arXiv:2601.10651v1 [cs.AI] 15 Jan 2026

多属性综合

Christoph Weinhuber[1, *], **Yannik Schnitzer**[1, *], **Alessandro Abate**[1], **David Parker**[1], **Giuseppe De Giacomo**[1], **Moshe Y. Vardi**[2]
[1]牛津大学，牛津，英国
[2]莱斯大学，休斯敦，得克萨斯州，美国

摘要

我们研究具有多个性质的线性时序逻辑$_f$ 合成问题，在这类问题中，可能无法同时满足所有性质。我们不再枚举性质的子集，而是在一次不动点计算中求出乘积博弈状态与可从这些状态实现的目标集合之间的关系，并合成能够实现极大可实现集合的策略。我们提出一种完全符号化的算法，通过引入布尔目标变量并利用单调性，紧凑地表示指数级数量的目标组合。与基于枚举的基线方法相比，我们的方法性能大幅提升，加速比最高可达两个数量级。

1 引言

基于时序规范进行合成，是人工智能中许多决策问题的核心，尤其是自动规划和控制器合成。在这种设定下，我们自动构造一种智能体策略，使其能够确保无论环境采取何种可接受行为，都满足给定的时序规范。线性时序逻辑（LTL）长期以来一直是表达无限时域下时序扩展目标和假设的标准形式体系，尤其应用于反应式合成 [Pnueli and Rosner，1989]。

相比之下，许多现实世界中的 AI 任务本质上是有限时域任务：计划旨在终止，任务具有明确的完成概念，并且一旦达到期望的条件，就算取得成功。LTL 的有限轨迹变体记作 LTL$_f$，已成为在此类定义域中描述目标的一种自然且富有表现力的形式化方法。近年来，LTL$_f$ 合成已发展为一种标准工具，可为有限时域任务生成正确性由构造保证的行为 [De Giacomo and Vardi, 2013; De Giacomo and Vardi,2015]。典型应用领域包括经典规划与条件规划 [De Giacomo and Rubin, 2018; Camacho 等，2018], 服务组合 [Montali 等，2010], 以及业务流程建模 [Pesic and van der Aalst, 2006; De Giacomo 等，2014], 其目标并非无限期地作出响应，而是达成令人满意的结果并停止。

尽管现有的线性时序逻辑$_f$ 合成技术非常适合对单一、整体式规范进行推理，但它们在很大程度上受限于全有或全无范式：该过程要么成功合成一个满足整个规范的策略，要么完全无法生成策略，并判定问题无解 [Duret-Lutz 等，2025]。

然而，在许多现实场景中，例如机器人学、多服务编排，或受资源限制的规划 [van den Briel等, 2004年; Baier and McIlraith,2008], 智能体会面临超额订阅 [Smith，2004]。也就是说，规范通常自然地分解为多个属性（或目标），但由于冲突要求、有限资源或对抗性环境选择，同时满足所有属性可能是不可能的。

在这类情形下，全有或全无范式过于严格，因为失败很少是非黑即白的。即使所有目标的合取无法实现，智能体仍可能能够保证满足其中某个有意义的子集。我们研究一个根本性问题：如果无法同时实现所有目标，那么哪些子集可以实现？其中哪些最值得追求？回答这一问题对于在过度订阅领域中进行知情权衡和有原则的决策至关重要。

解决这一问题的一种直接方法是枚举所有目标子集，并针对每个子集求解其元素的合取所对应的综合问题 [Agrawal等人， 1993]。然而，即使性质数量适中，这种方法也可能变得不可行，因为它需要求解数量呈指数级增长的综合实例。

我们提出了一种多性质综合技术，可同时考虑所有目标组合。我们的方法通过一次单次不动点计算，求出状态与可从该状态同时实现的目标子集之间的完整关系。我们不再逐个检查子集是否可实现，而是通过一次计算，确定哪些目标组合能够在所有可容许的环境行为下得到保证。

至关重要的是，我们的方法不仅能决定哪些目标子集能够同时实现，还能合成实现这些目标的策略。智能体与环境交互，直到达到这样一种状态：最理想的可实现目标集合已达成，此时

*作者贡献相同。[1]firstname.lastname@cs.ox.ac.uk, [2] vardi@cs.rice.edu

它便可以停止。构建策略的目的不是让其无限期地做出反应，而是确保朝着满足期望目标的方向持续推进，并在这一目标达成时停止。

除了不动点刻画之外，我们还开发了一种用于多属性计算的完全符号化技术。我们利用布尔综合 [Fried 等人， 2016] 以紧凑的符号表示进行不动点计算，这通常比显式构造 [Zhu 等人， 2017]高效得多。请注意，尽管问题定义域涵盖所有目标子集，但其结构具有单调性——实现一组目标意味着也能实现其中任意子集。我们的符号编码旨在直接利用这一结构，使单一布尔表示能够紧凑地刻画指数级数量的目标组合，而无需显式枚举它们。

总之，本文作出以下贡献：

1. 我们提出一种用于多属性线性时序逻辑f 合成的技术，能够刻画过度订阅领域中任意目标子集的可实现性。 2. 我们提出一种不动点刻画，通过单次构造计算所有最大可实现目标集及实现这些目标集的策略。 3. 我们开发了一种符号综合算法，能够以单一紧凑的布尔形式刻画指数级数量的目标组合。 4. 我们通过实验证明，与当前最先进的基于枚举的方法相比，我们的方法性能显著更优，最高可达两个数量级。

2 相关工作

多属性验证。 在验证领域，针对固定模型分析多个性质也已有研究。Cabodi 和 Nocco（2011）研究了如何检查同一电路上的多个不变式，并展示了如何利用性质之间的共享结构来减少重复工作。由于验证查询彼此独立，一个性质成立并不会以牺牲另一个性质为代价，因此主要挑战在于摊销重复的证明工作。相比之下，在我们的合成设定中，性质可能会相互竞争：强制满足一个目标可能会限制可用行为，从而妨碍满足另一个目标。

在概率验证中，针对马尔可夫决策过程（MDP）的多目标模型检查会询问是否存在一种策略，能够同时满足多个性质各自的满足概率界限，并通过计算帕累托前沿来支持权衡分析 [Etessami等人， 2007]。我们的多属性综合可视为定性、对抗性设定下的类似权衡分析：我们得到的不是概率阈值上的连续帕累托前沿，而是由极大可实现集合构成的离散前沿。

最大可实现性综合 Dimitrova 等人（2018年）研究具有硬规范和软安全要求的无限轨迹 线性时序逻辑合成，其中软安全要求可能无法在单次执行中全部得到满足。他们采用多值定量语义，对满足以下条件的无限执行中各项软性质的满足程度进行分级：满足硬约束 [Dimitrova 等人， 2020]。他们的方法由优化驱动：基于有界综合，在控制器大小上限内寻找能够使该定量得分最大化的控制器 [Dimitrova 等人， 2018]。

在我们的设定中，给定多个线性时序逻辑f 目标，我们计算哪些子集能够同时实现，并给出满足这些目标的策略。我们利用线性时序逻辑f 合成可归约为确定有限自动机可达性博弈 [De Giacomo 和 Vardi， 2015], 其中，满足条件意味着到达接受状态。由此可通过局部接受基础情形进行直接的不动点计算，在一次构造中计算所有状态与目标组合的可实现性关系。

超额订阅与偏好规划。 超额订阅和部分满足规划针对的是受限条件导致无法实现所有目标的情形，它们通过启发式方法或效用函数选择有价值的子集 [Smith，2004年; van den Briel 等， 2004], 或优化用户指定的偏好 [Baier 和 McIlraith，2008；Bienvenu 等，2006；Sohrabi等， 2009]。然而，这些方法针对非对抗性 模型，只生成单个计划。与之不同，我们合成的策略能够针对所有可容许的环境行为满足目标集合。

3 预备知识

本节回顾有限轨迹线性时序逻辑的基础知识，以及将合成问题标准地归约为确定有限自动机可达性博弈的方法。

3.1 有限轨迹线性时序逻辑

有限轨迹线性时序逻辑（LTLf）与 LTL [Pnueli，1977] 具有相同的语法。给定一组原子命题 AP，LTLf 公式的语法定义如下：

$$\varphi ::= \text{true} \mid a \in \mathsf{AP} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi.$$

轨迹是一个有限 序列 $\rho = \rho[0]\rho[1]\cdots\rho[m]$，其中每个 $\rho[i] \in 2^{\mathsf{AP}}$ 指定在时刻 i 哪些原子命题成立。我们用 $|\rho| := m+1$ 表示 ρ 的长度。我们使用满足关系 $\rho, i \models \varphi$ 表示 φ 在 ρ 的位置 i 成立。满足关系定义如下：

$$\begin{array}{lll}
\rho, i \models a & \iff & a \in \rho[i], \\
\rho, i \models \neg\varphi & \iff & \rho, i \not\models \varphi, \\
\rho, i \models \varphi_1 \wedge \varphi_2 & \iff & \rho, i \models \varphi_1 \text{ and } \rho, i \models \varphi_2, \\
\rho, i \models \mathsf{X}\,\varphi & \iff & i+1 < |\rho| \text{ and } \rho, i+1 \models \varphi, \\
\rho, i \models \varphi_1\,\mathsf{U}\,\varphi_2 & \iff & \exists j.\ i \le j < |\rho| \text{ s.t. } \rho, j \models \varphi_2 \\
 & & \text{and } \forall k.\ i \le k < j,\ \rho, k \models \varphi_1.
\end{array}$$

轨迹 ρ 满足 φ，记作 $\rho \models \varphi$，当且仅当 $\rho, 0 \models \varphi$。

3.2 线性时序逻辑f 合成与确定有限自动机博弈

线性时序逻辑f 性质的合成，是指计算一种智能体策略，使其能够针对环境所有允许的行为强制满足时序规范。我们将原子命题集 AP 划分为两个不相交的集合：输入 $\mathcal{X}$ 由环境控制，输出 $\mathcal{Y}$ 由智能体控制。执行是一条有限轨迹 ρ，其生成方式如下：在每个时刻 i，智能体先选择一个赋值 $Y_i \in 2^{\mathcal{Y}}$，然后环境作出响应

并给出一个赋值 $X_i \in 2^{\mathcal{X}}$。由此得到的组合赋值 $(X_i,\ Y_i)$ 构成轨迹中的第 i 个位置 $\rho[i]$。

一个智能体策略是一个函数 $g : (2^{\mathcal{X}})^* \to 2^{\mathcal{Y}}$，它将观测到的输入的（可能为空的）历史记录 $X_0X_1\cdots X_{i-1}$ 映射为下一步输出选择 $Y_i = g(X_0\cdots X_{i-1})$。可实现性问题要求确定是否存在这样一种策略 g，使得与 g 一致的每条有限轨迹都满足 φ。合成问题还要求构造一个作为见证的获胜策略。由于 LTL$_f$ 是在有限轨迹上解释的，获胜策略必须确保对局到达某个位置 i，使得轨迹 $\rho = (X_0 \cup Y_0)\ldots(X_i \cup Y_i)$ 满足 φ，随后即可终止。

定义 1（LTL$_f$ 合成）。设 φ 为一个关于原子命题 $AP = \mathcal{X} \cup \mathcal{Y}$ 的 LTL$_f$ 公式。给定一个无限输入序列 $\xi = X_0X_1X_2\cdots \in (2^{\mathcal{X}})^\omega$ 和一个策略 g，截至时域 k 所诱导的有限轨迹 $\rho_k(g,\xi)$ 定义如下：

$$(g(\epsilon) \cup X_0)(g(X_0) \cup X_1)\cdots(g(X_0\cdots X_{k-1}) \cup X_k).$$

我们称 φ 是可实现的，如果存在一个策略 g，使得对于任意 $\xi \in (2^{\mathcal{X}})^\omega$，都存在某个 $k \geq 0$ 满足 $\rho_k(g,\xi) \models \varphi$。合成问题是构造这样一个获胜策略。

请注意，与基于无限轨迹的 LTL 反应式合成不同，智能体和环境不会无限期地交互：获胜策略确保交互生成一条满足 φ 的有限轨迹，然后选择在该点停止。

LTL$_f$ 合成的标准方法是求解相应的确定有限自动机博弈 [De Giacomo 和 Vardi, 2013]。给定一个关于 AP $= \mathcal{X} \cup \mathcal{Y}$ 的 LTL$_f$ 公式 φ，我们可以构造一个在字母表 $2^{\mathcal{X}\cup\mathcal{Y}}$ 上的确定有限自动机 $\mathcal{G}_\varphi$，它恰好接受满足 φ 的有限轨迹。于是，合成问题归结为求解在 $\mathcal{G}_\varphi$ 上进行的可达性博弈，也称为博弈场。

定义 2(确定有限自动机博弈场)。确定有限自动机博弈场是一个元组 $\mathcal{G} = (2^{\mathcal{X}\cup\mathcal{Y}}, S, s_0, \delta, F)$，其中 $2^{\mathcal{X}\cup\mathcal{Y}}$ 是博弈的字母表，S 是一个有限状态集，$s_0 \in S$ 是初始状态，$\delta:\ S \times 2^{\mathcal{X}\cup\mathcal{Y}} \to S$ 是转移函数，而 $F \subseteq S$ 是终态集。

DFA 博弈按回合进行：在第 i 回合，博弈处于状态 $s_i \in S$，智能体选择一个输出赋值 $Y_i \in 2^{\mathcal{Y}}$，然后环境选择一个输入赋值 $X_i \in 2^{\mathcal{X}}$，接着博弈转移到 $s_{i+1} := \delta(s_i,\ Y_i \cup X_i)$。如果一局对局到达属于 F 的终止状态，则智能体获胜，此时可以认为对局已经结束。

由于 $\mathcal{G}_\varphi$ 恰好识别满足 φ 的有限轨迹，因此，在自动机中到达终止状态 F 等价于生成了一个满足 φ 的有限前缀。因此，当且仅当对于每个无限输入序列 $\xi \in (2^{\mathcal{X}})^\omega$，诱导对局都在有限回合后访问 F 中的某个状态时，智能体策略 $g: (2^{\mathcal{X}})^* \to 2^{\mathcal{Y}}$ 才是 DFA 博弈中的获胜策略。

为了刻画智能体能够从哪些状态强制获胜，我们使用标准的可控前驱算子 [De Giacomo和 Vardi, 2015]：对于集合 $\mathcal{E} \subseteq S$，我们定义

$$\mathsf{PreC}(\mathcal{E}) := \{s \in S \mid \exists Y \in 2^{\mathcal{Y}}.\forall X \in 2^{\mathcal{X}}.\delta(s, Y \cup X) \in \mathcal{E}\}.$$

直观地说，PreC($\mathcal{E}$) 包含这样的状态：从这些状态出发，智能体可以一步移动，并保证无论环境如何行动，都能到达 $\mathcal{E}$。获胜状态集合是反复添加此类前驱状态所得到的最小不动点，起始于接受状态：

$\mathsf{Win}_0 := F$ 和 $\mathsf{Win}_{i+1} := \mathsf{Win}_i \cup \mathsf{PreC}(\mathsf{Win}_i)$。由于 S 是有限的，该序列至多经过 $|S|$ 次迭代便会在不动点 $\mathsf{Win} := \bigcup_{i\geq 0}\mathsf{Win}_i$ 处稳定，从而实现关于博弈状态数量的线性时间计算。当且仅当 $s_0 \in \mathsf{Win}$ 时，智能体才有获胜策略。

计算出获胜区域 Win 后，我们便可以为每个获胜状态 $s \in \mathsf{Win}$ 选择一个合适的智能体输出 $Y \in 2^{\mathcal{Y}}$，从而提取具体的获胜策略。形式上，我们选择一个函数 $\omega: \mathsf{Win} \to 2^{\mathcal{Y}}$，使得对于每个满足 $i > 0$ 的 $s \in \mathsf{Win}_i$，

$$\forall X \in 2^{\mathcal{X}}.\ \delta(s, \omega(s) \cup X) \in \mathsf{Win}_{i-1}.$$

根据 Win 的构造，这样的选择必然存在 [De Giacomo 和 Vardi, 2013]。对于状态 $s \in \mathsf{Win}_0 = F$，智能体会选择一个新的 完成 动作，表示其终止该对局。获胜策略可以按照变换器的标准形式，表示为确定性有限状态控制器：

$$T = (2^{\mathcal{X}\cup\mathcal{Y}}, Q, q_0, \eta, \omega),$$

其中 $Q := \mathsf{Win}$，$q_0 := s_0$，输出函数是所选的 ω，转移函数为 $\eta(q,\ X) := \delta(q,\ \omega(q) \cup X)$。对于输入序列 $X_0X_1\cdots$，变换器输出 $Y_i := \omega(q_i)$ 并更新 $q_{i+1} := \eta(q_i,\ X_i)$，从而在博弈 G 中生成一个获胜博弈过程。因此，该变换器实现了一个具体的获胜策略。

4 多属性线性时序逻辑$_f$ 合成

我们将标准的合成概念从单个性质扩展到性质集合。我们的目标是构造一个单一策略，使其能够在应对所有输入行为时，强制对局生成一条满足多个性质的轨迹。

定义 3(多属性线性时序逻辑$_f$ 合成)。设 $\Phi = \{\varphi_1, \ldots, \varphi_n\}$ 为同一组原子命题 $AP = \mathcal{X} \cup \mathcal{Y}$ 上的一组 n 线性时序逻辑$_f$ 性质（或目标）。若存在一个策略 $g: (2^{\mathcal{X}})^* \to 2^{\mathcal{Y}}$，使得对于每个输入序列 $\xi = X_0X_1\cdots \in (2^{\mathcal{X}})^\omega$，都存在一个 $k \geq 0$，使得诱导出的有限轨迹 $\rho_k(g,\xi) \models \varphi_i$ 对所有 $\varphi_i \in C$ 均成立，则目标集 $C \subseteq \Phi$ 是可实现的。多属性综合问题是构造这样一个获胜策略。

我们将 DFA 博弈方法从针对单个性质的 LTL$_f$ 合成扩展到针对一组性质的合成，方法是构造一个能够同时跟踪所有目标进展情况的博弈场。我们考虑一组固定的 nLTL$_f$ 性质（或目标）$\Phi = \{\varphi_1, \ldots, \varphi_n\}$，它们基于相同的原子命题 AP $= \mathcal{X} \cup \mathcal{Y}$。对于每个目标 $\varphi_i \in \Phi$，令 $\mathcal{G}_i = (2^{\mathcal{X}\cup\mathcal{Y}},\ S_i,\ q_i^0,\ \delta_i,\ F_i)$ 为一个确定有限自动机，它恰好识别满足 φ_i 的有限轨迹。

定义 4（多性质博弈场）。该多性质博弈场由 Φ 的笛卡尔积构成：

$$\mathcal{G}^\times = (2^{\mathcal{X}\cup\mathcal{Y}},\ S^\times,\ s_0^\times,\ \delta^\times),$$

其中，$S^\times := \times_{i=1}^n S_i$ 是乘积状态的集合，$s_0^\times := (q_1^0,\ \ldots,\ q_n^0)$ 是初始乘积状态，转移函数按分量定义如下

$$\delta^\times\big((q_1,\ldots,q_n), a\big) := \big(\delta_1(q_1,a),\ldots,\delta_n(q_n,a)\big).$$

请注意，$\mathcal{G}^\times$ 并没有指定的终态集合：在多属性情形下，不存在单一的

完成概念，因为一个乘积状态可能见证某些目标得到满足，却无法见证其他目标得到满足。基于枚举的方法会为某个子集$\Phi' \subseteq \Phi$确定一个目标合取式，由此导出相应的接受条件，然后针对每种选择 Φ'分别求解一个综合博弈，并以指数级的次数重建乘积竞技场。相比之下，由于我们的目标是同时计算所有可实现的子集，因此不会将 $\mathcal{G}^\times$ 限制在单一接受条件上。接下来，我们将提出一种多属性综合过程，它在单一乘积竞技场 $\mathcal{G}^\times$ 上运行，同时确定对于所有 $\Phi' \subseteq \Phi$，从每个乘积状态出发哪些目标组合是可实现的，并综合出能够实现这些目标组合的策略。

状态 $s = (q_1, \ldots\ldots, q_n) \in S^\times$ 由各个自动机的分量状态组成，每个性质 φ_i对应一个分量。因此，当所有相应分量都处于接受状态时，目标集合 $C \subseteq \Phi$ 便在 s 处同时得到满足。

定义 5（多属性满足）。设 $s = (q_1, \ldots\ldots, q_n) \in S^\times$，并设 $C \subseteq \Phi$。若对每个性质 $\varphi_i \in C$ 都有 $q_i \in F_i$，则记作 $s \models C$。

若存在策略 $g\colon (2^{\mathcal{X}})^* \to 2^{\mathcal{Y}}$，使得对于每个输入序列 $\xi = X_0 X_1 \cdots \in (2^{\mathcal{X}})^\omega$，对局 $s_0 := s$ 和 $s_{t+1} := \delta^\times(s_t, g(X_0 \cdots X_{t-1}) \cup X_t)$ 都能到达某个满足 $s_k \models C$ 的 k，则称目标集 $C \subseteq \Phi$ 可由状态 $s \in S^\times$ 实现。此时，策略可以停止对局；由于每个 $\mathcal{G}_i$ 都会识别满足 φ_i 的轨迹，因此，到达满足 $s_k \models C$ 的乘积状态 $s_k \in S^\times$ 意味着所产生的有限轨迹被每个用于 $\varphi_i \in C$ 的自动机 $\mathcal{G}_i$ 接受，从而同时满足 C 中的所有目标。

4.1 求解多属性博弈

我们介绍一种方法，用于计算作用于对 (s, C) 上的多属性获胜关系，其中 $s \in S^\times$ 且 $C \subseteq \Phi$。该关系精确刻画了对于每个乘积状态 s，哪些目标集合 C 是可实现的，并使我们能够为任意此类 C 提取获胜策略。与单个性质的情况一样，该关系通过不动点计算得到：将获胜区域从状态提升到状态—性质集对。其基础情形由满足以下条件的 (s, C) 构成：C 中的所有目标在 s处均已满足；否则，智能体必须选择一个输出 $Y \in 2^{\mathcal{Y}}$，使得对于所有 $X \in 2^{\mathcal{X}}$，智能体都能确保从任意后继状态同时实现 C 中的所有性质。如此一来，所有目标集合在所有状态上的可实现性便得以传播。

对于一个关系 $\mathcal{E} \subseteq S^\times \times 2^\Phi$，我们将可控多属性前驱定义为：

$$\mathsf{PreMC}(\mathcal{E}) := \{(s, C) \mid \exists Y \in 2^{\mathcal{Y}}\ \forall X \in 2^{\mathcal{X}}.\ (\delta^\times(s, Y \cup X), C) \in \mathcal{E}\}.$$

直观而言，PreMC($\mathcal{E}$)包含这样的状态对（s，C）：从这些状态对出发，智能体可以执行一步操作，从而保证同一目标集 C 仍然可实现，而不受环境输入选择的影响。随后，通过从当前乘积状态中 C 已经成立的基础状态对开始，不断加入这样的前驱，便可得到多属性获胜关系的最小不动点：

$$\mathsf{WinM}_0 := \{(s, C) \mid s \models C\},$$
$$\mathsf{WinM}_{i+1} := \mathsf{WinM}_i \cup \mathsf{PreMC}(\mathsf{WinM}_i).$$

请注意，WinM_0 可以在线性时间内计算：因为检查 $s \models C$ 只需查看状态 $s = (q_1, \ldots\ldots, q_n)$ 的哪些分量是接受状态 $q_i \in F_i$，从而确定哪些目标集 C 已在 s处满足。由于 $S^\times$ 和Φ 都是有限的，该序列至多经过 $|S^\times| \cdot 2^{|\Phi|}$次迭代便会在不动点 $\mathsf{WinM} := \bigcup_{i \geq 0} \mathsf{WinM}_i$处稳定。

定理 1。不动点 WinM刻画了多属性可实现性：对于所有 $s \in S^\times$ 和 $C \subseteq \Phi$，

$$(s, C) \in \mathsf{WinM} \iff C \textit{ is realizable from } s.$$

关系 WinM 推广了传统的获胜区域：对于任意目标集 $C \subseteq \Phi$，存在一个获胜策略来满足 C，当且仅当 $(s_0^\times, C) \in \mathsf{WinM}$。将获胜关系投影到所需的性质集合 C 上：

$$\mathsf{WinM}_i|_C := \{\, s \in S^\times \mid (s, C) \in \mathsf{WinM}_i \,\}.$$

与第 3 节讨论的单属性情形一样，我们可以构造相应的获胜策略，将其表示为一个变换器 T_C：选择 $\omega_C : \mathsf{WinM}|_C \to 2^{\mathcal{Y}}$，使得对于每个满足 $i > 0$ 的 $s \in \mathsf{WinM}_i|_C$，

$$\forall X \in 2^{\mathcal{X}}.\ \delta^\times(s, \omega_C(s) \cup X) \in \mathsf{WinM}_{i-1}|_C,$$

并且对于 $s \in \mathsf{WinM}_0|_C = \{s \mid s \models C\}$，输出一个指定的完成动作并终止。根据 WinM 的构造和 PreMC 的定义，这样的选择是存在的。

定理 2. 对于任意满足 $(s_0^\times, C) \in \mathsf{Win}^M$ 的 $C \subseteq \Phi$，上述构造得到的变换器 T_C 存在，并且实现一个满足 C 的获胜策略：对于每个输入序列 $\xi \in (2^{\mathcal{X}})^\omega$，由 T_C 诱导的对局会在有限步后到达一个满足 $s \models C$ 的状态$s \in S^\times$，并且生成的有限轨迹满足每个 $\varphi_i \in C$。

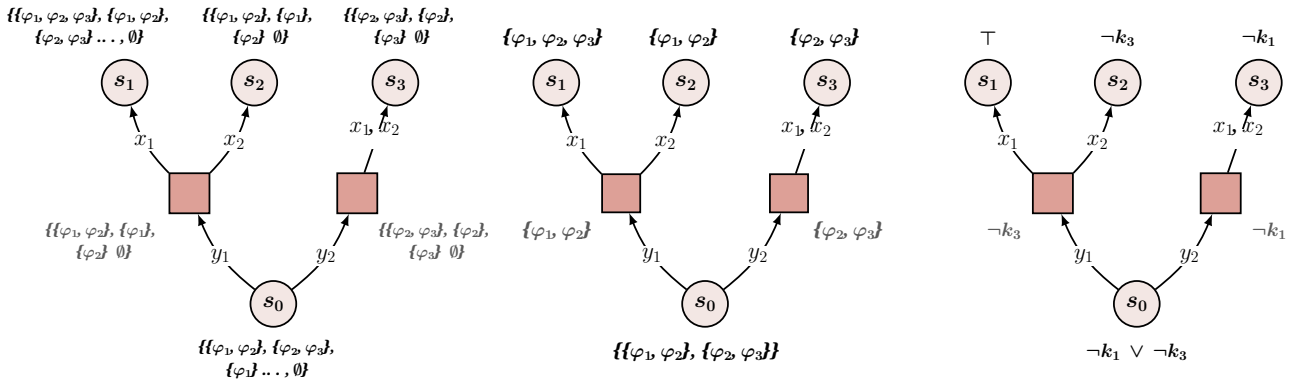
4.2 最大多属性综合

上述构造避免了针对每个 $C \subseteq \Phi$ 显式求解一个单独的综合实例，但关系 WinM $\subseteq S^\times \times 2^\Phi$ 仍可能很大：在最坏情况下，单个状态可能与指数多个性质集合相关联。尽管如此，多属性可实现性具有内在的单调性：如果 $(s, C) \in \mathsf{WinM}$，那么对于每个$D \subseteq C$，$(s, D) \in \mathsf{WinM}$。因此，WinM 中的许多元素都是冗余的，且被更大的可实现集合所包含。因此，自然的做法是，对于每个状态 s，仅保留极大 可实现集合。

我们用一个算子来描述这种压缩，该算子会移除在同一状态下被严格更大的性质集合所包含的所有对。对于关系 $\mathcal{E} \subseteq S^\times \times 2^\Phi$，我们定义

$$\mathsf{Max}(\mathcal{E}) := \{(s, C) \in \mathcal{E} \mid \neg\exists (s, D) \in \mathcal{E} \text{ with } C \subset D\}.$$

因此，Max($\mathcal{E}$)恰好保留那些性质集合 C 相对于 $\subseteq$ 同一状态所关联的所有集合而言是最大的配对。在下文中，我们在不动点计算中应用 Max，使整个过程中只维护最大的可实现集合，而所有更小的可实现集合则保持隐式。然而，一旦我们只保留最大的集合，前驱步骤就不必要求同一集合 C 为每个后继显式存储。相反，只需每个后继状态都存在某个可实现集合 D 能够覆盖 C，即 $C \subseteq D$，因为 D的可实现性



(a) 多属性。 **(b)** 最大多属性。 **(c)** 符号。

图1：展示了我们在一个包含 $2^{\mathcal{Y}} = \{y_1, y_2\}$、 $2^{\mathcal{X}} = \{x_1, x_2\}$ 和性质 $\Phi = \{\varphi_1, \varphi_2, \varphi_3\}$ 的问题上执行综合过程的示意图。(a) 和 (b) 展示了投影到状态上的结果获胜关系 WinM：每个状态都标注了从该状态出发可实现的目标集合。(a) 展示了显式的多性质不动点，它跟踪所有可实现的目标集合（第 4.1 节）；(b) 展示了最大变体（第 4.2 节）。(c) 展示了符号变体（第 5 节），其中获胜关系以布尔公式的形式紧凑表示。

意味着其所有子集均可实现。因此，我们将 $\mathcal{E} \subseteq S^{\times} \times 2^{\Phi}$ 的最大多属性前驱定义为

$$\mathsf{PreMMC}(\mathcal{E}) := \{(s, C) \mid \exists Y \in 2^{\mathcal{Y}}\ \forall X \in 2^{\mathcal{X}}\ \exists D \supseteq C.\ (\delta^{\times}(s, Y \cup X), D) \in \mathcal{E}\ \}.$$

直观地说，PreMMC($\mathcal{E}$) 恰好包含这样的状态对 (s，C)：从这些状态对出发，无论环境输入是什么，智能体都可以强制转移到一个后继状态，并在该状态下确保实现某个 $D \supseteq C$，因此也能确保实现 C。

我们不再先构造 WinM、再剪除冗余的目标集合，而是在整个过程中只保留那些对应状态下性质集为最大的状态对。由此得到以下最大不动点迭代:

$$\mathsf{WinM}_0 := \mathsf{Max}(\{(s, C) \mid s \models C\}),$$
$$\mathsf{WinM}_{i+1} := \mathsf{Max}\big(\mathsf{WinM}_i \cup \mathsf{PreMMC}(\mathsf{WinM}_i)\big),$$

并令 $\mathsf{WinM} := \bigcup_{i \geq 0} \mathsf{WinM}_i$ 表示所得的不动点。与第 4.1 节中的构造不同，后者可能会累积许多被支配的对；上述迭代确保在迭代过程中消除被支配的对，同时根据单调性将所有更小的可实现集合隐式保留。

定理 3. 对于所有 $s \in S^{\times}$ 和 $C \subseteq \Phi$,

$$(s, C) \in \mathsf{WinM} \iff C \textit{ is realizable from } s \wedge \neg\exists D \supset C.\ D \textit{ is realizable from } s.$$

迭代 WinM_i 最多经过 $|S^{\times}| \cdot 2^{|\Phi|}$ 步后稳定。

WinM 将每个状态 s与其 $\subseteq$-极大可实现性质集关联起来。对于任意可实现的性质集 $C \subseteq \Phi$，若 $(s_0^{\times}, C) \in \mathsf{WinM}$，则可以通过将 WinM 投影到 C，构造一个获胜策略来满足 C，并将其表示为有限变换器；这与第 4.1 节中的非极大构造完全相同。

例1。 图*1*对比了多性质不动点与最大变体。每个状态 s 都标注了满足条件的集合 C，使得 $(s, C) \in \mathsf{WinM}$。在多性质不动点（图*1a*）中，WinM 包含所有可实现的性质集合。这里， s_1 满足所有目标，因此与每个 $\{\varphi_1, \varphi_2, \varphi_3\}$ 的子集相关联，而 s_2 （分别是 s_3）与 $\{\varphi_1, \varphi_2\}$ （分别是 $\{\varphi_2, \varphi_3\}$）的所有子集相关联。在状态s_0，输出 y_1 会导致 s_1 或 s_2，因此只有两个后继状态共有的子集（即 $\{\varphi_1, \varphi_2\}$ 的所有子集）是可实现的；无论环境输入是什么，输出 y_2 都会导致 s_3 ，从而使 $\{\varphi_2, \varphi_3\}$ 的所有子集都可实现。因此，s_0 恰好与这两个集合的并集相关联。

在最大变体（图 *1b*）中，WinM 仅包含极大可实现集合，将较小的子集隐式保留。因此，这些集合会缩减为例如 $\{\{\varphi_1, \varphi_2\}, \{\varphi_2, \varphi_3\}\}$ 在 s_{0}。

我们的不动点构造避免了针对每个 $C \subseteq \Phi$显式求解一个单独的综合博弈，但在最坏情况下，我们的算法仍可能需要为每个状态跟踪指数数量的目标集合。最大变体通过仅保留最大的目标集合来减少这一数量。然而，在最坏情况下，最大集合的数量仍可能随性质数量呈指数增长，因为集合 Φ 中包含 $|\Phi| = n$的 $\subseteq$极大子集的数量至多为 $\binom{n}{\lfloor n/2 \rfloor} \in \Theta(\frac{2^n}{\sqrt{n}})$ [（Stirling，1730；Sperner， 1928]）。

定理 4. 多属性线性时序逻辑f 可实现性与合成是双指数时间-完全的。

这意味着，我们的方法和基于枚举的基线方法都无法改善该问题的最坏情况复杂度。另一方面，我们的方法在所有目标集合间共享开销高昂的博弈构造，并在单个不动点中传播可实现性信息。在下一节中，我们将通过一个符号化过程进一步放大这一效果。

5 符号综合

现在，我们从博弈场的显式表示转向符号表示，其中状态、转移和接受条件均使用布尔公式编码。这种符号方法以布尔综合为基础 [Fried等人， 2016] 并带来了显著的实际优势，因为可以通过高效的布尔运算执行不动点迭代 [Zhu 等人， 2017]。我们首先回顾符号化 DFA 博弈场、单属性符号综合

并介绍我们的符号化多属性综合过程。**定义 6**（符号化 DFA 博弈竞技场）。设 $\mathcal{G} = (2^{\mathcal{X}\cup\mathcal{Y}}, S, s_0, \delta, F)$ 为一个 *DFA* 博弈场。相应的 符号化 DFA 博弈竞技场是一个元组

$$\mathcal{F} = (\mathcal{X}, \mathcal{Y}, \mathcal{Z}, Z_0, \eta, f),$$

其中： $\mathcal{Z}$ 是一个由 $\lceil \log_2 |S| \rceil$ 布尔状态变量组成的集合，使得每个状态 $s \in S$ 都由一个赋值 $Z \in 2^{\mathcal{Z}}$ 编码，$Z_0 \in 2^{\mathcal{Z}}$ s_0 η $2^{\mathcal{X}} \times 2^{\mathcal{Y}} \times 2^{\mathcal{Z}} \to$ 编码初始 $2^{\mathcal{Z}}$ 状态，： 是一个布尔转移函数：当其编码 s时，赋值 $\eta(X, Y, Z)$ 编码其后继$\delta(s, X \cup Y)$；而 f 是关于 $\mathcal{Z}$ 的布尔公式，满足当且仅当 Z 编码一个接受状态时，$f(Z)$ 成立，即 $s \in F$。

直观地说， $\mathcal{F}$ 通过将$X \in 2^{\mathcal{X}}$、 $Y \in 2^{\mathcal{Y}}$和 $Z \in 2^{\mathcal{Z}}$ 视为 $\mathcal{X}$、$\mathcal{Y}$和 $\mathcal{Z}$中变量的布尔赋值，以符号方式表示该游戏；当某个变量包含在相应赋值中时，该变量取真。因此， η 会根据转移关系更新状态变量，而 f则刻画接受状态 [Zhu 等人, 2017]。

5.1 符号化单性质合成

按照 [Zhu等人 的方法， 2017], 给定一个符号化博弈场$\mathcal{F}$，我们在状态变量上维护两类公式：获胜状态公式 $w_i(Z)$ 和获胜步公式 $t_i(Z, Y)$。当赋值 $Z \in 2^{\mathcal{Z}}$ 满足 $w_i(Z)$ 时，它编码一个获胜状态；当一对 (Z, Y) 满足 $t_i(Z, Y)$ 时，它编码该状态下的获胜输出选择。

我们以接受条件 $t_0(Z, Y) \equiv f(Z)$ 和 $w_0(Z) \equiv f(Z)$ 作为初始值，因为智能体一旦到达接受状态，就可以立即终止。随后，我们通过符号化不动点计算迭代地细化这些公式：

$$\begin{aligned} t_{i+1}(Z, Y) \;&\equiv\; t_i(Z, Y) \vee \\ &\qquad \big(\neg w_i(Z) \wedge \forall X.\, w_i\big(\eta(X, Y, Z)\big)\big), \\ w_{i+1}(Z) \;&\equiv\; \exists Y.\, t_{i+1}(Z, Y). \end{aligned}$$

直观地说， t_{i+1} 恰好添加那些无论环境输入如何，都能迫使下一状态进入当前获胜区域的输出选择。合取项 $\neg w_i$ 确保每一步都严格地朝接受状态推进。当 $w_{i+1} \equiv w_i =: w$ 时，迭代稳定；此时， w 刻画了所有获胜状态。布尔运算可以借助二元决策图（BDDs）等符号化表示高效地执行 [Fried 等人， 2016]。

如果 $w(z_0)$ 为真，就可以以符号方式 提取获胜策略。根据最终移动公式 $t(Z, Y)$，我们可以构造一个函数 $\tau\colon 2^{\mathcal{Z}} \to 2^{\mathcal{Y}}$，它为每个获胜状态编码 Z 选择一个获胜输出 $Y = \tau(Z)$，即满足 $t(Z, Y)$ 为真的输出 [Fried 等人， 2016], 从而得到一个实现该获胜策略的符号化变换器 [Zhu 等人， 2017]。

5.2 符号多属性合成

现在，我们介绍用于多属性线性时序逻辑f 合成的符号化方法。除了符号化方法的一般优势外，我们的方法还利用了一项关键的结构优势：我们构造的布尔公式能够自然地刻画多属性可实现性的单调性。我们对最大可实现集合的布尔表示以紧凑形式同时刻画所有 可实现子集，而无需显式枚举这些子集并将其写入公式中。值得注意的是，我们的符号化过程在算法层面上几乎与单属性版本完全相同。我们增加线性数量的布尔变量来表示这些性质，然后像单属性方法一样应用相同的符号化不动点计算。与第 4 节中需要新增前驱算子来处理性质集合的显式状态过程不同，符号化版本不需要新增运算。最终得到一个关于状态和属性变量的单一公式，以紧凑形式刻画对于每个状态，可从该状态同时实现的所有 性质集合。

定义 7（符号化多属性博弈竞技场）。 对于每个目标 $\varphi_i \in \Phi$，令 $\mathcal{F}_i = (\mathcal{X}, \mathcal{Y}, \mathcal{Z}_i, Z_i^0, \eta_i, f_i)$ 为其符号化确定有限自动机博弈场。符号化多属性博弈场为

$$\mathcal{F}^{\times} = (\mathcal{X}, \mathcal{Y}, \mathcal{Z}, Z_0, \eta),$$

其中 $\mathcal{Z} = \mathcal{Z}_1 \cup \cdots \cup \mathcal{Z}_n$ 是状态变量， $Z_0 = (Z_1^0, \ldots, Z_n^0)$ 是初始状态，而 $\eta = (\eta_1, \ldots, \eta_n)$ 是逐分量转移函数。

为表示性质集合，我们引入 n 布尔变量 $\mathcal{K} = \{k_1, \ldots, k_n\}$，其中赋值 $K \in 2^{\mathcal{K}}$编码一个子集 $C \subseteq \Phi$，且当且仅当 $\varphi_i \in C$ 时， $k_i =$ 为真。随后，我们将初始获胜关系公式定义为

$$w_0(Z, K) := \bigwedge_{i=1}^{n} \big(k_i \Rightarrow f_i(Z_i)\big),$$

当由 $Z = (Z_1, \ldots, Z_n) \in 2^{\mathcal{Z}}$ 表示的状态满足 $K \in 2^{\mathcal{K}}$ 所表示的所有性质时，该公式成立。我们方法的一个关键要素是在此公式中使用蕴含式，它直接刻画了多属性可实现性的单调性：如果 $w_0(Z, K)$ 对某个性质集成立，那么它对该集合的每个子集也成立。这样，我们便能紧凑地表示指数级多个目标组合，而无需显式枚举。

现在，我们采用与单性质方法相同的符号化不动点计算，公式的变量范围同时涵盖状态变量 Z 和目标变量 K：

$$\begin{aligned} t_{i+1}(Z, Y, K) &\equiv t_i(Z, Y, K) \\ &\qquad \vee \big(\neg w_i(Z, K) \wedge \forall X.\, w_i(\eta(X, Y, Z), K)\big), \\ w_{i+1}(Z, K) &\equiv \exists Y.\, t_{i+1}(Z, Y, K), \end{aligned}$$

其中 w_0 的定义如上，而 $t_0(Z, Y, K) = w_0(Z, K)$。至关重要的是，该计算在每次迭代中都保持单调性。这使我们能够像单性质情形一样执行符号化操作，而无需修改算法来处理指数级多个性质组合。计算稳定后，所得公式 $w(Z, K)$ 即刻画了完整的多属性获胜关系 WinM。与单性质情形一样，对于可实现的 C，可以通过从 $t(Z, Y, K)$ 中选择适当的输出赋值，将获胜策略提取为一个变换器。

例2。 图*1c*说明了符号过程如何紧凑地表示WinM。例如，设 z_2 是状态 s_2 的符号编码，并假设它对于 φ_1 和 φ_2 是接受状态，但对于 φ_3不是，即 $f_1(z_2) = f_2(z_2) =$ 为真

族	# 目标	# 状态	MPSynth [s]	枚举 [s]
链	2	16	**0.0001**	0.0106
	4	256	**0.0002**	0.0196
	6	15,625	**0.0700**	0.157
计数器	5	86,400	**0.880**	2.067
	5	181,200	**2.113**	7.492
	5	373,920	**4.713**	60.74
直到	6	10,077,696	**81.10**	299.55
	8	16,777,216	**119.30**	468.29
	10	60,466,176	**562.70**	2640.49
next	7	35,831,808	**109.28**	988.56
	7	62,748,517	**291.60**	2713.31
	7	105,413,504	**611.90**	4925.10
机器人导航	15	12,582,912	**56.50**	170.41
	16	50,331,648	**426.98**	702.01
	17	100,663,296	**1003.21**	2812.53

表 1：基准实例的结果。对于每个族，我们报告目标数量、竞技场规模以及以秒为单位的运行时间。

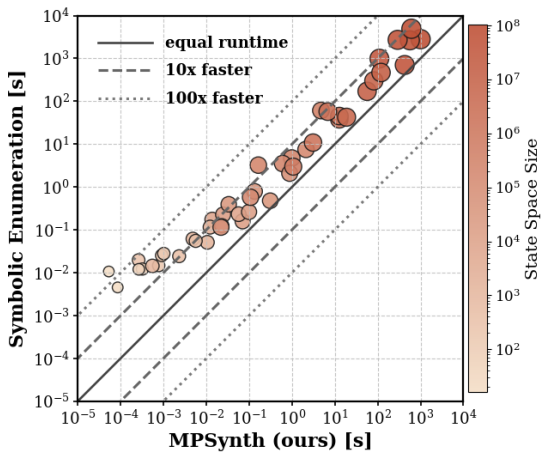


图 2：我们的方法与枚举基线在所有基准实例上的运行时间比较。每个点代表一个实例，颜色表示状态空间大小。

而 $f_3(z_2) = \text{为假}$。因此，在将 Z 固定为 z_2时，初始获胜关系公式$w_0(Z, K)$ 变为如下形式：

$$w_0(z_2, K) \equiv (k_1 \Rightarrow \mathit{true}) \wedge (k_2 \Rightarrow \mathit{true}) \wedge (k_3 \Rightarrow \mathit{false}) \equiv \neg k_3.$$

因此，w_0（z_2，K）恰好由那些 $k_3 = \text{为}$ *false* 的目标编码满足，也就是说，恰好是 $\{\varphi_1, \varphi_2\}$ 的子集。由此得到的状态 s_0 的获胜关系公式为：

$$w(z_0, K) \equiv \neg k_1 \vee \neg k_3,$$

它恰好刻画了可从 s_0 实现的目标集合。这凸显了符号编码的关键优势：关于 K 的一个简洁布尔公式，就能简明地表示涉及指数级数量目标组合的可实现性关系。实践中，底层 *BDD* 表示能够高效处理化简，并在不动点计算过程中对公式进行规范化和约简。

最后，提取一个最优目标集，就相当于选择一个获胜目标赋值 K，使得设为 true 的目标变量 k_i 数量最多。给定 $w($$z_0$，$K)$ 的 BDD，我们可以通过对 K 进行量化并选取一个未被支配的见证，高效地获得这样的赋值：不存在另一个获胜赋值 K' 严格扩展它，也就是说，将 K 的所有目标以及至少一个额外目标都设为 true。

6 实验

我们在名为 *MPSynth*[1] 的工具中实现了符号化多属性综合过程，该工具基于符号化线性时序逻辑（LTL）f 综合工具*LydiaSyft* [ZhuandFavorito, 2025]。我们的方法计算获胜关系，并提取一个最大的可实现集合及其对应的获胜策略；报告的运行时间包含这两个步骤。我们将其与枚举基线进行比较：该基线遍历子集 $C \subseteq \Phi$，并通过单性质符号化 $\bigwedge_{\varphi \in C} \varphi$ 线性时序逻辑（LTL）f 综合来检查可实现性。为避免冗余检查，该基线按大小递增的顺序探索子集，并剪枝不可实现集合的超集 [Agrawal 等人，1993]。

基准测试。 我们在一组多样化的基准测试系列上进行评估，这些系列着重考察多属性合成的不同方面。链、直到和下一步 系列是参数化基准测试，可生成具有复杂时间依赖关系的大型乘积竞技场 [Weinhuber等， 2025]。计数器系列扩展了 [Aminof等提出的计数器博弈， 2024],其中智能体管理一个 k 位计数器，并且必须满足环境指定的多种相互冲突的递增策略。机器人导航系列模拟复杂的机器人导航任务，其中包含多个受环境约束的空间目标 [Aminof等， 2024]。在这些系列中，我们改变性质的数量，以评估可扩展性。完整的基准测试说明、参数配置和实验结果见附录 B。

结果。 表 1 展示了涵盖所有基准测试系列的代表性实例，体现了我们方法显著的性能优势。图 2 可视化了所有实例的运行时间对比。对于所有问题实例，我们的符号化多属性综合始终显著优于枚举基线，加速比最高可达两个数量级。这表明，与针对不同问题特征的最先进枚举方法相比，我们的符号化多属性综合方法有了显著改进。

7 结论

我们提出了一种用于多属性线性时序逻辑f合成的符号方法，通过单次不动点计算得出状态与目标集合之间的完整可实现性关系，从而避免显式进行子集枚举。借助单调性，我们使用布尔变量对目标进行编码，并采用标准的符号不动点计算。我们的实验表明，与枚举基线方法相比，该方法的加速比最高可达两个数量级，

[1] https://github.com/SymbolicSynthesis/MPSynth

使我们的方法能够有效地分析过度订阅领域中的目标权衡。

我们计划从三个关键方向拓展我们的贡献：(i) 在线策略适应，利用预先计算的获胜关系，在执行过程中，如果环境的对抗性较弱，使更大的目标集合变得可实现，则动态调整策略；(ii) 定量偏好，引入加权性质；以及 (iii)概率环境，将方法扩展到马尔可夫决策过程，以概率贝尔曼更新取代可控前驱，从而最大化满足多个目标的可能性。

致谢

本研究部分得到了英国研究与创新署（UKRI）埃尔朗根人工智能数学与计算基础中心（项目编号 EP/Y028872/1）的资助。

参考文献

[Agrawal等, 1993] Rakesh Agrawal、Tomasz Imielinski 和 Arun N. Swami。在大型数据库中挖掘项目集之间的关联规则。载于 *SIGMOD* 会议。ACM 出版社，1993年。

[Aminof等, 2024] Benjamin Aminof、Giuseppe De Gia- como、Gianmarco Parretti 和 Sasha Rubin。在多层环境中实现 LTLf 尽力综合的有效方法。载于*IJCAI*。ijcai.org，2024年。[Baier 和 McIlraith， 2008] Jorge A. Baier 和 Sheila A. McIlraith。偏好规划。《人工智能杂志》，29(4)，2008。[Bienvenu等, 2006] Meghyn Bienvenu、Christian Fritz 和 Sheila A. McIlraith。定性时序偏好规划。载于第十届知识表示与推理原理国际会议论文集（*KR 2006*），2006。[Cabodi 和 Nocco， 2011] Gianpiero Cabodi 和 Sergio Nocco。多个性质的优化模型检查。载于*DATE*。IEEE，2011。[Camacho等， 2018] Alberto Camacho、Jorge A. Baier、Christian J. Muise 和 Sheila A. McIlraith。将有限轨迹 LTL 综合视为规划。载于*ICAPS*。AAAI 出版社，2018。[De Giacomo 和 Rubin， 2018] Giuseppe De Giacomo 和 Sasha Rubin。有限轨迹线性时序逻辑和线性动态逻辑目标的 FOND 规划自动机理论基础。载于*IJCAI*。ijcai.org，2018。[De Giacomo 和 Vardi， 2013] Giuseppe De Giacomo 和 Moshe Y. Vardi。有限轨迹上的线性时序逻辑与线性动态逻辑。载于*IJCAI*。IJCAI/AAAI，2013年。[De Giacomo 和 Vardi， 2015] Giuseppe De Giacomo 和 Moshe Y. Vardi。有限轨迹上的 LTL 与 LDL 综合。载于*IJCAI*。AAAI 出版社，2015年。

[De Giacomo等， 2014] Giuseppe De Giacomo、Riccardo De Masellis、Marco Grasso、Fabrizio Maria Maggi 和 Marco Montali。基于 LTL 和 LDL 的有限轨迹业务元约束监控。载于 *BPM*，《计算机科学讲义》第 8659 卷。施普林格，2014年。

[Dimitrova 等， 2018] Rayna Dimitrova、Mahsa Ghasemi 和 Ufuk Topcu。线性时序逻辑规范的最大可实现性。载于 *ATVA*，《计算机科学讲义》第 11138 卷。施普林格，2018年。[Dimitrova 等， 2020] Rayna Dimitrova、Mahsa Ghasemi 和 Ufuk Topcu。具有线性时序逻辑规范最大可实现性的反应式合成。*Acta Informatica*，57(1-2)，2020年。

[Duret-Lutz 等, 2025] Alexandre Duret-Lutz、Shufang Zhu、Nir Piterman、Giuseppe De Giacomo 和 Moshe Y. Vardi。《有限轨迹线性时序逻辑综合工具的工程实现》。载于 Giuseppa Castiglione 和 Sabrina Mantaci 编，自动机的实现与应用——第*29*届国际会议，*CIAA 2025*，意大利巴勒莫，*2025*年*9*月*22*日至*25*日，论文集，第15981卷，《计算机科学讲义》。施普林格，2025年。

[Etessami 等, 2007] Kousha Etessami、Marta Z. Kwiatkowska、Moshe Y. Vardi 和 Mihalis Yannakakis。马尔可夫决策过程的多目标模型检测。载于*TACAS*，第4424卷，《计算机科学讲义》。施普林格，2007年。

[Fried等, 2016] Dror Fried、Lucas M. Tabajara 和 Moshe Y. Vardi。基于BDD的布尔函数合成。载于*CAV (2)*，第9780卷，《计算机科学讲义》。施普林格，2016年。

[Montali 等, 2010] Marco Montali、Maja Pesic、Wil M. P. van der Aalst、Federico Chesani、Paola Mello 和 Sergio Storari。《服务编排的声明式规范与验证》。*ACM*网络学报，4(1)，2010年。

[Pesic and van der Aalst, 2006] Maja Pesic 和 Wil M. P. van der Aalst。《灵活业务流程管理的声明式方法》。载于业务流程管理研讨会，第4103卷，收录于《计算机科学讲义》。施普林格，2006。

[Pnueli and Rosner, 1989] Amir Pnueli 和 Roni Rosner。《异步反应模块的合成》。载于*ICALP*，第372卷，收录于《计算机科学讲义》。施普林格，1989年。

[Pnueli, 1977] Amir Pnueli。《程序的时序逻辑》。载于*FOCS*。IEEE计算机学会，1977。

[Smith, 2004] David E. Smith. 过度订阅规划中的目标选择。载于 *ICAPS*。AAAI，2004年。

[Sohrabi等, 2009] Shirin Sohrabi、Jorge A. Baier 和 Sheila A. McIlraith。带偏好的 HTN规划。载于*IJCAI*，2009。

[Sperner, 1928] Emanuel Sperner. 关于有限集合子集的一个定理。《数学杂志》，27(1)，1928。

[Stirling, 1730] JamesStirling. 微分法：无穷级数求和与插值论. G. Strahan, 1730.[塔斯基， 1955] 阿尔弗雷德·塔斯基. 格论不动点定理及其应用。《太平洋数学杂志》，5(2)，1955。

[van den Briel 等， 2004] Menkes van den Briel、Romeo Sanchez Nigenda、Minh Binh Do 和 Sub- barao Kambhampati。部分满足（超额订阅）规划的有效方法。载于*AAAI*。AAAI 出版社 / 麻省理工学院出版社，2004年。

[Weinhuber 等人， 2025] Christoph Weinhuber、Giuseppe De Giacomo、Yong Li、Sven Schewe 和 Qiyi Tang。《用于随机 LTL规划的 MDP 状态精简》。*CoRR*，abs/2511.09073，2025年。

[Zhu 和 Favorito， 2025] Shufang Zhu 和 Marco Favorito。《LydiaSyft：一个用于 LTLf 规范的组合式符号综合框架》。载于 *TACAS(1)*，第 15696 卷，施普林格计算机科学讲义。施普林格，2025年。

[Zhu 等人， 2017] Shufang Zhu、Lucas M. Tabajara、Jianwen Li、Geguang Pu 和 Moshe Y. Vardi。《符号化 LTLf 综合》。载于*IJCAI*。ijcai.org，2017年。

A 证明

在本附录中，我们给出所有已陈述定理的形式化证明，以确立多属性综合过程的正确性、相应获胜策略的构造以及时间复杂度结果。

A.1 多属性合成的正确性

我们给出定理 1 的证明，该定理表明关系 WinM 刻画了多属性可实现性。首先介绍一些有用的定义和引理。

引理1（PreMC的单调性。可控多属性前驱是单调的：对于任意两个关系$\mathcal{E}$、$\mathcal{E}' \subseteq S^{\times} \times 2^{\Phi}$，若 $\mathcal{E} \subseteq \mathcal{E}'$，则 PreMC（$\mathcal{E}$） $\subseteq$ PreMC（$\mathcal{E}'$）。

证明。设 $(s,\ C) \in \mathsf{PreMC}(\mathcal{E})$。根据定义，存在一个输出 $Y \in 2^{\mathcal{Y}}$，使得对于所有环境响应$X \in 2^{\mathcal{X}}$，后继状态满足 $(\delta^{\times}(s,\ X \cup Y)\ C) \in \mathcal{E}$。由于 $\mathcal{E} \subseteq \mathcal{E}'$，可得对于所有$X$，$(\delta^{\times}(s,\ X \cup Y)\ C) \in \mathcal{E}'$，这意味着 $(s,\ C) \in \mathsf{PreMC}(\mathcal{E}')$。□

定义 8 (i步可实现性)。 目标集 $C \subseteq \Phi$从状态 $s \in S^{\times}$ 出发是 i步多目标可实现的，当且仅当存在策略 g，使得对于每个环境输入序列 $\xi \in (2^{\mathcal{X}})^{\omega}$，从 s 开始的诱导对局 $s = s_0 s_1 \ldots$会在某个 $0 \le j \le i$到达状态 $s_j \models C$。

引理2 (WinM_i的刻画)。 对于任意 $i \in \mathbb{N}$、状态 $s \in S^{\times}$和目标集 $C \subseteq \Phi$，$(s, C) \in \mathsf{WinM}_i$ 当且仅当 C 从 s出发是 i步多目标可实现的。

证明。 我们对 i进行归纳法。

基础情形（$i = 0$）：根据定义，$(s, C) \in \mathsf{WinM}_0$ 当且仅当 $s \models C$。这等价于说，智能体可以在 0 步内通过立即在 s终止来满足 C。

归纳步骤： ($\Rightarrow$)：设 $(s,\ C) \in \mathsf{WinM}_{i+1} = \mathsf{WinM}_i \cup \mathsf{PreMC}(\mathsf{WinM}_i)$。若 $(s,\ C) \in \mathsf{WinM}_i$，则由归纳假设（IH）可得结论。否则，$(s,\ C) \in \mathsf{PreMC}(\mathsf{WinM}_i)$。因此，存在 $Y \in 2^{\mathcal{Y}}$，使得对于所有 $X \in 2^{\mathcal{X}}$，后继状态 $s' = \delta^{\times}(s,\ X \cup Y)$ 满足 $(s',\ C) \in \mathsf{WinM}_i$。根据归纳假设，$C$ 可从每个这样的 s' 出发，在 i 步内实现。在状态 s 采取对局 Y，智能体可确保在最多 $1 + i$ 步内实现 C。

($\Leftarrow$)：假设 C 从 s出发是 $(i + 1)$ 步可实现的。若 C 已经是 i步可实现的，则根据归纳假设，$(s, C) \in \mathsf{WinM}_i \subseteq \mathsf{WinM}_{i+1}$。否则，必定存在一个输出 Y，使得对于所有X，C 从 $s' = \delta^{\times}(s, X \cup Y)$出发是 i步可实现的。根据归纳假设，对于所有 X，$(s', C) \in \mathsf{WinM}_i$。根据前驱的定义，$(s, C) \in \mathsf{PreMC}(\mathsf{WinM}_i)$，因此$(s, C) \in \mathsf{WinM}_{i+1}$。

□

定理5（定理 1 的重述）。 不动点$\mathsf{WinM} = \bigcup_{i \ge 0} \mathsf{WinM}_i$ 刻画了多属性可实现性：对于所有 $s \in S^{\times}$ 和 $C \subseteq \Phi$，

$$(s, C) \in \mathsf{WinM} \iff C \textit{ is realizable from } s.$$

此外，序列 $(\mathsf{WinM}_i)_{i \in \mathbb{N}}$ 至多经过$|S^{\times}| \cdot 2^{|\Phi|}$ 步便会稳定。

证明。 证明分为两部分：稳定性和语义正确性。

稳定性： 多属性获胜关系被计算为算子 $F(\mathcal{E}) = \mathsf{WinM}_0 \cup \mathsf{PreMC}(\mathcal{E})$ 的最小不动点。由于幂集格 $(2^{S^{\times} \times 2^{\Phi}},\ \subseteq)$ 是完备的，且 PreMC 是单调的（引理1），根据*Knaster-Tarski* 定理 [Tarski， 1955]，最小不动点的存在性得以保证。由于定义域 $S^{\times} \times 2^{\Phi}$是有限的，序列 WinM_i 至多经过 $|S^{\times}| \cdot 2^{|\Phi|}$步便会到达该不动点。

正确性： 我们证明，$(s,\ C) \in \mathsf{WinM}$ 当且仅当 C 可由 s实现。

- **可靠性**（$\Rightarrow$）：假设 $(s,\ C) \in \mathsf{WinM} = \bigcup_{i \ge 0} \mathsf{WinM}_i$。那么，对于某个有限$i$，$(s,\ C) \in \mathsf{WinM}_i$。根据引理2，$C$ 是从 s出发的 i步可实现的，这意味着 C 是可实现的。

- **完备性**（$\Leftarrow$）：如果 C 可由 s实现，那么根据线性时序逻辑$_f$ 合成的定义（定义1），必然存在一个有限时域 i，使得智能体能够在至多 i步内s获胜。根据引理 2，这意味着 $(s,\ C) \in \mathsf{WinM}_i$。由于 $\mathsf{WinM}_i \subseteq \mathsf{WinM}$，我们有 $(s,\ C) \in \mathsf{WinM}$。

□

A.2 策略提取的正确性

现在，我们证明将获胜策略提取为有限状态转换器的正确性。

定理 6（定理 2 的重述）。 对于任何满足 $(s_0^{\times},\ C) \in \mathsf{Win}^M$ 的 $C \subseteq \Phi$，存在一个通过策略提取构造得到的变换器 T_C，其实现了一个用于满足 C 的获胜策略：对于每个输入序列 $\xi \in (2^{\mathcal{X}})^{\omega}$，由 T_C 诱导的对局经过有限步后会到达一个状态 $s \in S^{\times}$，满足 $s \models C$，且生成的有限轨迹满足每个 $\varphi_i \in C$。

证明。固定一个目标集 $C \subseteq \Phi$。限制在 C 上的多属性获胜关系定义为 $\mathsf{WinM}|_C = \{s \in S^{\times} \mid (s,\ C) \in \mathsf{WinM}\}$。展开多属性不动点迭代的定义可知，$\mathsf{WinM}|_C$ 满足：

1. $\mathrm{WinM}_0|_C = \{s \in S^\times \mid s \models C\}$,

2. $\mathrm{WinM}_{i+1}|_C = \mathrm{WinM}_i|_C \cup \{s \in S^\times \mid \exists Y \in 2^{\mathcal{Y}}.\forall X \in 2^{\mathcal{X}}.\delta^\times(s, X \cup Y) \in \mathrm{WinM}_i|_C\}$。

这恰好还原了在具有终态 $F_C = \{s \in S^\times \mid s \models C\}$的确定有限自动机博弈场 $\mathcal{G}^\times$ 上进行单属性可实现性的传统不动点刻画。

如 [De Giacomo 和 Vardi 2015], 所示，这类博弈解决了线性时序逻辑f 规范$\bigwedge_{\varphi_i \in C} \varphi_i$的综合问题。我们构造的变换器 T_C 与 [De Giacomo 和 Vardi（2015 年，定理 4]）确立的单属性情形一致。因此，对局会在有限步内到达 F_C，从而确保生成的有限轨迹同时满足 C 中的所有性质。 □

A.3 最大多属性综合的正确性

本节将证明，最大不动点迭代能够正确计算最大可实现目标集。下文中，我们用 WinM 表示第 4.1 节中的多属性可实现性关系，用 WinMM 表示第 4.2 节中的最大变体。

对于任意关系 $\mathcal{E} \subseteq S^\times \times 2^\Phi$，我们将其向下闭包定义为 $\downarrow(\mathcal{E}) := \{(s,\ D) \mid \exists C\colon (s,\ C) \in \mathcal{E}$ 且 $D \subseteq C\}$。该算子可重构由最大集合所蕴含的所有可实现子集。

引理 3 (向下闭包保持性)。 对于任意关系 $\mathcal{E} \subseteq S^\times \times 2^\Phi$，Max 算子保持向下闭包，即 $\downarrow(\mathsf{Max}(\mathcal{E})) = \downarrow(\mathcal{E})$。此外，最大前驱和标准前驱满足$\downarrow(\mathsf{PreMMC}(\mathcal{E})) = \mathsf{PreMC}(\downarrow(\mathcal{E}))$。

证明。 第一个恒等式可由 Max 的定义直接得出：由于 $\mathsf{Max}(\mathcal{E})$ 只会删除那些包含于某个现有 $(s,\ D) \in \mathcal{E}$ 的对 (s, C)，这些被删除的子集仍然保留在 $\downarrow(\mathsf{Max}(\mathcal{E}))$ 中。

对于第二个等式，固定一对 $(s,\ C)$。根据定义，$(s,\ C) \in \mathsf{PreMC}(\downarrow(\mathcal{E}))$ 当且仅当存在 $Y \in 2^{\mathcal{Y}}$，使得对于所有 $X \in 2^{\mathcal{X}}$，后继状态 s'满足 $(s',\ C) \in \downarrow(\mathcal{E})$。这等价于存在 $\exists D \supseteq C$ 使得 $(s',\ D) \in \mathcal{E}$。这正是 $(s,\ C) \in \mathsf{PreMMC}(\mathcal{E})$ 的定义谓词；当其向下封闭时，即可得到所需结果。 □

定理 7 （定理 3 的重述）。 最大不动点 $\mathsf{WinMM} = \bigcup_{i \geq 0} \mathsf{WinMM}_i$ 刻画了最大可实现的目标集合：

$$(s, C) \in \mathsf{WinMM} \iff C \textit{ is realizable from } s \wedge \neg\exists D \supset C.\ D \textit{ is realizable from } s.$$

该序列至多经过 $|S^\times| \cdot 2^{|\Phi|}$ 次迭代后稳定。

证明。 令 $(\mathsf{WinMM}_i)_{i\in\mathbb{N}}$ 为最大迭代，并考虑向下封闭关系序列 $E_i = \downarrow(\mathsf{WinMM}_i)$。利用引理 3 中的恒等式，可得：

$$\begin{aligned} E_{i+1} &= \downarrow\big(\mathsf{Max}(\mathsf{WinMM}_i \cup \mathsf{PreMMC}(\mathsf{WinMM}_i))\big) \\ &= \downarrow(\mathsf{WinMM}_i) \cup \downarrow(\mathsf{PreMMC}(\mathsf{WinMM}_i)) \\ &= E_i \cup \mathsf{PreMC}(E_i). \end{aligned}$$

由于 $E_0 = \downarrow(\mathsf{Max}(\{(s,\ C) \mid s \models C\})) = \mathsf{WinM}_0$, , 该序列 $(E_i)_{i\in\mathbb{N}}$ 与标准的多属性迭代 $(\mathsf{WinM}_i)_{i\in\mathbb{N}}$相同。根据定理 1，该序列的极限就是完整的可实现性关系 WinM。因此 $\downarrow(\mathsf{WinMM}) = \mathsf{WinM}$。这意味着：

1. **可靠性：** 每个 $(s,\ C) \in \mathsf{WinMM}$ 都包含于 WinM 中，因此，C 可从 s 实现。2. **极大性：** 通过在每次迭代中应用 Max，如果某个严格更大的$D \supset C$ 已经被证明可实现，则不会保留任何对 $(s,\ C)$。因此，WinMM 只包含 $\subseteq$-极大元素。

反过来，如果 C 是在 s处的极大可实现集合，那么 $(s,\ C) \in \mathsf{Max}(\mathsf{WinM})$。由于 $\mathsf{WinM} = \downarrow(\mathsf{WinMM})$，必定存在 $(s,\ D) \in \mathsf{WinMM}$，且满足 $C \subseteq D$。由于 D 可实现且 C 是极大的，因此必有 $C = D$，从而证明 $(s,\ C) \in \mathsf{WinMM}$。由于定义域 $S^\times \times 2^\Phi$有限，稳定性随之成立。 □

A.4 多属性综合的复杂度

本节将确立多属性线性时序逻辑f 可实现性和合成的计算复杂度。

定理 8 （定理 4 的重述）。 多属性线性时序逻辑f 可实现性和合成是双指数时间-完全的。

证明。 通过证明该问题属于 2EXPTIME 复杂度类且对其困难，即可得出结论。对于一组性质 $\Phi = \{\varphi_1, \ldots, \varphi_n\}$，我们用 $m := \sum_{i=1}^n |\varphi_i|$ 和 $|\mathcal{G}_i|$ 表示自动机 $\mathcal{G}_i$的大小。我们用 poly(m) 表示关于 m的某个未明确指定的多项式函数。

成员资格： 多属性综合过程由三个主要计算阶段组成：

1. **自动机构造：** 我们为每个性质 φ_i构造一个 DFA 博弈场 $\mathcal{G}_i$。根据线性时序逻辑 f 公式构造确定有限自动机，其复杂度至多为公式大小的双指数级 [De Giacomo 和 Vardi, 2015]。因此，$|\mathcal{G}_i| \leq 2^{2^{\mathrm{poly}(m)}}$。

2. **博弈场构造：** 多性质博弈场是各个博弈场的笛卡尔积 $\mathcal{G}_i$。其大小满足$|S^\times| \leq (2^{2^{\mathrm{poly}(m)}})^n = 2^{n \cdot 2^{\mathrm{poly}(m)}} \leq 2^{2^{\mathrm{poly}(m)}}$，因此该博弈场的规模至多为输入大小的双指数级。

3.**不动点计算：** 在该博弈场上求解可达性博弈，无论采用标准的基于对的迭代（WinM）还是最大变体（ WinMM），所需时间都与乘积状态的数量 $|S^\times|$呈线性关系，并随性质数量 n呈指数增长。由于 $2^{|\Phi|} \leq 2^m$ 仅为单指数级，而 $|S^\times| \leq 2^{2^{\mathrm{poly}(m)}}$ 已达到双指数级，因此整体而言，不动点阶段的复杂度受限于 $2^{2^{\mathrm{poly}(m)}}$。

由于总体复杂度由确定有限自动机构造步骤主导，该问题属于双指数时间复杂度。

困难性： 这一困难性可直接由单性质线性时序逻辑f 综合问题已确立的双指数时间完全性 [De Giacomo 和 Vardi, 2015]得出。对于性质 φ，任何单性质综合实例都可以通过设置 $\Phi = \{\varphi\}$，平凡地归约为多性质实例。□

B 扩展实验

我们补充说明所用基准测试的细节，并提供完整的实验结果。所有代码、基准实例和工具参数均收录于补充材料中。所有实验均在配备 8 个核心和 32 GiB 内存的 AMD EPYC 中央处理器上进行。

B.1 基准测试系列

我们在五个基准测试系列上进行评估，这些系列分别对多属性线性时序逻辑f 合成的不同方面构成压力。

计数器。 计数器 族基于文献中提出的计数器博弈基准。[2] 直观地说，控制器维护一个 k 位计数器，并且必须响应环境控制的输入；这些输入会随时间请求（可能相互冲突的）计数器更新，从而产生无法始终联合强制实现的目标集合。这会形成一个庞大的符号博弈场，即使对于中等规模的 k 也是如此，并且目标之间还存在非平凡的权衡结构。

机器人导航。 机器人导航 族对带有多个区域到达目标、且受对抗性约束的导航进行建模 [Aminof 等， 2024]。环境可以禁用世界的部分区域，例如关闭门；而到达某些区域可能不可逆地限制对其他区域的访问。因此，满足一个导航目标可能会排除另一个目标，使该系列非常适合评估最大可实现目标集及实现这些目标的策略。

链、直到、下一步。链、直到和下一步系列是参数化的时序模式基准测试，能够生成大型乘积竞技场，同时保持各个公式在语法上简单。它们旨在对特定时序结构进行压力测试，例如长依赖链、嵌套和展开的 U 结构，以及深层 X 推进；这些基准改编自近期用于大规模评估时序推理与合成的基准集合 [Weinhuber 等， 2025]。在这些系列中，我们扩展性质的数量 $|\Phi|$和模式参数，即公式长度与嵌套深度，从而获得规模从小到极大的实例。

B.2 完整结果

表 2 扩展了表 1，报告了所有基准实例的结果。对于每个实例，我们列出工具*MPSynth*的端到端运行时间，其中包括：(i) 通过第 5 节中的不动点计算符号化多属性获胜关系 $w(Z,\ K)$；以及 (ii) 提取最大基数的可实现集合 $C \subseteq \Phi$，并给出相应的获胜策略。表 2 还报告了步骤 (ii) 所用的运行时间。最后，我们还列出枚举基线的运行时间，该基线会枚举各个子集，并利用先前计算的结果剪除已知不可实现的子集；同时也报告了为最大可实现子集提取获胜策略所用的运行时间。

[2]请参阅 [Aminof 等人的基准测试说明， 2024]。

族	实例 (深度 / 长度)	# 目标	# 状态	MPSynth 不动点 [s]	MPSynth 最大可满足性 & 策略提取 [s]	枚举 & 策略提取 [s]
链	3	2	16	4.93e-05	3.42e-06	0.0106
	3	4	256	2.46e-04	1.55e-05	0.0196
	4	6	15,625	0.0697	3.08e-04	0.157
直到	3	3	125	3.10e-04	1.06e-05	0.0121
	4	5	7,776	0.0136	8.04e-05	0.173
	4	6	46,656	0.133	4.02e-04	0.786
	4	7	279,936	0.984	0.0012	4.722
	4	8	1,679,616	12.66	0.0024	37.73
	4	9	10,077,696	80.34	0.762	299.55
	4	10	60,466,176	557.99	4.705	2640.49
	5	5	16,807	0.0245	6.90e-05	0.226
	6	5	32,768	0.0331	6.64e-05	0.393
	6	6	262,144	0.603	3.91e-04	3.505
	6	7	2,097,152	12.74	9.64e-04	45.06
	6	8	16,777,216	119.30	0.0027	468.29
计数器	2	3	168	7.60e-04	9.30e-06	0.0146
	3	3	396	0.0023	1.00e-05	0.0241
	4	3	912	0.0048	1.14e-05	0.0611
	5	3	2,016	0.0122	1.13e-05	0.115
	3	4	1,584	0.0105	2.36e-05	0.0508
	4	4	3,648	0.0229	2.42e-05	0.124
	5	4	8,064	0.0565	2.47e-05	0.231
	2	5	18,240	0.0990	7.53e-05	0.261
	3	5	40,320	0.309	9.14e-05	0.476
	4	5	86,400	0.880	1.18e-04	2.067
	5	5	181,200	2.113	1.49e-04	7.492
	6	5	373,920	4.713	1.14e-04	60.74
next	8	3	1,000	5.36e-04	1.17e-05	0.0143
	10	5	248,832	0.163	1.18e-04	3.225
	10	6	2,985,984	6.825	6.03e-04	57.37
	10	7	35,831,808	109.28	0.0023	988.56
	11	7	62,748,517	291.60	0.0029	2713.31
	12	7	105,413,504	611.90	0.0329	4925.10
机器人导航	2	10	98,304	0.0168	0.0051	0.114
	3	11	196,608	0.0908	0.0164	0.573
	4	12	786,432	0.975	0.0906	2.975
	5	13	1,572,864	2.946	0.145	10.83
	6	14	6,291,456	18.28	0.469	42.38
	7	15	12,582,912	55.41	1.085	170.41
	8	16	50,331,648	422.12	4.856	702.01
	9	17	100,663,296	993.25	9.962	2812.53
env	3	3	18	7.73e-05	7.49e-06	0.0045
优先级	4	4	54	2.45e-04	1.90e-05	0.0119
资源	3	5	162	8.57e-04	4.50e-05	0.0251
监控	3	6	486	0.0055	1.27e-04	0.0549
复杂	3	5	162	9.66e-04	4.62e-05	0.0278

表 2：所有基准实例的完整结果。对于每个实例，我们报告目标数量、乘积博弈场规模以及以秒为单位的运行时间。对于 MPSynth，我们将符号不动点计算时间与计算最大可实现性质集并提取相应获胜策略的时间分别列出。