在这种情况下，不存在遵循该模板的策略。我们将策略模板中的此类情形称为<风格 id='6'>冲突</风格>。为形式化这一概念，我们将 [7] 中的<风格 id='8'>无冲突性</风格>概念推广到并发博弈。

定义 4。给定博弈图 $G = (V, \Gamma_1, \Gamma_2, \delta)$，若策略模板 $\Lambda = \Lambda_{UNSAFE}(S) \cap \Lambda_{LIVE}(H, P) \cap \Lambda_{COLIVE}(C)$ 对每个状态 $v \in V$ 都满足以下条件，则称其为无冲突的：*(i)* $\Gamma_1(v) \not\subseteq S(v) \cup C(v)$，以及 *(ii)* 对于每个 $\gamma \in H(v)$，$\gamma \not\subseteq S(v) \cup C(v)$。

直观地说，条件 (i) 确保在每个状态 v 至少有一个可用动作不被安全性或共活性模板禁止。条件 (ii) 确保即使不考虑不安全动作和共活跃动作，活性模板仍然可以得到满足，即对于每个对手动作，在 v 至少有一个可用的活跃动作不被安全性或共活性模板禁止。

基于上述直觉，从给定的无冲突策略模板中提取无记忆策略的一种简单方法如下：对于每个状态，移除所有不安全动作和共活跃动作，然后为剩余动作赋予正概率。根据定义 4 的条件 (i)，移除不安全动作和共活跃动作后，每个状态仍至少有一个可用动作，因此该策略定义良好。显然，该策略遵循安全性和共活性模板。此外，根据定义 4 的条件 (ii)，对于每个对手动作，至少有一个活跃动作被赋予正概率，因此该策略也遵循活性模板。因此，该策略确实遵循该策略模板，从而得到以下结果。

定理 4. 给定一个博弈图 $G = (V, \Gamma_1, \Gamma_2, \delta)$ 和一个无冲突策略模板 Λ，可以在时间 $\mathcal{O}(|V| \cdot |\Gamma_1| \cdot |\Gamma_2|)$内提取一个遵循该模板的无记忆策略。

与 [7]，类似，我们也可以证明，我们在第 4 节中提出的所有算法都会生成无冲突策略模板。

定理 5. *SafetyTemp*、*BüchiTemp* 和 *coBüchiTemp* 过程总是返回无冲突策略模板。

作为定理 5 的推论，我们还可以得到第 4 节中算法的完备性，即对于获胜区域中的每个状态，我们都可以提取一个遵循所计算策略模板的获胜策略。

推论 1. 给定定理 *2*（分别为定理 *3*）的前提，对于每个状态 $v \in \mathcal{W}_1$，都存在一个获胜策略 π_1，使玩家 1 从 v 出发并遵循由算法 *2*（分别为算法 *1*）计算出的策略模板 Λ。

6 策略模板的组合

借助无冲突性这一性质，我们现在还可以通过求交来组合不同的策略模板，并且仍能确保存在一种遵循组合后模板的策略。当我们希望组合多个目标时，这尤其有用，因为我们可以分别计算每个目标对应的策略模板，再将它们求交，从而得到一个组合后的策略模板。首先，我们来定义两个策略模板的组合。

定义 *5.* 对于博弈图 $G = (V, \Gamma_1, \Gamma_2, \delta)$，策略模板 $\Lambda_i = \Lambda_{UNSAFE}(S_i) \cap \Lambda_{LIVE}(H_i, P_i) \cap \Lambda_{COLIVE}(C_i)$ 的组合（其中 $i \in \{1, 2\}$）定义为策略模板 $\Lambda_1 \cap \Lambda_2 = \Lambda_{UNSAFE}(S) \cap \Lambda_{LIVE}(H, P_1 \cup P_2) \cap \Lambda_{COLIVE}(C)$，其中 $S(v) = S_1(v) \cup S_2(v)$，$H(v) = H_1(v) \cup H_2(v)$，且 $C(v) = C_1(v) \cup C_2(v)$。

直观地说，如果某个动作在任一单独模板中是不安全的，组合策略模板就会将其标记为不安全；共活跃动作也同样如此。对于活跃动作，组合模板会纳入两个模板中的所有活跃动作组，并同时考虑两个单独模板的划分。这确保一个策略遵循组合模板，当且仅当它同时遵循两个单独模板。根据此定义，只要组合策略模板无冲突，我们就可以确信存在一个同时遵循两个单独模板的策略，从而得到以下结果。

定理 6. 设 (G, Φ_1) 和 (G, Φ_2) 为并发博弈，其对应的获胜区域分别为 $\mathcal{W}_1$ 和 $\mathcal{W}_2$，获胜策略模板分别为 Λ_1 和 Λ_2。若组合策略模板 $\Lambda_1 \cap \Lambda_2$ 无冲突，则它是博弈 $(G, \Phi_1 \wedge \Phi_2)$ 的获胜策略模板，其获胜区域为 $\mathcal{W}_1 \cap \mathcal{W}_2$。

请注意，一般而言，博弈（G，$\Phi_1 \wedge \Phi_2$）的获胜区域不一定等于 $\mathcal{W}_1 \cap \mathcal{W}_2$。然而，如果合并后的策略模板无冲突，那么合取博弈的获胜区域确实是 $\mathcal{W}_1 \cap \mathcal{W}_2$。

7 实验

实现与设置。 我们的算法以 Python 工具 ConSTel 实现，可处理 Büchi 和 co-Büchi 博弈，因此也能处理具有两种颜色的奇偶博弈。目前我们尚未发现针对并发博弈的大规模基准测试。因此，为了进行评估，我们将 SYNTCOMP 基准测试 [21] 中的轮流行动奇偶博弈转换为并发博弈。具体而言，我们选取了规模较小的模型，这些模型具有交替结构（即玩家1状态之后总是玩家2状态，反之亦然），且可转换为 Büchi（或 co-Büchi）博弈。转换时，我们首先将每个玩家1转移与其对应后继玩家2状态的转移合并。随后移除所有玩家2状态，并将基于转移的获胜条件转换为基于状态的获胜条件[2]。我们总共转换了 171 个博弈。所有实验均在配备 Ubuntu 22.04、Intel i7-1165G7 CPU 和 32GB 内存的机器上运行。我们的实现和实验数据见 [10]。

我们将原型实现与 PeSTel [7], 进行了比较。PeSTel 是一种用于计算回合制博弈策略模板的工具。结果见附录第 E 节，表明 ConSTel 明显较慢。我们认为，性能较慢是由编程语言差异以及解决并发博弈本身的难度所致，并将工程改进留待

[2] 我们注意到，所得的并发博弈并未捕捉到与原轮流制博弈相同的交互动态。转换过程及所考虑的模型详见附录E节

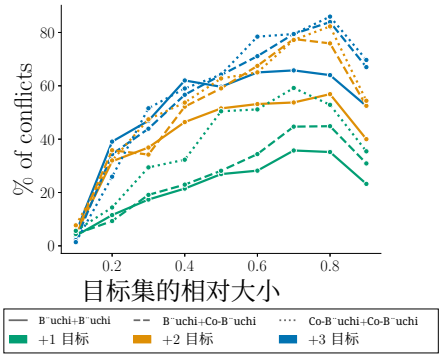


图2：冲突分析。

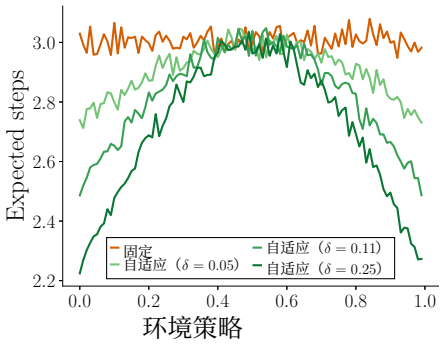


图3：达到 S_e 所需的预期步数。

未来工作。接下来，我们将探究这些模板在增量综合和运行时策略优化中的效用。

增量综合。 我们研究了该方法处理新到达目标的能力。以初始目标的模板为起点，我们关注将现有模板与新目标的模板结合时产生的冲突数量。我们选择了四个转换后的博弈，这些博弈至少包含20个状态，且初始目标的获胜区域非空。对于每个博弈，我们随后引入了 $k \in \{1, 2, 3\}$ 个不同的附加目标，其类型相同（Büchi 或余 Büchi），每个目标的大小为$s \in \{10\%,\ 20\% \ldots,\ 90\%\}$（相对于状态数量）。对于每种配置（$s$, k），我们考虑了100个随机样本，并报告了将现有模板与新目标的模板结合时的相对冲突数量。结果如图2所示，其中附加目标的数量 k 用颜色表示，目标类型则通过线型表示。正如预期，随着附加目标数量的增加，冲突也会增多。大多数冲突发生在目标集合较大时（60%至80%）。直观而言，较小的集合会导致较小的获胜区域，因此可能发生冲突的状态也更少。对于最大的集合，冲突反而较少，因为目标变得更容易满足。值得注意的是，无论目标大小如何，我们通常都可以向现有的 *Büchi* 模板中添加附加的 *Büchi* 模板 而不会产生冲突（对于 $+1$ 目标，冲突为 $\leq 36\%$）。当涉及余 Büchi 目标时，冲突更为普遍，但对于较小的目标而言仍处于合理范围内。更多细节和图表见附录E节。

运行时优化。 最后，我们展示了如何在机器人示例（图 1）中应用我们的模板进行运行时策略优化。按照我们的策略模板，为通往绿色单元格的动作分配正概率后，机器人 R_C 几乎必然会独自到达一个绿色单元格。请注意，不同策略下机器人到达 S_e 所需的预期时间 可能不同。从概念上讲，该模板使机器人能够在运行时改变策略，同时确保它几乎必然到达 S_e 。利用这种灵活性，机器人可以在模板规定的策略中进行调整，以最小化

预期时间。假设环境策略固定，我们考虑一种简单的自适应方法，根据观察到的环境动作来更新机器人的策略。起初，机器人为通往绿色单元格的两个动作分配等概率。随着博弈的进行，机器人观察环境的动作，并通过 δ 调整概率，偏向更有可能成功的动作，同时确保仍遵循该模板。

我们针对不同的环境策略和 δ 的取值，对该方法进行了 10000 次模拟。结果如图 3 所示。x 轴对应环境策略，并表示环境选择顺时针 移动的概率。y 轴表示机器人到达S_e所需的预期时间。固定策略为两个动作分配等概率，通常需要三步才能到达 S_e。根据 δ 和环境策略，机器人遵循自适应方法时能够显著减少步数。这表明，在动态适应具有优势的场景中，我们的基于模板的方法具有应用潜力。

数据可用性声明。 包含模型、实验数据、实现以及用于复现实验的脚本的工件可在 https://doi.org/10.5281/zenodo.18187322 找到。工件的最新版本可在 https://doi.org/10.5281/zenodo.17357028 [10] 找到。

参考文献

1. de Alfaro, L., Henzinger, T.：并发 ω-正则博弈。载于：第十五届 IEEE 计算机科学逻辑研讨会论文集（目录号：99CB36332）。第141–154页（2000）。https://doi.org/10.1109/LICS.2000. 855763 2. de Alfaro, L., Henzinger, T.A., Kupferman, O.：并发可达性博弈。载于：第39届计算机科学基础年会，FOCS 1998，美国加利福尼亚州帕洛阿尔托，1998年11月8日至11日。第564–575页。IEEE计算机学会（1998）。https://doi.org/10.1109/SFCS.1998.743507, https://doi.org/10.1109/SFCS.1998.743507 3. Alur, R., Henzinger, T.A.：反应式模块。载于：论文集，第11届 IEEE 计算机科学逻辑年会，美国新泽西州新不伦瑞克，1996年7月27日至30日。第207–218页。IEEE计算机学会（1996）。https://doi.org/10. 1109 561320 4 1996 561320, 10 1109 1996 /LICS. . https://doi.org/ . /LICS. . . Alur, R., Henzinger, T.A., Kupferman, O.：交替时间时序逻辑。《ACM期刊》**49**(5)，672–713（2002）。https://doi.org/10.1145/585265.585270, https: //doi.org/10 585270 5 1145 585265. / . . Alur, R., Henzinger, T.A., Mang, F.Y.C., Qadeer, S., Rajamani, S.K., Tasiran, S.：MOCHA：模型检测中的模块化。载于：Hu, A.J., Vardi, M.Y.（编）：计算机辅助验证，第十届国际会议，CAV ’98，加拿大不列颠哥伦比亚省温哥华，1998年6月28日至7月2日，论文集。计算机科学讲义，第1427卷，第521–525页。施普林格（1998）。https://doi.org/10.1007/BFB0028774, https://doi.org/10 0028774 6 1007. /BFb . Anand, A., Mallik, K., Nayak, S.P., Schmuck, A.：为综合计算充分宽松的假设。载于：Sankaranarayanan, S., Sharygina, N.（编）：系统构造与分析工具和算法——第二十九届国际

会议，TACAS 2023，作为欧洲软件理论与实践联合会议ETAPS 2022的一部分举办，法国巴黎，2023年4月22日至27日，论文集，第二部分。计算机科学讲义，第13994卷，第211–228页。施普林格（2023）。https://doi.org/10.1007/978-3-031-30820-8_15, https: //doi.org/10.1007/978-3-031-30820-8_157. Anand, A., Nayak, S.P., Schmuck, A.K.：综合奇偶博弈的宽容型获胜策略模板。载于：Enea, C., Lal, A.（编）：计算机辅助验证。第436–458页。Springer Nature Switzerland，查姆（2023） 8. Anand, A., Nayak, S.P., Schmuck, A.：策略模板——交互系统的鲁棒认证接口。载于：Akshay, S., Niemetz, A., Sankaranarayanan, S.（编）：自动化验证与分析技术——第二十二届国际研讨会，ATVA 2024，日本京都，2024年10月21日至25日，论文集，第一部分。计算机科学讲义，第15054卷，第22–41页。施普林格（2024）。https://doi.org/10.1007/978-3-031-78709-6_2, https://doi.org/10.1007/978-3-031-78709-6_29. Anand, A., Schmuck, A.K., Prakash Nayak, S.：基于契约的分布式逻辑控制器综合。载于：第27届 ACM混合系统：计算与控制国际会议论文集。HSCC ’24，计算机协会，纽约州纽约市，美国（2024）。https://doi.org/10.1145/3641513. 3650123, https://doi.org/10.1145/3641513.3650123 10. 匿名：TACAS 2026工件（2025年10月）。https://doi.org/10.5281/ zenodo.17357028, https://doi.org/10.5281/zenodo.17357028 11. Bowling, M.H., Veloso, M.M.：同步对抗式多机器人学习。载于：Gottlob, G., Walsh, T.（编）：IJCAI-03，第十八届人工智能国际联合会议论文集，墨西哥阿卡普尔科，2003年8月9日至15日 699 704 2003 - 。第–页。摩根·考夫曼（），http://ijcai.org/Proceedings/ 03/Papers/102.pdf 12. Brenguier, R.：PRALINE：用于计算并发博弈中纳什均衡的工具。载于：Sharygina, N., Veith, H.（编）：计算机辅助验证——第二十五届国际会议，CAV 2013，俄罗斯圣彼得堡，2013年7月13日至19日。论文集。计算机科学讲义，第8044卷，第890–895页。施普林格（2013）。https://doi.org/10.1007/978-3-642-39799-8_63, https: //doi.org/10.1007/978-3-642-39799-8_6313. Chatterjee, K.：具有尾部目标的并发博弈。《理论计算机科学》**388**(1- 3)，181–198（2007）。https://doi.org/10.1016/J.TCS.2007.07.047, https:// doi.org/10.1016/j.tcs.2007.07.047 14. Cheng, C., Knoll, A.C., Luttenberger, M., Buckl, C.：GAVS+：用于算法博弈求解研究的开放平台。载于：Abdulla, P.A., Leino, K.R.M.（编）：系统构造与分析工具和算法——第十七届国际会议，TACAS 2011，作为欧洲软件理论与实践联合会议ETAPS 2011的一部分举办，德国萨尔布吕肯，2011年3月26日至4月3日。论文集。计算机科学讲义，第6605卷，第258–261页。施普林格（2011）。https://doi.org/10.1007/978-3-642-19835-9_22, https: //doi.org/10.1007/978-3-642-19835-9_2215. Emerson, E.A., Jutla, C.S.：树自动机、μ演算与确定性。载于：FOCS’91。第368–377页（1991） 16. Etessami, K., Yannakakis, M.：论纳什均衡及其他不动点的复杂性。《SIAM计算杂志》**39**(6)，2531–2597（2010）。https://doi.org/10. 1137/080720826, https://doi.org/10.1137/080720826

17. Fijalkow, N., Aiswarya, C., Avni, G., Bertrand, N., Bouyer, P., Brenguier, R., Carayol, A., Casares, A., Fearnley, J., Gastin, P., Gimbert, H., Henzinger, T.A., Horn, F., Ibsen-Jensen, R., Markey, N., Monmege, B., Novotnÿ, P., Ohlmann, P., Randour, M., Sankur, O., Schmitz, S., Serre, O., Skomra, M., Sznajder, N., Vandenhove, P.:《图上的博弈：从逻辑与自动机到算法》。在线版（2025）18. Finkbeiner, B.: 反应式系统综合。载于：可信软件系统工程，第 72–98 页。IOS Press（2016）19. Gutierrez, J., Najib, M., Perelli, G., Wooldridge, M.J.: EVE：时序均衡分析工具。载于：Lahiri, S.K., Wang, C.（编）《自动化验证与分析技术——第 16 届国际研讨会，ATVA 2018，2018 年 10 月 7 日至 10 日，美国加利福尼亚州洛杉矶，论文集》。计算机科学讲义，卷 11138，第 551–557 页。施普林格（2018）。 https://doi.org/10.1007/ 978 3 030 01090 4_35 10 1007 978 3 030 01090 4_35 ---- 20 , https://doi.org/ . / - - - - . Henzinger, T.A.: 系统设计与验证中的博弈。载于：第 10 届理性与知识理论方面会议论文集，第 1–4 页。TARK '05，新加坡国立大学，新加坡（2005）21. Jacobs, S., Pérez, G.A., Abraham, R., Bruyère, V., Cadilhac, M., Colange, M., Delfosse, C., van Dijk, T., Duret-Lutz, A., Faymonville, P., Finkbeiner, B., Khalimov, A., Klein, F., Luttenberger, M., Meyer, K.J., Michaud, T., Pom- mellet, A., Renkin, F., Schlehuber-Caissier, P., Sakr, M., Sickert, S., Staquet, G., Tamines, C., Tentrup, L., Walker, A.: 反应式综合竞赛（SYNTCOMP）：2018–2021。《国际软件工具技术与转移杂志》**26**(5), 551–567（2024）。 https://doi.org/10.1007/S10009-024-00754-1, https://doi.org/10.1007/s10009-024-00754-1 22. Kozen, D.: 命题 μ-演算的结果。《理论计算机科学》**27**(3), 333–354（1983）23. Kress-Gazit, H., Lahijanian, M., Raman, V.: 机器人综合：机器人行为的保证与反馈。《控制、机器人与自主系统年度评论》**1**(1), 211–236（2018）24. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Prism-games 3.0：包含并发、均衡与时间的随机博弈验证。载于：Lahiri, S.K., Wang, C.（编）《计算机辅助验证》，第 475–487 页。Springer International Publishing，查姆（2020）25. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS：用于验证多智能体系统的开源模型检测器。《国际软件工具技术与转移杂志》**19**(1), 9–30（2017）。 https://doi.org/10.1007/S10009-015-0378-X, https://doi.org/10.1007/s10009-015-0378-x 26. Nayak, S.P., Egidio, L.N., Della Rossa, M., Schmuck, A.K., Jungers, R.M.: 基于上下文触发抽象的控制设计。《IEEE 开放控制系统期刊》**2**, 277–296（2023）27. Nayak, S.P., Schmuck, A.K.: 图博弈中最一般的获胜安全均衡综合。载于：Finkbeiner, B., Kovács, L.（编）《系统构造与分析工具和算法》，第 173–193 页。Springer Nature Switzerland，查姆（2024）28. Nisan, N., Roughgarden, T., Éva Tardos, Vazirani, V.V.（编）：《算法博弈论》。剑桥大学出版社（2007）。 https://doi.org/10.1017/ CBO9780511800481, 9780511800481 29 10 1017 https://doi.org/ . /CBO . Phalakarn, K., Pruekprasert, S., Hasuo, I.: 随机奇偶博弈的获胜策略模板：迈向许可式与弹性控制。载于：Anutariya, C., Bon-

sangue, M.M.（编）《计算理论方面——ICTAC 2024》，第 197–214 页。Springer Nature Switzerland，查姆（2025）30. Pnueli, A., Rosner, R.: 关于反应式模块的综合。载于：第 16 届 ACM SIGPLAN-SIGACT 编程语言原理研讨会论文集，第 179–190 页（1989）31. Shapley, L.S.: 随机博弈。《美国国家科学院院刊》**39**(10), 1095–1100（1953）32. Tabuada, P.:《混合系统的验证与控制：一种符号方法》。施普林格（2009）33. Wang, Y., Dantam, N.T., Chaudhuri, S., Kavraki, L.E.: 将任务与运动策略综合视为活性博弈。载于：Coles, A.J., Coles, A., Edelkamp, S., Magazzeni, D., Sanner, S.（编）《第二十六届国际自动规划与调度会议论文集》，ICAPS 2016，17, 2016 12 536 2016 伦敦，英国，6 月 - 。第 页。AAAI 出版社（），http://www.aaai.org/ocs/index.php/ICAPS/ ICAPS16 13146 34 /paper/view/ . Xiong, L., Liu, Y.: 情境演算中不完全信息并发博弈的策略表示与推理。载于：Kambhampati, S.（编）《第二十五届国际人工智能联合会议论文集》，IJCAI 2016，美国纽约州纽约市，2016 年 7 月 9 日至 15 日。第 1322–1329 页。IJCAI/AAAI 出版社（2016 191 35 16 ），http://www.ijcai.org/Abstract/ / . Yin, X., Gao, B., Yu, X.: 安全关键型自主系统控制器的形式化综合：进展与挑战。《控制年度评论》**57**, 100940 2024 10 1016 2024（）。https://doi.org/https://doi.org/ . /j.arcontrol. . 100940

A 算法

本节给出算法 2 中计算 Büchi 博弈策略模板的完整算法。

若 $(G,\ \Box\Diamond I)$ 是一个并发 Büchi 博弈，则算法 BüchiTemp（见算法 2）会计算出一个安全函数 S 和一个活性元组 (H，P)，使得模板 $\Lambda = \Lambda_{\text{UNSAFE}}$(S) $\cap \Lambda_{\text{LIVE}}$(H，P) 在该博弈中对玩家1获胜。算法首先通过 AlmostBüchi $(G,\ I)$ 计算获胜区域 $\mathcal{W}_1$，然后使用过程 SafetyTemp($G,\ \mathcal{W}_1$) 计算安全函数 S。之后，它将目标集限制为 $I \cap \mathcal{W}_1$，并使用过程 LivenTemp($G,\ I$) 计算活性元组 (H，P)。最后，我们将获胜区域 $\mathcal{W}_1$之外的每个状态的所有动作都指定为活跃动作（包括 I中的某些状态），以确保活跃函数 H 定义良好。因此，对于获胜区域 $\mathcal{W}_1$之外的每个状态以及 $I \cap \mathcal{W}_1$ 中的每个状态，算法都在 H 中将所有动作指定为活跃动作（第 4 行）。

算法 2 $\textsc{BüchiTemp}(G, I)$

输入： 一个博弈图 G，以及一个状态子集 I
输出： 一个安全函数 S 和一个活性元组 (H,P)

1: $\mathcal{W}_1 \leftarrow$ AlmostBüchi（$G,\ I$）；S $\leftarrow$ SafetyTemp（$G,\ \mathcal{W}_1$）；
2: $I \leftarrow I \cap \mathcal{W}_1$;
3: (H，P) $\leftarrow$ LivenTemp($G,\ I$);
4: **对于** $v \in I \cup (V \setminus \mathcal{W}_1)$ **执行** H(v) $\leftarrow \{\Gamma_1(v) \setminus$ S(v)$\}$;
5: **返回** (S, (H,P))
6: **过程** $\textsc{LivenTemp}(G, I)$
7: P $\leftarrow \emptyset$;
8: **当** $I \neq \mathcal{W}_1$ **执行**
9: $I' \leftarrow$ Apre$_1(\mathcal{W}_1,\ I)$; P $\leftarrow$ P $\cup \{(I' \setminus I)\}$;
10: **对** $v \in I' \setminus I$ **执行**
11: H(v) $\leftarrow \emptyset$;
12: **对** $b \in \Gamma_2(v)$ **执行**
13: H(v) $\leftarrow$ H(v) $\cup \{\{a \in \Gamma_1(v) \setminus$ S(v) $\mid \delta(v,\ a,\ b) \in I\}\}$;
14: $I \leftarrow I'$;
15: **返回** (H,P)

B 结果的形式化证明

本节将重述论文正文中的所有结果，并给出其形式化证明。

定理 1. 设 $(G, \Box I)$ 是一个并发安全博弈，其具有 $\mathcal{W}_1 = \mathcal{W}_1(\Box I)$ 且

$$S(v) := \begin{cases} \Gamma_1(v) \setminus A^v_{\mathcal{W}_1}(\emptyset) & \text{if } v \in \mathcal{W}_1, \\ \emptyset & \text{if } v \notin \mathcal{W}_1 \end{cases}. \tag{4}$$

则 $\Lambda_{UNSAFE}(S)$ 在$(G,\Box I)$ 中几乎必然获胜，且是最大宽松的。

证明。设 π_1 为遵循 Λ_{UNSAFE}(S) 的策略。为简便起见，在证明的其余部分中，我们用 $\mathcal{W}_1$ 表示 $\mathcal{W}_1(\Box I)$。我们需要证明，对于每个 $v \in \mathcal{W}_1$， π_1 从 v出发几乎必然获胜。不过，我们将证明一个更强的性质，即对于每个 $v \in \mathcal{W}_1$，从 v 出发的每个 π_1-博弈过程都属于 $\mathcal{L}(\Box I)$。此外，很容易看出，策略无法从安全集 I之外的状态获胜。因此，$\mathcal{W}_1 \subseteq I$，所以只需证明每个从 $\rho_0 = v$ 出发的 π_1-博弈过程 $\rho \in V^\omega$ 都包含在获胜区域 $\mathcal{W}_1$中，即对于所有 $i \geq 0$， $\rho_i \in \mathcal{W}_1$。我们使用归纳法来证明这一论证。

对于基础情形，根据构造有 $\rho_0 \in \mathcal{W}_1$ 。对于归纳情形，我们证明若 $\rho_i \in \mathcal{W}_1$，则 $\rho_{i+1} \in \mathcal{W}_1$。假设 $\rho_i \in \mathcal{W}_1$，则根据 (4)，有：

$$\mathsf{S}(\rho_i) = \Gamma_1(\rho_i) \setminus \mathsf{A}^{\rho_i}_{\mathcal{W}_1}(\emptyset)$$

并且，由于 π_1 遵循 Λ_{UNSAFE}(S)，根据 (3) 可得：

$$\mathsf{supp}(\pi_1(\rho_{\leq i})) \cap \left(\Gamma_1(\rho_i) \setminus \mathsf{A}^{\rho_i}_{\mathcal{W}_1}(\emptyset)\right) = \emptyset$$

由于 $\mathsf{supp}(\pi_1(\rho_{\leq i})) \subseteq \Gamma_1(\rho_i)$，我们有：

$$\mathsf{supp}(\pi_1(\rho_{\leq i})) \subseteq \mathsf{A}^{\rho_i}_{\mathcal{W}_1}(\emptyset).$$

因此，根据 (1)，对于每个 $a \in \mathsf{supp}(\pi_1(\rho_{\leq i}))$ 和每个 $b \in \Gamma_2(\rho_i)$，都有 $\delta(\rho_i,\ a,\ b) \in \mathcal{W}_1$。由于 ρ 是一个 π_1-博弈过程，按照定义，对于某个$a \in \mathsf{supp}(\pi_1(\rho_{\leq i}))$ 和 $b \in \Gamma_2(\rho_i)$，有 $\rho_{i+1} = \delta(\rho_i,\ a,\ b)$。因此，必有 $\rho_{i+1} \in \mathcal{W}_1$。所以，根据归纳法，我们证明了从 v 开始的每个 π_1-博弈过程都包含于 $\mathcal{L}(\Box I)$ 中。

假设 π 是一个获胜策略，那么根据定义，对于每个 $\rho \in \mathsf{plays}(\pi)$ 和每个 $i \geq 0$，每个动作 $a \in \mathsf{supp}(\pi(\rho_{\leq i}))$ 都应当导向 $\mathcal{W}_1$中的一个状态。这意味着，给定一个 $\rho \in \mathsf{plays}(\pi)$，对于每个 $i \geq 0$，都有如下结论：

$$\forall a \in \mathsf{supp}(\pi(\rho_{\leq i})) \cdot \forall b \in \Gamma_2(\rho_i) \cdot \delta(\rho_i, a, b) \in \mathcal{W}_1.$$

根据 (1)，上式可以改写为：

$$\forall a \in \mathsf{supp}(\pi(\rho_{\leq i})) \cdot a \in A^v_{\mathcal{W}_1}(\emptyset).$$

这意味着：

$$\begin{aligned}
&\mathsf{supp}(\pi(\rho_{\leq i})) \subseteq A^v_{\mathcal{W}_1}(\emptyset) \\
\Rightarrow &\mathsf{supp}(\pi(\rho_{\leq i})) \cap (\Gamma_1(v) \setminus A^v_{\mathcal{W}_1}(\emptyset)) = \emptyset \\
\Rightarrow &\mathsf{supp}(\pi(\rho_{\leq i})) \cap \mathsf{S}(v) = \emptyset.
\end{aligned}$$

然后，根据安全模板 Λ_{UNSAFE}(S) 的定义，可得 $\pi \in \Lambda_{\text{UNSAFE}}$(S)。因此，并发安全博弈 $\mathcal{G}$ =$(G,\Box I)$ 中的每个几乎必然获胜策略 π 都遵循 (4) 中提出的 Λ_{UNSAFE}(S)。

定理 2。 设$(G,\Box\Diamond I)$ 为一个并发 Büchi 博弈，其获胜区域为$\mathcal{W}_1$，安全函数 S =$\mathit{SafetyTemp}(G,\mathcal{W}_1)$，活跃组函数$(H,\ P)$ =$\mathit{LivenTemp}(G, I)$，则 $\Lambda = \Lambda_{UNSAFE}(S) \cap \Lambda_{LIVE}(H,\ P)$ 是针对$(G,\ \Box\Diamond I)$的获胜策略模板。

证明。 设 π_1 为玩家1的一种策略，该策略遵循模板 $\Lambda = \Lambda_{\text{UNSAFE}}(\mathsf{S}) \cap \Lambda_{\text{LIVE}}(\mathsf{H},\mathsf{P})$。我们需要证明， π_1 是在博弈 $\mathcal{G} = (G, \square\Diamond I)$ 中玩家1的几乎必然获胜策略。也就是说，对于每个 $v \in \mathcal{W}_1$，$\Pr_v^{\pi_1}(\mathcal{L}(\square\Diamond I)) = 1$。首先，根据 $\Lambda_{\text{UNSAFE}}(\mathsf{S})$ 的性质，从任意状态 $v \in \mathcal{W}_1$ 开始的每个 π_1-博弈过程 ρ 都将留在获胜区域 $\mathcal{W}_1$ 内，即 $\rho \in \mathcal{W}_1^\omega$。

现在，$\mathsf{AlmostBüchi}\,(G, I)$ 给出状态集合 $X_0 \subseteq X_1 \subseteq \ldots \subseteq X_k = X_{k+1} = \mathcal{W}_1$，其中 $X_1 = I \cap \mathsf{pre}_1(\mathcal{W}_1)$，且 $X_{i+1} = (\neg I \cap \mathsf{Apre}_1(\mathcal{W}_1, X_i)) \cup X_1$ 对于 $i \geq 1$。令 $U_i \in \mathsf{P}$ 为集合 P 的第 i 个元素，即 $U_i = X_i \setminus X_{i-1}$，且$U_1 = I$。令 E_i 为这样一个对局集合：它会无限多次访问 X_i 中的某个状态，即 $E_i = \{\rho \mid X_i \cap \mathsf{inf}(\rho) \neq \emptyset\}$。由于每个 π_1-博弈过程 ρ 都是一个无限状态序列，因此它会无限多次访问 $X_k = \mathcal{W}_1$。因此，对于每个 $v \in \mathcal{W}_1$，$\Pr_v^{\pi_1}(E_k) = 1$。现在，我们将通过归纳证明：如果对于某个 $i \geq 2$，有 $\Pr_v^{\pi_1}(E_i) = 1$ 对于所有 $v \in \mathcal{W}_1$ 均成立，那么对于所有 $v \in \mathcal{W}_1$，$\Pr_v^{\pi_1}(E_{i-1}) = 1$ 也成立。

假设对于某个 $i \geq 2$，对所有 $v \in \mathcal{W}_1$ 都有 $\Pr_v^{\pi_1}(E_i) = 1$，但对于某个 $v \in \mathcal{W}_1$，$\Pr_v^{\pi_1}(E_{i-1}) \neq 1$。考虑属于 E_i 但不属于 E_{i-1} 的 π_1-对局序列 ρ。因此， ρ 无限次访问 X_i，但只有限次访问 X_{i-1}，这意味着 ρ 必须无限次访问 $U_i = X_i \setminus X_{i-1}$。由于 π_1 遵循 $\Lambda_{\text{LIVE}}(\mathsf{H}, \mathsf{P})$，根据 (5)， $U_i \cap \mathsf{inf}(\rho) \neq \emptyset$ 推出：

$$\sum_{\{j \mid \rho_j \in U_i\}} \mathsf{minProb}_{\pi_1(\rho_{\leq j})}(\mathsf{H}(\rho_j)) = \infty.$$

根据最小概率 (6) 的定义，我们可以将上式写为：

$$\sum_{\{j \mid \rho_j \in U_i\}} \left[\min_{\gamma_1 \in \mathsf{H}(\rho_j)} \left\{ \sum_{a \in \gamma_1} \pi_1(\rho_{\leq j})(a) \right\} \right] = \infty$$

当一个集合的最小值之和为无穷大时，这意味着该集合中每个元素的和都为无穷大。此外，如果我们在 $\mathsf{BüchiTemp}$ 中将 $\rho_j = v'$ 命名为，对于每个 $b \in \Gamma_2(v')$，$\{a \in \Gamma_1(v') \setminus \mathsf{S}(v') \mid \delta(v', a, b) \in I\}$ 已添加到 $\mathsf{H}(v')$ 中。那么：

$$\exists v' \in U_i . \forall b \in \Gamma_2(v') . \exists \gamma_1 \subseteq \Gamma_1(v') . \left[\delta(v', \gamma_1, b) \subseteq U_{i-1} \wedge \sum_{\{j \mid \rho_j = v'\}} \left\{ \sum_{a \in \gamma_1} \pi_1(\rho_{<j})(a) \right\} = \infty \right]$$

这意味着，在 v' 的第 U_{i-1} 次访问中采取相应动作的概率之和为无穷大。因此，我们得出结论：在会无限次访问 U_i 的 π_1-对局序列的样本空间中，无限次访问 U_{i-1} 的概率为 1。

如果在第 j 次访问 v' 时从 v' 访问 U_{i-1} 的概率为 p_j，那么在第 j 次访问 v' 时不从 v' 访问 U_{i-1} 的概率为 $(1 - p_j)$。根据无穷乘积收敛判别法，由于 $\forall j \cdot 0 < p_j < 1$ 且 $\sum p_j{}^j$
发散，则 $\prod(1 - p_j)$ 收敛于 0。因此，无限多次访问 $U_{i-1}{}^j$ 的概率为 1。因此，这与我们的假设矛盾：$\Pr_v^{\pi_1}(E_i) = 1$ 且 $\Pr_v^{\pi_1}(E_{i-1}) \neq 1$。

通过上述归纳论证，我们已经证明，对于所有 $i \geq 1 \mathrm{Pr}_v^{\pi_1}(E_i) = 1$，对所有 $v \in \mathcal{W}_1$ 均成立。因此，对所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(E_1) = 1$ 均成立，这意味着对所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(\mathcal{L}(\Box\Diamond I)) = 1$ 均成立。因此，π_1 是博弈 $\mathcal{G} = (G, \Box\Diamond I)$ 中玩家1的几乎必然获胜策略。

定理 3. 设 $(G, \Diamond\Box I)$ 是一个并发余 Büchi 博弈，其获胜区域为$\mathcal{W}_1$，若$(S, (H, P), C) = CoB\ddot{U}CHITEMP(G, I)$，则 $\Lambda = \Lambda_{UNSAFE}(S) \cap \Lambda_{LIVE}(H, P) \cap \Lambda_{COLIVE}(C)$ 是$(G, \Diamond\Box I)$的获胜策略模板 。

证明。 设 π_1 是一个遵循模板 $\Lambda = \Lambda_{\mathrm{UNSAFE}}(\mathrm{S}) \cap \Lambda_{\mathrm{LIVE}}(\mathrm{H}, \mathrm{P}) \cap \Lambda_{\mathrm{COLIVE}}(\mathrm{C})$的玩家1策略。我们需要证明，$\pi_1$ 是玩家1在博弈 $\mathcal{G} = (G, \Diamond\Box I)$中的几乎必然获胜策略。也就是说，对于每个 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(\mathcal{L}(\Diamond\Box I)) = 1$ 。首先，根据 $\Lambda_{\mathrm{UNSAFE}}(\mathrm{S})$的性质，从任意状态 $v \in \mathcal{W}_1$ 出发的每个 π_1-博弈过程 ρ 都会留在获胜区域 $\mathcal{W}_1$中，即$\rho \in \mathcal{W}_1^\omega$。由于每个$\pi_1$-博弈过程 ρ 都是无限状态序列，因此它会无限次访问 $\mathcal{W}_1$ 。因此，$\mathsf{inf}(\rho) \cap \mathcal{W}_1 \neq \emptyset$。

现在，令 AlmostCo-Büchi (G, I) 给出状态集合 $X_0 \subseteq X_1 \subseteq \ldots \subseteq X_k = X_{k+1} = Z^* = \mathcal{W}_1$，其中 $X_1 = \mathcal{W}_1(\Box I)$。令 $U_i \in \mathrm{P}$ 为集合 P 的第 i 个元素，即 $U_i = X_i \setminus X_{i-1}$，且 $U_1 = X_1$。令 E_i 为对局集合，其中的博弈过程会无限多次访问 X_i 中的一个状态，即 $E_i = \{\rho \mid X_i \cap \mathsf{inf}(\rho) \neq \emptyset\}$。现在，我们将证明：若对所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(E_1) = 1$ ，则对所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}({E_1}^{\complement}) = 0$，其中 ${E_1}^{\complement}$ 是 E_1 的补集。直观而言，这意味着若无限次访问 X_1 的概率为 1，则无限次访问 ${X_1}^{\complement}$ 的概率为 0。

由于 π_1 遵循 $\Lambda_{\mathrm{COLIVE}}(\mathrm{C})$，根据 (7)，对于每个 $v \in X_1$ ，我们有：

$$v \in \mathsf{inf}(\rho) \implies \sum_{\{i \mid \rho_i = v\}} \pi_1(\rho_{\leq i})(\mathsf{C}(\rho_i)) \neq \infty$$

在算法 1 中，我们使用 SafetyTemp(G, X_1) 为每个$v \in X_1$定义 $\mathsf{C}(v)$。因此，$\mathsf{C}(v)$ 包含所有可能导致离开 X_1的动作。于是，根据 *Borel-Cantelli* 引理，上述等式意味着，对于每个 $v \in X_1$，可能导致离开 X_1 的动作在 ρ中只会被选择有限次。这意味着，如果对于所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(E_1) = 1$ ，那么 $\mathrm{Pr}_v^{\pi_1}(E_1^{\complement}) = 0$。现在，我们将证明，对于所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(\mathcal{L}(\Box\Diamond\neg I)) = 0$ ，即无限次访问 $\neg I$ 的概率为 0。为此，我们采用反证法。假设存在一个状态 $v' \in \mathcal{W}_1 \cap \neg I$ ，使得 $v' \in \mathsf{inf}(\rho)$ 以正概率发生。首先注意到，$X_1 \cap \neg I = \emptyset$，因为 $X_1 = \mathcal{W}_1(\Box I)$。因此，对于某个 $i \geq 2$，$v' \notin X_1$ 且$v' \in X_i$ 。由于 π_1 遵循 $\Lambda_{\mathrm{LIVE}}(\mathrm{H}, \mathrm{P})$，根据定理 2，我们知道，如果 $\mathrm{Pr}_v^{\pi_1}(E_i) = 1$，那么对于所有 $v \in \mathcal{W}_1$ 和 $i \geq 2$，$\mathrm{Pr}_v^{\pi_1}(E_{i-1}) = 1$ 。运用与定理 2 相同的归纳论证，我们可以证明，对于所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(E_1) = 1$ 。但是，我们已经证明，如果对于所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(E_1) = 1$ ，那么 $\mathrm{Pr}_v^{\pi_1}(E_1^{\complement}) = 0$。因此，我们得到了矛盾，因为我们假设 $v' \in \mathsf{inf}(\rho) \cap {E_1}^{\complement}$ 以正概率发生。

现在，根据逻辑公式的否定，我们有 $\mathrm{Pr}_v^{\pi_1}(\mathcal{L}(\Diamond\Box I)^{\complement}) = 0$，对所有$v \in \mathcal{W}_1$都成立；这等价于对所有 $v \in \mathcal{W}_1$，$\mathrm{Pr}_v^{\pi_1}(\mathcal{L}(\Diamond\Box I)) = 1$ 。因此，π_1 是博弈 $\mathcal{G} = (G, \Diamond\Box I)$ 中玩家1的几乎必然获胜策略。

定理 5. *SafetyTemp*、*BüchiTemp* 和 *coBüchiTemp* 过程总是返回无冲突的策略模板。

证明。 我们将证明由 SafetyTemp、BüchiTemp 和 coBüchiTemp 计算得到的模板的无冲突性。

对于安全博弈： 假设 S $=$ SafetyTemp(G, I)，那么根据构造，模板 Λ_{UNSAFE}(S) 不会限制获胜区域 $\mathcal{W}_1$之外状态的任何动作。此外，根据定理 1 的证明，我们知道，对于每个状态$v \in \mathcal{W}_1$ 以及每个满足 $\rho_i = v$ 的 $\rho \in$ 对局序列(π)，都存在一个从 v 出发、面向玩家1的获胜策略π，使得 supp($\pi(\rho_{\leq i})$) $\cap$ S(v) $= \emptyset$。因此， Λ_{UNSAFE}(S) 是无冲突的。

对于 Büchi 博弈： 假设 (S, (H,P)) $=$ BüchiTemp(G, I)，那么根据构造，模板 $\Lambda = \Lambda_{\text{UNSAFE}}(S)\cap\Lambda_{\text{LIVE}}$(H,P) 不会限制获胜区域 $\mathcal{W}_1$之外状态的任何动作。此外，根据算法 2，对于每个状态 $v \in \mathcal{W}_1$，H(v) 不包含 S(v) 中的任何不安全动作。因此， Λ 是无冲突的。

对于余 Büchi 博弈： 假设 (S,(H, P), C) $=$ coBüchiTemp(G, I)，那么根据构造，模板 $\Lambda = \Lambda_{\text{UNSAFE}}$(S) $\cap\, \Lambda_{\text{LIVE}}$(H,P) $\cap\, \Lambda_{\text{COLIVE}}$(C) 不会限制获胜区域 $\mathcal{W}_1$ 之外任何状态的动作。此外，根据算法 1，对于每个状态 $v \in \mathcal{W}_1$，C(v) 不包含 S(v) 中的任何不安全动作。同样，根据算法 1，colive 动作只包含来自 X 的动作；并且根据算法 1，对于每个状态 $v \in X$，H(v) 包含来自 v 的所有可能动作。由于 SafetyTemp 的无冲突性，H(v) 中始终存在来自 X 、但不在 C(v) 中的可用动作。因此， Λ 是无冲突的。

C 算法完备性的补充证明

定理 7。给定定理 *2* 的前提，模板 $\Lambda = \Lambda_{UNSAFE}(S) \cap \Lambda_{LIVE}(H, P)$ 是完备的，这意味着如果博弈 $\mathcal{G} = (G, \square\lozenge I)$中存在玩家 1 的获胜策略，那么就存在一个遵循模板 Λ 的策略。

证明。 假设 π 是博弈 $\mathcal{G} = (G,\ \square\lozenge I)$中玩家1的获胜策略。首先，由于 π 是玩家1的获胜策略，它不应离开获胜区域 $\mathcal{W}_1$，也就是说， π 不应选择会导致状态离开 $\mathcal{W}_1$的动作。正如我们在定理1中所证明的，模板 Λ_{UNSAFE}(S)非空。

所有能够通过某个策略恰好用一步（以概率 1）到达 I 的状态构成的集合是 Apre$_1(\mathcal{W}_1,\ I)$，我们在过程 LivenTemp($G,\ I$) 中将其定义为 U_1 。如果一个 π-博弈过程 ρ 无限次访问了 U_1，那么必定存在一个状态 $v \in U_1$，使得 π 是在 v 处玩家1的获胜策略；也就是说，π 应当针对玩家2的每一个可能动作，都对某个动作赋予正概率，从而能够无限多次到达 $U_0 = I$ 。如果事实并非如此，那么玩家2就可以选择一个动作，阻止玩家1从 v到达I，这与 π 是玩家1的获胜策略这一事实相矛盾。

我们可以将这一事实写成如下形式：

$$U_1 \cap \mathsf{inf}(\rho) \neq \emptyset \rightarrow \exists v \in U_1 \cdot \forall b \in \Gamma_2(v) \cdot \exists \gamma_1 \subseteq \Gamma_1(v) \cdot \left[\delta(v, \gamma_1, b) \subseteq U_0 \wedge \sum_{\{j \mid \rho_j = v\}} \left\{ \sum_{a \in \gamma_1} \pi(\rho_{<j})(a) \right\} = \infty \right]$$

由于在 BüchiTemp(G, I) 中，对于每个 $b \in \Gamma_2(v)$，我们都将集合$\gamma_1^b = \{a \in \Gamma_1(v) \setminus \mathsf{S}(v) \mid \delta(v, a, b) \in U_0\}$ 添加到 $\mathsf{H}(v)$ 中（如第 4.3 节所定义），因此可以得出结论：γ_1 是 γ_1^b 的子集，因为 γ_1^b 包含了所有能够到达 U_0 的可能 安全动作。因此，对于所有$a \in \gamma_1$ $a \in \gamma_1^b$，$\sum \pi(\rho_{<j})(a)$ 始终小于 $\sum \pi(\rho_{<j})(a)$ j。因此，如果第一个求和为无穷大，那么第二个求和也为无穷大。因此，我们可以将上面的方程改写如下：

$$U_1 \cap \mathsf{inf}(\rho) \neq \emptyset \rightarrow \exists v \in U_1 \cdot \forall b \in \Gamma_2(v) \cdot \exists \gamma_1^b \subseteq \Gamma_1(v) \cdot \left[\delta(v, \gamma_1^b, b) \subseteq U_0 \wedge \sum_{\{j \mid \rho_j = v\}} \left\{ \sum_{a \in \gamma_1^b} \pi(\rho_{<j})(a) \right\} = \infty \right]$$

由于对于每个 $b \in \Gamma_2(v)$，对 γ_1^b 的求和都是无穷大的，因此这些求和的最小值 也是无穷大。所以，如果我们按照第 4.3 节中的定义固定 γ_1^b，则有：

$$U_1 \cap \mathsf{inf}(\rho) \neq \emptyset \rightarrow \exists v \in U_1 \cdot \min_{b \in \Gamma_2(v)} \left[\sum_{\{j \mid \rho_j = v\}} \left\{ \sum_{a \in \gamma_1^b} \pi(\rho_{<j})(a) \right\} \right] = \infty$$

这里，最小值的变量独立于第一个求和，因此二者可以交换：

$$U_1 \cap \mathsf{inf}(\rho) \neq \emptyset \rightarrow \exists v \in U_1 \cdot \sum_{\{j \mid \rho_j = v\}} \left[\min_{b \in \Gamma_2(v)} \left\{ \sum_{a \in \gamma_1^b} \pi(\rho_{<j})(a) \right\} \right] = \infty$$

根据式 (6) 的定义，我们可以将上述等式写为：

$$U_1 \cap \mathsf{inf}(\rho) \neq \emptyset \rightarrow \exists v \in U_1 \cdot \left[\sum_{\{j \mid \rho_j = v\}} \mathsf{minProb}_{\pi(\rho_{<j})}(\{\gamma_1^b \mid \forall b \in \Gamma_2(v)\}) \right] = \infty$$

如前所述，$\{\gamma_1^b \mid \forall b \in \Gamma_2(v)\}$ 就是我们在过程 LivenTemp(G, I) 中定义的 $\mathsf{H}(v)$。因此，我们可以写成：

$$U_1 \cap \mathsf{inf}(\rho) \neq \emptyset \rightarrow \exists v \in U_1 \cdot \left[\sum_{\{j \mid \rho_j = v\}} \mathsf{minProb}_{\pi(\rho_{<j})}(\mathsf{H}(v)) \right] = \infty$$

那么，我们可以得出结论： U_1 中所有状态的求和将是无穷大的，因为其中一个是无穷大的：

$$U_1 \cap \mathsf{inf}(\rho) \neq \emptyset \rightarrow \sum_{v \in U_1} \left[\sum_{\{j \mid \rho_j = v\}} \mathsf{minProb}_{\pi(\rho_{<j})}(\mathsf{H}(v)) \right] = \infty$$

最后，根据 (5)，我们已经证明，对于 $\mathcal{W}_1$ 的这个划分，模板 Λ_{LIVE}(H,P) 非空。然后，我们可以对 U_2 使用相同的论证； U_2 是恰好经过一步即可到达 U_1 的状态集合，依此类推，直到得出结论：对于每个 U_i， $i \geq 1$，模板 Λ_{LIVE}(H,P) 都非空。

我们只需确保 Λ 对于 U_0 中的状态非空，而 U_0 是目标集 I 的子集。这也由 Λ_{UNSAFE}(S) 保证，因为我们唯一使用的模板就是它，用于将博弈图 G 限制到获胜区域 $\mathcal{W}_1(\square\Diamond I)$；并且我们已在定理 1 中证明，对于 $\mathcal{W}_1$ 中的每个状态，模板 Λ_{UNSAFE}(S) 都非空。
因此，我们已经证明，如果在博弈 $\mathcal{G} = (G, \square\Diamond I)$ 中存在玩家1的获胜策略，就可以得出结论：对于 $\mathcal{W}_1$ 中的每个状态（包括 U_0 中的状态），模板 $\Lambda = \Lambda_{\text{UNSAFE}}$(S) $\cap \Lambda_{\text{LIVE}}$(H,P) 都非空；也就是说，算法 BüchiTemp 是完备的。

定理 8. 在定理 *3* 的前提下，模板 $\Lambda = \Lambda_{UNSAFE}(S) \cap \Lambda_{LIVE}(H, P) \cap \Lambda_{COLIVE}(C)$ 是完备的，也就是说，如果博弈 $\mathcal{G} = (G, \Diamond\square I)$ 中存在玩家1 的获胜策略，那么就存在一种遵循模板 Λ 的策略。

证明。 设 π 是玩家 1 在博弈 $\mathcal{G} = (G, \Diamond\square I)$ 中的获胜策略。首先，由于 π 是玩家 1 的获胜策略，因此它绝不会离开获胜区域 $\mathcal{W}_1$，即 π 绝不应选择会导向 $\mathcal{W}_1$ 之外某个状态的动作。正如我们在定理 1 中所证明的，对于 $\mathcal{W}_1$ 中的每个状态，模板 Λ_{UNSAFE}(S) 均非空。
现在，设 AlmostCo-Büchi (G, I) 给出状态集合 $X_0 \subseteq X_1 \subseteq \ldots \subseteq X_k = X_{k+1} = Z^* = \mathcal{W}_1$，其中 $X_1 = \mathcal{W}_1(\square I)$，并按如下方式定义 inf($\pi(v)$)：

$$\mathsf{inf}(\pi(v)) = \underset{\{\gamma \subseteq \Gamma_1(v) \mid \sum_{\{i \mid v_i = v\}} \min_{a \in \gamma} \pi(v_0 v_1 \ldots v_i)(a) = \infty\}}{\operatorname{argmax}} |\gamma| \tag{9}$$

直观地说，inf($\pi(v)$) 是策略 π 在状态 v 下无限多次同时支撑的最大动作集。
现在，我们将证明，对于每个 $v \in X_1$，已知 $X_1 = \mathcal{W}_1(\square I)$ 时， $\pi \in \Lambda_{\text{COLIVE}}$ (C) 均成立。

对于每个 $v \in X_1$，我们通过 SafetyTemp(v, X_1) 在算法 1 中定义了 C(v)。此外，如果每个 π-博弈过程 $\rho = v_0 v_1 v_2 \ldots$ 无限多次访问 X_1，那么从某个时刻起，X_1 应当只离开有限次；也就是说，π 应当对可能导致离开 X_1 的动作赋予零概率，这些动作相对于 X_1 是不安全动作。否则，

ρ 将无限多次离开 X_1，这与 π 是玩家1的获胜策略这一事实相矛盾。我们可以将这一事实写成如下形式：

$$v \in \mathsf{inf}(\rho) \implies$$
$$\forall a \in \Gamma_1(v) \cdot \Big(\exists b \in \Gamma_2(v) \cdot \delta(v,a,b) \notin X_1 \implies \sum_{\{i|\rho_i = v\}} \pi(\rho_{\leq i})(a) \neq \infty\Big)$$

如果对蕴含式的左侧取否定，我们得到：

$$v \in \mathsf{inf}(\rho) \implies$$
$$\forall a \in \Gamma_1(v) \cdot \Big(\neg(\forall b \in \Gamma_2(v) \cdot \delta(v,a,b) \in X_1) \implies \sum_{\{i|\rho_i = v\}} \pi(\rho_{\leq i})(a) \neq \infty\Big)$$

根据方程 (1)，我们可以将上述方程改写为：

$$v \in \mathsf{inf}(\rho) \implies \forall a \in \Gamma_1(v) \cdot \Big(a \notin \mathsf{A}^v_{X_1}(\emptyset) \implies \sum_{\{i|\rho_i = v\}} \pi(\rho_{\leq i})(a) \neq \infty\Big)$$

这等价于：

$$v \in \mathsf{inf}(\rho) \implies \forall a \in \Gamma_1(v) \setminus \mathsf{A}^v_{X_1}(\emptyset) \cdot \Big(\sum_{\{i|\rho_i = v\}} \pi(\rho_{\leq i})(a) \neq \infty\Big)$$

根据 SafetyTemp (4) 的定义，并且由于我们在算法 1 中通过使用 SafetyTemp(G, X_1) 为每个 $v \in X_1$ 定义了 C(v)，我们知道 C(v) $= \Gamma_1(v) \setminus \mathsf{A}^v_{X_1}(\emptyset)$。因此，我们可以将上述方程改写为：

$$v \in \mathsf{inf}(\rho) \implies \forall a \in \mathsf{C}(v) \cdot \Big(\sum_{\{i|\rho_i = v\}} \pi(\rho_{\leq i})(a) \neq \infty\Big)$$

由于对于每个 $a \in$C(v)，求和并非无穷大，因此对 C(v) 中所有动作的求和也并非无穷大。因此，我们可以将上述方程改写为：

$$v \in \mathsf{inf}(\rho) \implies \Big(\sum_{\{i|\rho_i = v\}} \pi(\rho_{\leq i})(\mathsf{C}(\rho_i)) \neq \infty\Big)$$

我们证明了，对于每个 π-博弈过程 $\rho \in$ plays(π) 和每个状态 $v \in X_1$，这一论证都成立。因此，我们可以得出结论：$\pi \in \Lambda_{\mathrm{COLIVE}}$(C) (7)。
现在，我们将证明，对于每个 $v \in X_i$ where $i \geq 2$，都有 $\pi \in \Lambda$。我们知道，每个 X_i，$i \geq 2$ 都可能包含来自 I 和 $\neg I$的状态。因此，我们将分别讨论这两种情况。首先，从 $\neg I$中的状态开始。

在算法 1 中，我们像处理 Büchi 博弈一样，使用 Apre$_1$(Z，X_{i-1})，为每个 $v \in X_i \cap \neg I$、每个 $i \geq 2$ 定义了 H(v)。因此，利用与定理 7 中相同的论证，我们可以证明对于每个 $v \in X_i \cap \neg I$，$\pi \in \Lambda_{\mathrm{LIVE}}$(H，P)。但是，对于状态 $v \in X_i \cap I$、$i \geq 2$，我们使用 AF 前驱$_1$(Z，Y，X_{i-1}) 定义了 H(v)。因此，我们将给出新的引理，以帮助证明对于每个 $v \in X_i \cap I$，$\pi \in \Lambda_{\mathrm{LIVE}}$(H，P)。

引理 1。 设 $\mathcal{G} = (G, \Diamond\Box I)$ 是一个并发余 Büchi 博弈，状态集合 X_i 和 X_{i-1} 由 ***AlmostCo-Büchi*** (G, I)给出，其中 $i \geq 2$。若对于每个 $v \in X_i \cap I$，$\gamma \subseteq A^v_{X_i}(B^v_{X_{i-1}}(\gamma))$ 成立，则每个满足 $\gamma = \mathit{inf}(\pi(v))$ 的 π，都是博弈 $\mathcal{G}$中玩家1 的获胜策略。

证明。 我们可以将该引理的 if 条件写如下：

$$a^* \in \gamma \implies a^* \in \mathsf{A}^v_{X_i}((\mathsf{B}^v_{X_{i-1}}(\gamma)))$$

然后，根据公式 (1)，我们可以将上述方程改写如下：

$$a^* \in \gamma \implies \forall b \in \Gamma_2(v) \cdot \Big(\delta(v, a^*, b) \notin X_i \implies b \in \mathsf{B}^v_{X_{i-1}}(\gamma)\Big)$$

通过对蕴含式的左侧取否定，我们得到：

$$a^* \in \gamma \implies \forall b \in \Gamma_2(v) \cdot \Big(\delta(v, a^*, b) \in X_i \vee b \in \mathsf{B}^v_{X_{i-1}}(\gamma)\Big)$$

根据方程 (2)，我们可以将上述方程改写如下：

$$a^* \in \gamma \implies \forall b \in \Gamma_2(v) \cdot \Big(\delta(v, a^*, b) \in X_i \vee (\exists a \in \gamma \cdot \delta(v, a, b) \in X_{i-1})\Big)$$

直观地说，上述方程意味着：通过对 γ 中的每个动作赋予正概率，对于每个 $v \in X_i \cap I$，对手都无法采取任何动作来阻止玩家到达 X_{i-1} 并转移到 X_i 的外部。因此，通过对 γ 中的每个动作赋予正概率，策略 π 在 v 中是获胜的。（通过无限多次停留在 X_i 中，如果某个 $v \in X_i \cap \neg I$ 被无限次访问，那么根据与定理 2 相同的论证， X_{i-1} 也将被无限次访问。否则，它将永远停留在 X_i 中，并且只访问 $\neg I$ 有限次，这同样是获胜的。）

引理 2. 设 $\mathcal{G} = (G, \Diamond\Box I)$ 是一个并发余 Büchi 博弈，给定状态集合 X_i 和 X_{i-1}，它们由某个 $i \geq 2$对应的 ***AlmostCo-Büchi*** (G, I)给出。若对每个 $v \in X_i \cap I$都有 $B^v_{X_{i-1}}(\gamma) = \Gamma_2(v)$，那么每个满足 $\gamma = \mathit{inf}(\pi(v))$ 的 π 都遵循模板 $\Lambda_{LIVE}(H, P)$。

证明。 我们可以将引理中的条件写成如下形式：

$$b \in \Gamma_2(v) \implies \Big(b \in \mathsf{B}^v_{X_{i-1}}(\gamma)\Big)$$

然后，根据方程 (2)，我们可以将上面的方程改写如下：

$$b \in \Gamma_2(v) \implies \Big(\exists a \in \gamma \cdot \delta(v, a, b) \in X_{i-1}\Big)$$

由于在算法 1 中，对于每个 $b \in \Gamma_2(v)$，我们都会将 $\{a \in \Gamma_1(v) \mid \delta(v,\ a,\ b) \in X_{i-1}\}$ 添加到 H(v) 中，即 h_b，因此可以得出以下结论：

$$b \in \Gamma_2(v) \implies \Big(\exists a \in \gamma \cdot a \in h_b\Big)$$

根据 inf(π(v))（9）的定义，我们知道 $\sum_{\{i|v_i=v\}} \min^{a\in\gamma\ \pi}(v_0v_1...v_i)(a) = \infty$。

当 γ 中动作的最小概率之和为无穷大时，γ 的每个非空子集中动作的概率之和也为无穷大。因此，根据上述论证，我们可以得出结论：h_b 中动作的概率之和也为无穷大。因此，根据 (6)，我们可以写出：

$$\sum_{\{i|v_i=v\}} \mathsf{minProb}_{\pi(v_0v_1...v_i)}(\{h_b \mid b \in \Gamma_2\}) = \infty$$

这等价于：

$$\sum_{\{i|v_i=v\}} \mathsf{minProb}_{\pi(v_0v_1...v_i)}(\mathsf{H}(v)) = \infty$$

根据 Λ_{LIVE}(H,P) (5) 的定义，我们可以得出 $\pi \in \Lambda_{\text{LIVE}}$(H,P)。

引理 *3.* 设 $\mathcal{G} = (G, \Diamond\Box I)$ 是一个并发余 *Büchi* 博弈，具有一个针对玩家*1*的获胜策略 π，满足 $\gamma = inf(\pi(v))$，且给定状态集合 X_i 和 X_{i-1}，它们由某个 $i \geq 2$ 对应的 *AlmostCo-Büchi* (G, I) 给出。若 $B^v_{X_{i-1}}(\gamma) \neq \Gamma_2(v)$，则存在另一个针对玩家*1*的获胜策略 π'，满足 $\gamma' = inf(\pi'(v))$，使得对于每个 $v \in X_i \cap I$，都有 $B^v_{X_{i-1}}(\gamma') = \Gamma_2(v)$。

证明。 首先，我们定义 $\Gamma_2'(v) := \Gamma_2(v) \setminus \mathrm{B}^v_{X_{i-1}}(\gamma)$，该集合非空，因为这是引理前提。其次，对于每个 $b \notin \mathrm{B}^v_{X_{i-1}}(\gamma)$，对所有 $a \in \gamma$，根据方程 (2)，有 $\delta(v, a, b) \notin X_{i-1}$。但是，$\delta(v, a, b) \in I$，因为 π 是获胜策略；如果它不前往 X_{i-1}，就应该无限多次访问 I。否则，它将永远停留在 X_i，并无限多次访问 $\neg I$，这会导致失败。因此，我们可以写成：

$$\forall b \in \Gamma_2'(v) \cdot \forall a \in \gamma \cdot \Big[\delta(v, a, b) \notin X_{i-1} \wedge \delta(v, a, b) \in I\Big]$$

另一方面，由于 $v \in \mathcal{W}_1$，对于对手的每个动作，玩家都至少应该有一个动作可以到达 X_{i-1}：

$$\forall b \in \Gamma_2'(v) \cdot \exists a \in \Gamma_1(v) \cdot \delta(v, a, b) \in X_{i-1}$$

根据前面的论证，我们知道 $a \notin \gamma$。因此，我们可以将上面的方程改写为：

$$\forall b \in \Gamma_2'(v) \cdot \exists a^* \in \Gamma_1(v) \setminus \gamma \cdot \delta(v, a^*, b) \in X_{i-1}$$

如果我们将所有这些 a^* 添加到 γ 中，就会计算出一个名为 γ' 的新集合：

$$\gamma' := \gamma \cup \{a^* \in \Gamma_1(v) \setminus \gamma \mid \exists b \in \Gamma_2'(v) \cdot \delta(v, a^*, b) \in X_{i-1}\}$$

现在，如果策略 π' 对 γ' 中的每个动作都赋予正概率，那么对每个$v \in X_i \cap I$，inf(π′(v)) $= \gamma'$。由于对每个 $b \in \Gamma_2(v)$，我们都向 γ 中添加了这样的 $a^* \in \Gamma_1(v)$，其中 $\delta(v, a^*, b) \in X_{i-1}$，根据 (2) 的定义，我们可以写成：

$$\mathrm{B}^v_{X_{i-1}}(\gamma') = \Gamma_2(v)$$

因此，我们找到了一个新策略 π'，使得 $\mathsf{B}^{v}_{X_{i-1}}(\gamma') = \Gamma_2(v)$ 对每个 $v \in X_i \cap I$ 都成立。在证明的其余部分，我们需要证明 π' 是博弈 $\mathcal{G}$ 中玩家1的获胜策略。

$$\mathsf{A}^{v}_{X_i}(\mathsf{B}^{v}_{X_{i-1}}(\gamma')) = \mathsf{A}^{v}_{X_i}(\Gamma_2(v))$$

根据公式 (1)，我们知道 $\mathsf{A}^{v}_{X_i}(\Gamma_2(v)) = \Gamma_1(v)$。因此，显然:

$$\gamma' \subseteq \mathsf{A}^{v}_{X_i}(\mathsf{B}^{v}_{X_{i-1}}(\gamma'))$$

最后，利用引理 1，我们可以得出结论： π' 是博弈 $\mathcal{G}$中玩家1的获胜策略。

现在，我们将使用上述引理证明，对于每个 $v \in X_i \cap I$，$\pi \in \Lambda_{\mathrm{LIVE}}$(H,P)。

如果 π 是玩家1在博弈 $\mathcal{G}$中的获胜策略，对于每个 $v \in X_i \cap I$，我们计算 $\gamma = \inf(\pi(v))$。现在，$\mathsf{B}^{v}_{X_{i-1}}(\gamma)$有两种可能：I. $\mathsf{B}^{v}_{X_{i-1}}(\gamma) = \Gamma_2(v)$ II. $\mathsf{B}^{v}_{X_{i-1}}(\gamma) \neq \Gamma_2(v)$

对于情况 I，利用引理 2，我们可以得出结论： $\pi \in \Lambda_{\mathrm{LIVE}}$(H,P)。
对于情况 II，利用引理 3，我们可以为博弈 $\mathcal{G}$中的玩家1找到另一个获胜策略 π'，使得 $\mathsf{B}^{v}_{X_{i-1}}(\gamma') = \Gamma_2(v)$，其中 $\gamma' = \inf(\pi'(v))$。随后，利用引理 2，我们可以得出结论： $\pi' \in \Lambda_{\mathrm{LIVE}}$(H,P)。
现在，我们已经证明，对于每个 $v \in X_i \cap I$，每个获胜策略 π 要么遵循我们的模板，要么可以将其改为一个新的获胜策略 π'，而该策略将遵循我们的模板。

最后，由于我们证明了对于每个 $v \in X_1$, $\pi \in \Lambda_{\mathrm{COLIVE}}$(C) 和每个$v \in X_i$，要么 π，要么一个新的获胜策略属于 Λ_{LIVE}(H，P)，并且对于每个$v \in \mathcal{W}_1$， π 属于 $\Lambda_{\mathrm{UNSAFE}}$(S)，因此可以得出结论：模板 $\Lambda = \Lambda_{\mathrm{UNSAFE}}$(S) $\cap \Lambda_{\mathrm{LIVE}}$(H，P) $\cap \Lambda_{\mathrm{COLIVE}}$(C) 非空。

D 模板的非最大许可性示例

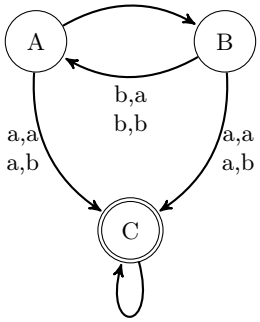


图4：一个布奇博弈，其中 $I = \{C\}$ 是目标集。

例*1*。考虑并发 Büchi 博弈 $\mathcal{G} = (G, \Box\Diamond I)$，其中 G 如图4所示，且 $I = \{C\}$。两个玩家在每个状态下的动作集合均为 $\Gamma_1(v) = \Gamma_2(v) = \{a, b\}$。AlmostBüchi ($I$) 将按如下方式计算集合 X_0，X_1，X_2:

- $X_0 = \emptyset$
- $X_1 = \{C\}$
- $X_2 = \{A, B, C\}$

玩家1的获胜区域为 $\mathcal{W}_1 = X_2$，而恰好一步可到达 C 的状态集合为 $\mathsf{Apre}_1(\mathcal{W}_1, I) = U_1 = \{A, B\}$。在状态 A 和 B，玩家1只需采取“a”即可赢得博弈。然而，假设玩家1改为在状态 A 以概率 1 选择“b”，并在状态 B 于第 n 次访问 B 时以概率 2^{-n} 采取“a”。这仍然是一种获胜策略，因为到达 C 的概率为 0。在这种情况下，我们有一种获胜策略，对于 U_1 中的两个状态，其总和都不

$$\sum_{\{i \mid \rho_i = v\}} \mathsf{minProb}_{\pi_1(\rho_{<i})}(\mathsf{H}(v))$$

等于 ∞。因此，该策略不遵循模板 $\Lambda_{\text{LIVE}}(\mathsf{H}, \mathsf{P})$，而模板 $\Lambda = \Lambda_{\text{UNSAFE}}(\mathsf{S}) \cap \Lambda_{\text{LIVE}}(\mathsf{H}, \mathsf{P})$ 也不是极大的。

例*2*。考虑并发余 Büchi 博弈 $\mathcal{G} = (G, \Diamond\Box I)$，其中 G 如图5所示，且 $I = \{S_0, S_1, S_2, S_3\}$。AlmostCo-Büchi ($I$) 将按如下方式计算集合 X_0，X_1，X_2：

- $X_0 = \{S_0, S_1\}$
- $X_1 = \{S_0, S_1, S_2, S_3\}$
- $X_2 = \{S_0, S_1, S_2, S_3, S_4\} = \mathcal{W}_1$

在此算法中，由于 S_2 属于 AF 前驱$_1(\mathcal{W}_1, X_1, X_0)$，因此已将其添加到 X_1 中。（因为 $\nu\gamma \cdot (\mathsf{A}^{S_2}_{\mathcal{W}_1}(\emptyset) \wedge \mathsf{A}^{S_2}_{X_2}((\mathsf{B}^{S_2}_{X_1}(\emptyset)))$ 等于 $\{a, b, x, y\} \neq \emptyset$。）算法 1 将按如下方式计算 S_2 的 H:

- $d \to \{a, y\}$
- $e \to \{b\}$
- $f \to \{x\}$

然后，$\mathsf{H}(S_2) = \{\{a, y\}, \{b\}, \{x\}\}$。
在这个博弈中，玩家1可以在 S_2 中以等概率进行博弈过程 $\{a, b\}$，并赢得该博弈。该策略不符合算法 1 的模板 $\Lambda_{\text{LIVE}}(\mathsf{H}, \mathsf{P})$，但它仍然是获胜策略，因此该模板并非极大。

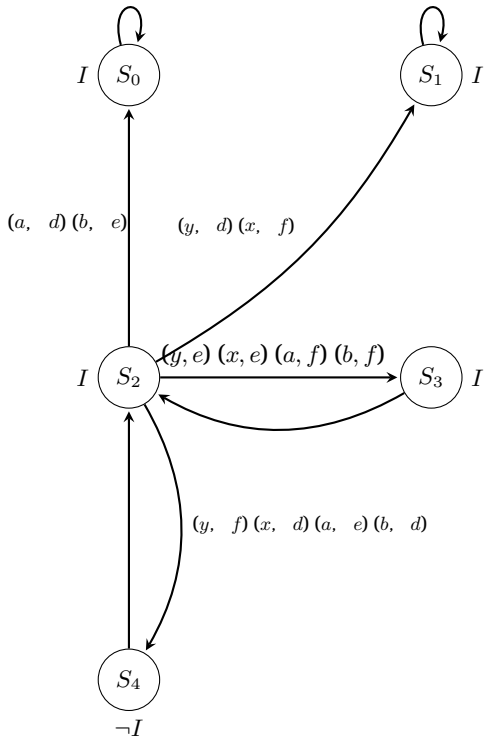


图 5：满足获胜条件 $\Diamond\Box I$ 的余 Büchi 博弈 $\mathcal{G}_2$ ，其中 $I = \{S_0, S_1, S_2, S_3\}$

E 实验的补充材料

E.1 将轮流制博弈转换为并发博弈

我们将轮流制博弈转换为并发博弈的步骤如下：

1. **转移合并**：对于并发博弈，我们保留玩家1的状态，并将轮流制博弈中每两个连续转移合并为并发博弈中的一个转移。具体而言，对于轮流制博弈中的每对连续转移 (u, a, v) 和 (v, b, w) ，其中 u 是玩家1的状态， v 是玩家2的状态，我们在并发博弈中创建一个转移 $(u, (a, b)\ w)$ 。随后，我们移除玩家2的状态。对于没有出边的状态，我们添加自环。请注意，这是可行的，因为我们考虑的是交替轮流制博弈。

2. **获胜条件转换**：我们的工具要求使用基于状态的获胜条件。对于任何具有入边转移 $(u,(a, b)\ v)$ 的状态 v ，如果原博弈中的获胜条件包含至少一个原始转移 $((u, a, v)$ 或 $(v, b, w))$ ，则会将该状态加入并发博弈的目标集合。

我们要指出，这一过程对于大型实例并不实用，因此仅应用于较小的博弈。排除无法处理的超大型实例后，我们选取了 171 个实例进行实验评估。所考虑（并转换）的模型列表见 [10]。

E.2 ConSTel 与 PeSTel 对比

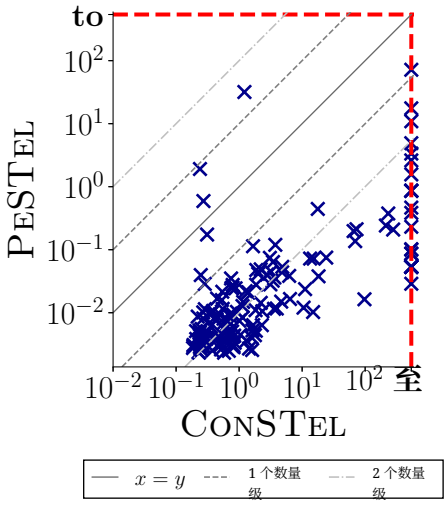


图 6：运行时比较

我们将原型工具 ConSTel 与现有工具 PeSTel 进行了比较。[7], PeSTel 用于计算轮流制奇偶博弈的策略模板。然而，由于 PeSTel只能处理轮流制博弈，我们采用了以下设置：对于转换后的 171 个博弈，我们分别使用 ConSTel 计算模板，并使用 PeSTel 在相应的原始轮流制博弈上计算模板。这样，我们便能更好地衡量 ConSTel 的性能，并通过实证研究转向并发设置的难度。结果汇总在图 6 的散点图中。图中的每个点（x，y）对应一个实例，其中 x 和 y 分别表示并发版 PeSTel 和 PeSTel 的运行时。对于较大的实例，PeSTel 的表现往往比我们的工具好几个数量级（OoM）。这凸显了相较于轮流制博弈，求解并发博弈所固有的复杂性有所增加，也表明要让我们的工具扩展到更大的实例，还需要进一步改进工程实现。

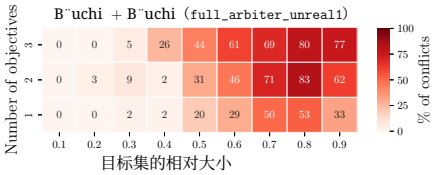
E.3 增量综合

在第 7 节中，我们在增量综合的背景下考察了策略模板。下面我们将详细介绍所考察的模型：

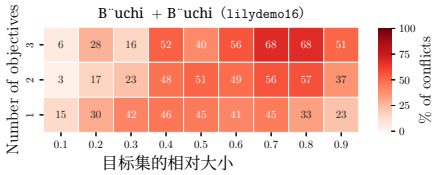
– full_arbiter_unreal1：55 个状态和 731 次转移– lilydemo16：28 个状态和 1228 次转移– full_arbiter：55 个状态和 731 次转移– amba_decomposed_tincr：26 个状态和 289 次转移

图2汇总了这些选定模型的数据。下文将在图7至图9中分别展示各个博弈中的冲突。热图中的每个单元格（s， k）对应于添加大小为 s的 k 个目标时发生的冲突百分比。

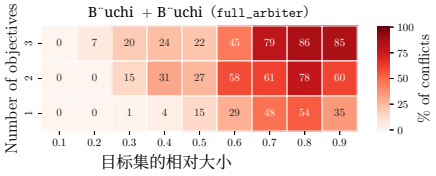
一个普遍趋势是，随着添加的目标增多，冲突的数量也会增加。这是意料之中的，因为需要组合更多模板，从而增加了发生冲突的可能性。另一个有趣的观察是，中等规模到较大规模的目标集合似乎更容易发生冲突。一种可能的解释是，对于这些规模，我们会得到更大的获胜区域，因此发生冲突的可能性也更高。关于组合目标的类型，我们观察到，Büchi ＋ Büchi 的组合似乎最不容易发生冲突。引入新的 co-Büchi 目标会导致更多冲突。直观而言，这是因为 co-Büchi 目标的模板（与 Büchi 目标的模板相比）还包含一个 co-live 模板。



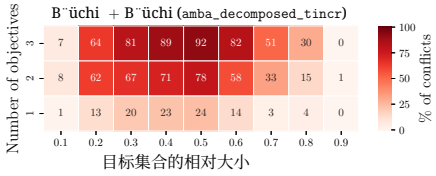
(a) full_仲裁器_unreal1



(b) lilydemo16

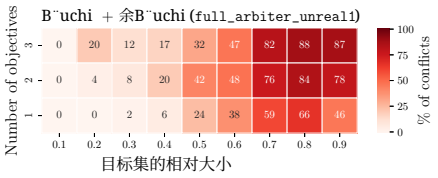


(c) full_arbiter

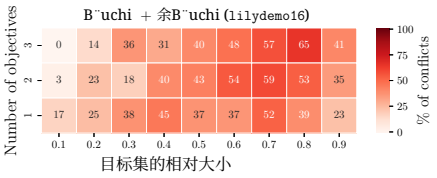


(d) amba_decomposed_tincr

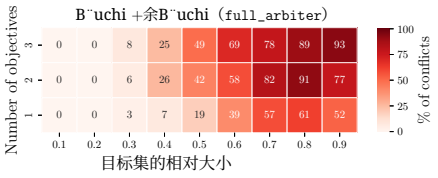
图 7：各博弈中 Büchi ＋ Büchi 的冲突分析。



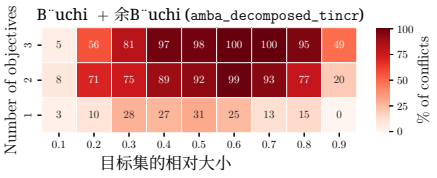
(a) 完全_仲裁者_非实现1



(b) lilydemo16

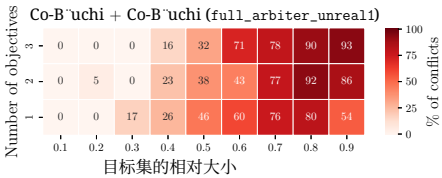


(c) 完全_仲裁器

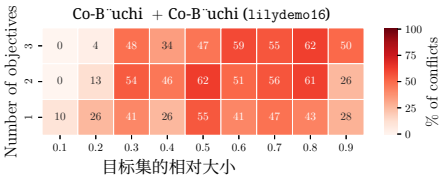


(d) amba_分解_tincr

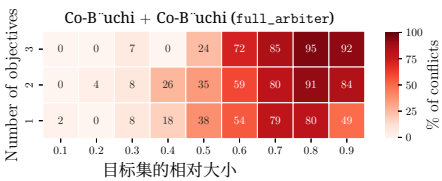
图 8：每个博弈中 Büchi ⊓ 余 Büchi 的冲突分析。



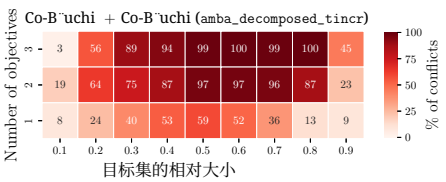
(a) 完全_仲裁器_unreal1



(b) lilydemo16



(c) 完整_仲裁器



(d) amba_分解后的_tincr

图9：每个博弈的余 Büchi ⊓ 余 Büchi 冲突分析。