《人工智能评论》稿件编号（由编辑填写）

用于审慎型机器人任务规划的逻辑编程

Daniele Meli · Hirenkumar Nakawala* **·**
Paolo Fiorini

本文的正式发表版本最初刊载于《人工智能评论》，现可通过出版商网站在线查阅：
http://dx.doi.org/10.1007/s10462-022-10389-w

arXiv:2301.07550v1 [cs.RO] 18 Jan 2023

摘要 过去十年间，机器人在生产和日常生活中的应用日益增多。随着任务日趋复杂，且机器人需要在包括人类在内的不同环境中进行交互，为实现高效审议，机器人必须具备更高水平的自主性。任务规划是审议的关键要素。给定有关机器人和环境的规范，任务规划将基本操作组合成结构化计划，以实现指定目标。本文综述了逻辑编程在任务规划问题中的最新应用进展。与其他方法相比，逻辑编程具有多项优势，包括更强的表达能力和可解释性，这有助于开发安全可靠的机器人。我们根据领域表示的表达能力、计算效率和软件实现，分析了不同规划器及其对特定机器人应用的适用性。通过这种方式，我们帮助机器人设计者为其应用选择最佳工具。

关键词 逻辑编程 · 任务规划 · 审慎型机器人

1 引言

机器人正越来越多地应用于各个行业，包括用于提高生产环境效率并与人类工作者协作的智能工业（Bogue 2011年，2018年）、面向老年人的服务和家政辅助（Mandel et al. 2005; Harmo et al. 2005），以及近年来的自动驾驶（Levinson et al. 2011; Campbell et al. 2010）。在此类复杂场景中，机器人面临的最具挑战性的要求之一是自主性，即无需外部监督即可完成任务 的能力。在各种运行场景中，自主机器人会受到系统外部状态的影响，

*研究工作是在意大利维罗纳大学任职期间完成的。Hiren Kumar 目前任职于英国的 CMR Surgical Ltd。

D. Meli、H. Nakawala、P. Fiorini 维罗纳大学计算机科学系 意大利维罗纳 Strada Le Grazie 15 电子邮件： {daniele.meli; hirenkumarchandrakant.nakawala; paolo.fiorini}@univr.it

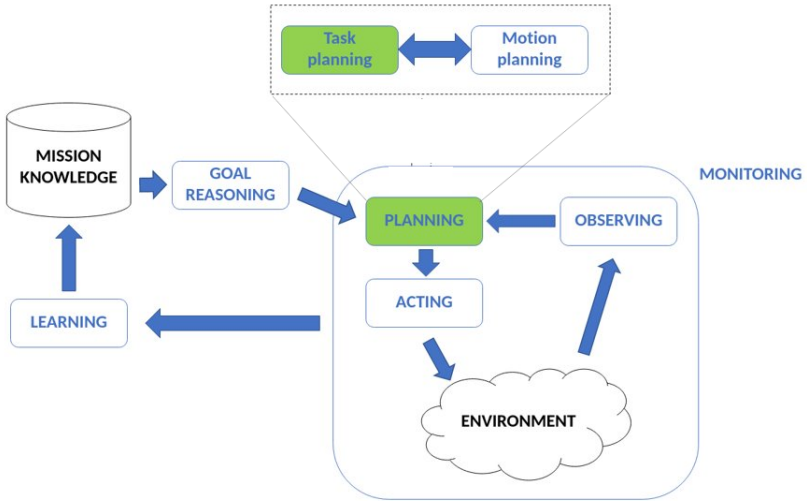

图 1：审议型机器人的功能。本文重点关注审议型机器人任务规划功能的基于逻辑的实现。

即机器人无法控制的状态，例如环境的不可预测性。机器人运行的环境通常充满不确定性且不断变化，工业等受控环境除外。此外，任务可能由一个或多个子使命（称为任务）组成，这些子使命可根据不断变化的情况在运行时定义。例如，考虑一台在家庭环境中运行的机器人。它的任务可能是在不同房间中搜索物体，并将它们全部移动到目标房间。这需要完成移动到每个房间、寻找一个物体、将该物体带到目标房间等任务。然而，如果某个物体很重，任务可能还需要请求人工帮助 将其搬运到目的地。此外，机器人还可以利用有关待探索环境的先验知识，例如物体的常见位置或房间之间的连接关系。

因此，自主性需要审议，其定义为根据对可用资源进行推理而做出决策的能力，即机器人具备的能力、对环境的实际描述以及给定的任务 (Ingrand 和 Ghallab 2017)。

1.1 审议型机器人系统的要求

过去 20 年间，研究者一直在探讨审议型机器人系统的需求（Ghallab 等人 2016；Ingrand 和 Ghallab 2017；Rajan 等，2013；Matari´c 2006）。尽管人工智能（AI）领域对审议提出了不同的解释和刻画方式（Bratman 1987；Bratmanet al. 1988；Bolisani 和 Bratianu 2017），本文采用 Ingrand 和 Ghallab（2017）提出的功能模型。该模型确定了审议所涉及的主要模块，即规划、行动、监控、观察、目标推理和学习（图 1）。规划 被定义为制定策略以完成任务的能力，也就是将多种策略组合起来以完成

各项任务。从现在起，我们将把针对单项任务的策略称为计划，而将策略 这一通用术语保留用于整个任务。规划模块会综合考虑机器人的资源、任务以及由观察 模块提供的传感器信息，以生成计划。由于规划模块结合了机器人和环境信息，并有助于确定机器人在环境中的行为，因此它是审慎型机器人的关键需求。根据各项任务的复杂程度，计划可以是由行动 模块直接执行的运动轨迹（运动规划），也可以是需要转换为运动轨迹的更高层次基本操作序列（任务规划）。任务在任务中的执行顺序通过目标推理来推理。此外，在涉及与环境和／或人类或其他机器人交互的复杂场景中，需要监控 模块来监督规划—执行—观察循环，确保执行过程整体正确，并在必要时触发重新规划。最后，当现有信息不足以制定完成任务的策略时，就需要具备学习 能力。这在机器人学的探索任务中尤为有用，例如环境未知或仅部分已知，必须在执行过程中获取相关信息。本文重点关注机器人审议过程中的任务规划方面。

1.2 面向审议的任务规划

现在，我们将更详细地定义本文所讨论的任务规划问题。受 Ghallab 等人（2016年）提出的定义启发，我们将机器人系统 视为一组具有共同任务的机器人，该任务可通过一系列动作来完成；动作即机器人能够在环境中执行的基本操作。动作取决于任务资源。资源是场景中的相关实体，具体包括机器人智能体和环境中的元素（例如可用于任务的工具或障碍物）。环境通过与特定特征相关的状态变量来定义（例如物体位置以及它们之间的时空关系）。任务规划问题由目标、资源和用户规定的规范共同定义。规范包括动作的前置条件，即执行动作前必须满足的一组条件；动作的后置条件或效果，即动作执行后成立的条件；以及执行约束，即不能同时成立的一组条件。规范中涉及的条件可以针对动作（例如，一个动作必须在另一个动作之后发生），也可以针对环境状态变量（例如，只有当环境处于特定状态时才能执行某个动作）。任务规划器的输出是一个计划，即为实现任务目标并满足任务规范而要执行的有序动作序列。

1.3 论文的范围与结构

运动规划技术自机器人领域创立之初便一直备受关注，因此针对各种应用的相关综述已十分详尽（Raja 和 Pugazhenthi，2012年；Kamil 等人，2015年；Gonz´alez 等人，2015年）；然而，针对机器人任务规划的详细实用综述仍付之阙如。这主要是因为任务规划所涉及的问题范围更广，关系到智能体与环境交互时，其长期行为中的动作协调。这一问题一直受到

人工智能界的关注（Ghallab 等人，2004年、2016年），而机器人领域直到相对近期才开始聚焦于此，当时机器人应用在更复杂的人类场景中迅速增多。根据具体应用的不同，任务规划问题可以通过多种形式化方法和框架来表示和求解。多位作者概述了可能采用的方法，例如，Ghallab 等人（2004年、2016年）主要从人工智能规划的角度进行了介绍，并仅对机器人应用中的通用任务—运动规划作了些许探讨；Jim´enez 等人（2012年）介绍了基于学习的规划；Karpas 和 Magazzeni（2020年）以及 Nakawala 等人（2018年）则探讨了适用于机器人学的通用动作表示形式化方法。本文旨在全面综述用于表示和求解审议型机器人任务规划问题的最先进解决方案，重点关注逻辑编程工具。我们希望不止于介绍高层动作模型和形式化方法，还将分析这些方法如何实现，评估关键软件实现的优缺点，并根据不同规划器的具体应用领域加以分类。近年来，逻辑编程（Lloyd，2012）的普及度在机器人领域显著提升，因为机器人越来越多地参与人类环境中的活动（近期关于人机协作的综述即可佐证，例如 Goodrich 和 Schultz（2008年）；Chandrasekaran 和 Conrad（2015年），以及特别关注安全人机交互的综述，例如 Zacharaki 等人（2020年）；Lasota 等人（2017年））。事实上，与基于学习的方法（例如 Goebel 等人，2018年）不同，逻辑形式体系能够以这样的方式定义任务规划问题：自主行为可由理解其工作原理的人类监督者理解和审查。也就是说，机器人系统的输入（通常包括环境和机器人自身的情境，例如其配置）与输出（计划）都便于操作员监督。这确保了机器人与其他人共同工作时的可靠性，例如在有其他工人的工业环境中（Zhang 等人，2017年），或在家庭环境中协助老年人或残障人士时（Alami 等人，2005年）。此外，可解释规划是最新欧洲高风险人工智能系统法规提案的一项要求[1]，，而人工智能和机器人研究领域对可解释规划的关注正迅速升温（有关这一主题的深入探讨，请参阅 Fox 等人（2017年）的研究）。仅聚焦于用于任务规划的逻辑编程解决方案，才能为希望使用这一工具解决任务规划问题的机器人学研究者提供足够详尽的信息；若要像 Ghallab 等人（2016年）那样涵盖所有相关方法，则难以做到这一点。

第2节基于近期的综述，介绍任务规划器可能的分类体系。随后，我们介绍一类专用于机器人学、基于逻辑编程的任务规划器；这类规划器能够以很强的表达能力描述规划问题。第3节接着说明综述过程的开展方式，并据此阐明本文接下来的组织结构。第4节展示如何使用动作语言描述任务规划问题；动作语言构成了实现基于逻辑编程的规划器的形式化基础。随后，第5节概述本综述的主要成果，详细内容见第6至第8节。最后，第9节对本文作出总结。

2 任务规划器分类

任务规划问题可以用大量形式化方法来描述，这些方法在人工智能领域已有广泛阐述，例如 Ghallab 等人（2016年）以及 Geffner 和 Bonet（2013年）的研究。不同的任务规划器分类法已被提出。

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX: 52021PC0206

Ghallab 等人（2016年）区分了领域依赖型和领域无关任务规划。领域无关规划（Wilkins，1984年）对机器人领域很有意义，因为它有望从非常高层次、通用的领域描述出发，在不同环境或运行场景中解决任务规划问题。例如，在涉及物体运输的复杂机器人任务中，领域无关规划器有助于制定通用计划，以实现运输所有物体的目标。然而，单个物体的操作取决于其特定属性，因此由领域依赖型规划器进行管理会更高效（Dornhege 等人，2009年）。为此，Ghallab 等人（2016年）提出可配置任务规划，将高层次的领域无关任务规划与领域依赖型子任务规划结合起来。Shivashankar 等人（2013年）提供了这种方法的说明性示例，其中将完整任务拆分为一系列更简单的任务。全局策略通过领域无关的分层任务规划来制定，而专门的领域依赖型规划器则负责各个具体任务。该方法在基准规划问题上成功得到验证，其中包括著名的积木世界示例。总体而言，领域依赖型规划有望在更多机器人应用中发挥作用，因为它弥合了问题形式化与实际实现之间的差距。

然而，这种对任务规划器的分类过于粗略，因为它在很大程度上未考虑具体应用场景的特征。Ingrand 和 Ghallab（2017年）提出了另一种分类方式，将确定性和非确定性任务规划器区分开来。非确定性（或概率）任务规划器通常基于马尔可夫决策过程和贝叶斯模型。它们能够捕捉环境的随机性以及传感器的不确定性，并输出最可能的计划（Kolobov，2012年；Teichteil-K¨onigsbuch 等，2010年；Likhachev 等，2005年）。概率规划器更便于与审议过程中的行动和观察模块集成。然而，它们由数据驱动，因此通常不足以支持复杂任务约束的定义，例如物体与机器人之间的时间关系和语义关系。

相反，确定性规划器依赖于对场景更形式化的描述，包括资源和规范。任何实现都始于一种高级动作语言（Erdem 和 Patoglu，2012年），它以不同的表达能力表示领域中的实体及其相互关系。机器人学领域一个最先进的例子是 Aeronautiques 等人（1998年）提出的规划领域描述语言（PDDL）及其扩展（Fox 和 Long，2003年；Younes 和 Littman，2004年）。

随后可以将规划表示转换为可实施的形式，例如 Armando 等（1999）、Vidal 和 Ghallab（1996）、Vidal 和 Fargier（1997）、Frank 和 J´onsson（2003）以及 Fratini 等（2011）针对时间区间时间约束网络提出的方法。基于逻辑形式体系的规划器则能保证更高的表达能力，它们用逻辑陈述来表示有关声明式任务的专家知识。逻辑形式体系的一项关键优势在于，人工监督者（可能不具备技术背景）评估生成的计划时，计划具有可解释性。这确保了机器人在人类身边运行时的可靠性。一些基于逻辑的规划器支持开放世界假设（无论某个陈述是否已知为真，都可以判定其真值）。在开放世界假设下，机器人领域的描述可以存储在本体（Poli 等，2010）中，即一个高层知识库。随后，描述逻辑（相关形式化方法的综述见 Baader 等（2003b））提供操作符（例如一阶逻辑算子，Smullyan（1995）），使得人们能够对知识库中编码的抽象概念和关系进行推理，并通过逻辑推理将它们实例化到现实世界中。这对于资源数量众多的复杂场景尤其有用，例如自主手术（Nagy 等，2018；Gibaud 等，2018；

表 1：使用逻辑程序进行机器人任务规划的最相关框架的表达能力。

	标准逻辑	时序逻辑	概率逻辑
SWI-Prolog（Wielemaker 等，2012年）	✓		
Readylog（Kirsch 等，2020年）	✓		
Clingo（Gebser 等，2019）	✓		
ROSPlan（Cashmore 等，2015年）	✓		
LTLMoP（Finucane等人，2010年）		✓	
TuLiP（Wongpiromsarn等人，2011a）		✓	
Problog（De Raedt等人，2007年）			✓
分布式子句（Gutmann等人，2011年）			✓
ProbCog（Jain等人，2009）			✓
概率 CTL（Rutten 等人，2004年）		✓	✓

Ginesi 等，2019）、人机交互（Lemaignan 等，2010；Bruno 等，2017）以及复杂的工业（Diab 等，2019）和家庭（Tenorth 和 Beetz，2013）场景。描述逻辑具有很强的表达能力，能够对知识库实体的子类、属性以及更复杂的关系进行推理。然而，Vitucci 和 Gini（2019）以及 Nakawala 等人（2019）的研究表明，表达能力的提升也增加了计算负担，这对于有实时需求的机器人应用而言是一项重大缺陷。计算上更易处理的方法是逻辑编程（Lloyd，2012）；与描述逻辑相比，它的表达能力略低，例如不支持量词 ∀，∃ 也不支持层级类关系和属性。不过，正如本文后文将展示的那样，这仍足以应用于现实的机器人领域。此外，在第 5 节（及后续部分）中，我们将重点介绍一些可与本体集成、用于开放世界推理的逻辑编程框架。

将任务规划器分类为确定性和概率性规划器，对于形式化刻画而言仍然不够充分。其中一个缺点是，有时无法明确决定应采用确定性表示还是概率表示。事实上，一些表面上具有确定性的框架存在扩展，支持概率形式体系（Younes 和 Littman，2004年；Vidal，1999年）。此外，正如 Ghallab 等人（2004年）以及 Karpas 和 Magazzeni（2020年）所报告的，规划场景可以根据规范的表达能力，更好地分为三大类。在这一意义上，可以区分经典规划器，它们假设机器人通过瞬时动作与确定性环境交互；时序规划器，它们处理复杂的时序规格；以及概率规划器，它们放宽了确定性假设。本文将这一分类体系用于机器人任务规划的逻辑编程框架。尽管这种区分并不总是明确（见表 1），我们认为逻辑形式体系的表达能力是构建分类体系的良好依据，因为它涵盖了确定性规划与概率规划之间的区分等内容。

3 综述方法

本综述中提及的逻辑编程框架及其相关应用，均按照系统检索方法选取。具体而言，我们在 Google Scholar 和 Scopus 搜索引擎中检索了从相关书籍和论……

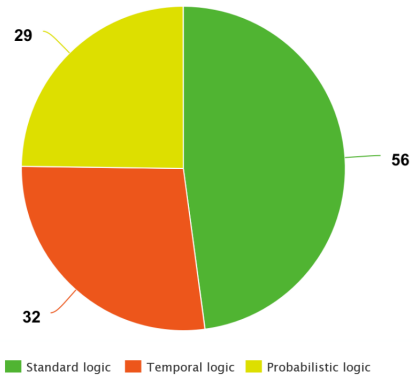

图 2：逻辑编程框架及相关机器人应用论文的分类（**共 114 篇**）。3 篇论文与概率 CTL 相关，因此同时归入时序逻辑和概率逻辑类别。

随后，我们仅选取了在真实机器人系统中有应用的框架及相关论文，因此排除了诸如 STRIPS（Fikes 和 Nilsson 1971）和简化动作结构（B¨ackstr¨om 和 Klein 1991）等广为人知的动作表示形式化方法——尽管 Karpas 和 Magazzeni（2020）等人并不反对在机器人学中使用这些方法。最后，为了向读者提供一份反映最新进展的指南，我们决定仅纳入过去 20 年中已在机器人领域得到具体应用的框架。这样一来，历史悠久的框架（例如 Van Emden 和 Kowalski（1976年）提出的 Prolog 和时序逻辑编程）仍然得以收录，但其最早的机器人应用（Kadonoff 1987；Tzafestas 和 Plouzennec 1989）则不予纳入。

如第2节所述，我们将论文按经典规划、时序规划和概率规划进行组织。在逻辑编程框架的背景下，这形成了对基础逻辑形式体系的分类，即标准布尔逻辑（经典逻辑算子）、时序逻辑 或概率逻辑。根据这一分类体系，表 1 展示了相关逻辑编程框架（即 ≥ 2 机器人应用案例）的分类，而图 2 则展示了不同框架及其相关应用在机器人领域中的分布。

下一节将介绍逻辑编程实现的基本概念，首先从机器人任务领域的动作语言描述讲起。

4 使用逻辑编程表示任务规划问题

从人类语言的任务描述出发，将其表示为一种动作语言，有助于正式定义任务实体和规范。动作语言为在任何特定逻辑程序中实现任务规划问题奠定了基础。本节以腹腔镜手术基础（Soper 和 Fried，2008年）中的代表性插销转移任务为例，展示其

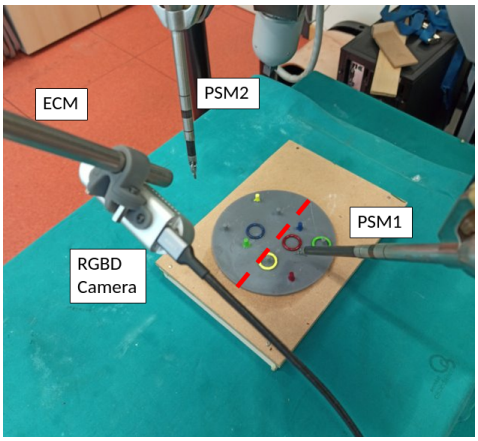

图 3：使用达芬奇手术机器人执行插销转移任务的设置。红色虚线标示了 PSM 的可达区域。

如何用 Kovacs（2011年）提出的先进动作语言规划领域描述语言（PDDL）进行形式化，并将其转换为 alogicprogram 的语法。

4.1 插销转移任务

插销转移（图 3）是初级外科医生的一项外科手术训练任务，在机器人手术系统上执行，例如，在使用先进的达芬奇手术机器人平台（直觉外科公司，美国加利福尼亚州桑尼维尔）学习操作时（DiMaio 等人，2011年）。该设置由两个患者侧操纵器（PSM）和一个安装了 RGBD传感器的内窥镜摄像机操纵器（ECM）组成。配备手术工具的 PSM 在插销底座上操作，底座上有（最多四个）彩色圆环，任务目标是将圆环放到颜色相同的插销上。执行工作流程会受到多项约束的影响。具体而言，为防止碰撞，需界定两个机械臂的可达区域。根据圆环和插销的可达性，单个机械臂可以拾取并放置圆环（例如图 3 中 PSM1 操作红色圆环），也可能需要将圆环从一个 PSM 转移到另一个 PSM（例如图 3 中的蓝色圆环）。圆环起初可能已放在插销上，因此移动到插销或转移点之前可能需要先将其取下。此外，彩色插销上可能放着颜色不匹配的圆环，因此可使用灰色插销作为圆环的临时放置点，直至任务完成。

本文选择插销转移作为典型的机器人应用场景，原因有三。首先，它属于拾取放置任务，因此是机器人领域和人工智能界中广为人知的应用案例。其次，该任务的规范引入了执行工作流程的多种变化。因此，枚举所有可能设置配置下的动作序列并不可行，直接基于任务知识进行推理是更便捷的规划方案。第三，插销转移目前是自主机器人手术（ARS）领域的基准任务（Nagy 和 Haidegger，2021）。自主机器人手术带来了安全性和可解释性方面的问题（Fiazza 和 Fiorini，2021），因此，这项任务适合作为本文所介绍的逻辑编程范式的应用案例。

4.2 从 PDDL 到逻辑编程形式化

根据第 1.2 节提出的定义，插销转移任务知识包括：

– 目标定义，即必须将环放置在颜色相同的插销上；– 资源，即环、插销（带有颜色属性）以及具有可能动作的机械臂。本文考虑四种主要动作，即移动到一个环或一个插销、转移和取出[2]；– 环境变量，用于描述资源之间的相对位置（例如，用于评估可达性或插销占用状态）；– 规范，如上一节所述。

PDDL[3] 允许用一种形式化表示来描述这些知识，其中规定了领域定义和问题实例。领域定义包含类型的声明，即资源中的对象类别（例如，类型为环或插销的物体）；用于表示领域特定特征的谓词名称及参数类型的声明（例如，表示环相对于插销的相对位置的 on(Ring, Color, Peg, Color)）；以及包含参数、前置条件和效果的动作。插销转移的领域定义如下：

```
(define(domain 插销转移)
    % 在实体之间定义相等关系（=），并允许在目标中使用量词
    (:requirements :相等关系, :通用前置条件)
    (:types 环,  Peg, Arm, 颜色)
    (:predicates (holding ?a,  ?r, ?c - 机械臂, 环,  颜色);
        (在 ?r,  ?c1, ?p, ?c2 - 环,  颜色,  Peg, 颜色)
        (可达 ?a,  ?p, ?c - 机械臂, Peg, 颜色)
        (可达 ?a,  ?r, ?c - 机械臂, 环,  颜色)
    (:action 移动
        参数: (?a - 机械臂; ?r - 环; ?c - 颜色)
        前置条件: (可达 ?a,  ?r, ?c)
        效果: (抓持  ?a, ?r, ?c))
    (:action 移动
        参数: (?a - 机械臂; ?p - 插销; ?c1 - 颜色)
        前置条件: (and (可达 ?a,  ?p, ?c1)
            (抓持  ?a, ?r, ?c2)
            (not (on ?r, ?c3, ?p, ?c1))))
            (not (on ?r, ?c2, ?p, ?c4))))
        效果:  (on ?r, ?c2 ?p, ?c1))
```

[2] 动作可以定义得更细致，例如考虑夹爪动作（打开和抓取）。为了使任务描述保持简洁，我们决定将这些动作隐含地纳入移动动作中（到达环后抓取，到达插销后释放）。[3]此处采用 https://helios.hud.ac.uk/scommv/IPC-14/ repository/kovacs-pddl-3.1-2011.pdf 中的 PDDL 3.1 语法。

(:action extract parameters: (?a - 机械臂; ?r - 环; ?c1 - 颜色)
precondition: (holding ?a, ?r, ?c1) effect: (not (on ?r, ?c1, ?p, ?
c2))) (:action transfer parameters: (?a1 - 机械臂; ?r - 环; ?c1 - 颜色)
precondition: (and (holding ?a1, ?r, ?c1) (not (on ?r, ?c1, ?p, ?
c2))) effect: (and (holding ?a2, ?r, ?c1) (not (holding ?a1, ?r, ?
c1)) (not (= (?a1, ?a2))))

给定领域定义，问题实例会指定任务的目标；初始状态*init*，即谓词的初始值；以及物体，即类型的实例（例如具有特定颜色的实际环和插销）。插销转移的一个可能问题实例如下：

(define(problem 插销转移 instance) (:domain 插销转移) (:objects (环 - 环) (插销 - 插销)
(blue - 颜色) (psm1 - 机械臂) (psm2 - 机械臂)) (: init (可达 psm1, 环, blue) (可达 psm2 插销
blue)) (:目标 (forall (?c - 颜色; ?r - 环) (and (on ?r, ?c ?p, ?c) (可达 ?a, ?r, ?c)))))

从这一高层表示出发，逻辑程序设计语言使用带排序签名 $\mathcal{D}$来编码任务知识，其中符号（项）可以按层次组织形成原子，即表示领域属性的项的谓词。项可以是变量或常量。常量项称为基项；当原子中的所有项都是基项时，该原子就是基项。根据是否依赖于时间，原子可分为流值 或静态项[4]。

为了阐明这些概念，请考虑 PDDL 中对插销转移任务的描述。环、插销、颜色 和机械臂 是静态变量，其可能的常量值由问题实例中的对象定义。以下提及类型作为尚未赋值的变量时，我们将分别使用简写`R, P, C, A`。流值是带有参数的环境特征（PDDL 谓词），即`reachable(A, P/R, C)`、`on(R, C1, P,C2)`、`holding(A, R, C)`，以及动作`move(A, R/P, C)`、`extract(A, R, C)`、`transfer(A,R, C)`。

[4] 正如下一节将阐明的那样，具体取决于所采用的逻辑编程框架，可以通过使用时间变量来明确表示对时间的依赖。

逻辑编程语言还引入操作符，用于编写规范。根据底层逻辑形式体系的不同，可以使用不同的操作符，例如时序操作符（如*until* 和*release* ）或布尔操作符（如逻辑蕴含和合取）。下一节将对此作进一步说明。例如，考虑插销转移任务的 PDDL 描述中的`move(A, P, C)`。前置条件和效果可以表示为原子之间的蕴含关系：

$$\begin{aligned} \texttt{move(A, P, C1)} \leftarrow\ & \texttt{reachable(A, P, C1)} \wedge \texttt{holding(A, R, C2)} \wedge \\ & \neg\texttt{on(R, C3, P, C1)} \wedge \neg\texttt{on(R, C2, P, C4)} \\ \texttt{on(R, C2, P, C1)} \leftarrow\ & \texttt{move(A, P, C1)} \end{aligned}$$

左侧通常称为头部，右侧则称为规范的体部 。类似地，可以将约束定义为 $\bot \leftarrow$ 原子，其中$\bot$ 逻辑假。例如，`move(A, P, C)` 的前置条件也可以表示为：

$$\begin{aligned} \text{移动(A, P, C1)} \leftarrow\ & \text{可达(A, P, C1)} \wedge \text{抓持(A, R, C2)} \\ \bot \leftarrow\ & \texttt{on(R, C3, P, C1)} \wedge \texttt{move(A, P, C1)} \\ \bot \leftarrow\ & \texttt{on(R, C2, P, C4)} \wedge \texttt{holding(A, R, C2)} \wedge \texttt{move(A, P, C1)} \end{aligned}$$

给定一个逻辑程序，特定于语言的求解器 会根据可用证据对环境变量进行初始基化，并以基化动作集的形式计算计划。基化会将程序转换为一组命题逻辑公式，其中基化原子会成为布尔变量。然后利用逻辑演绎，根据规范传播基化原子，直至找到计划。例如，考虑上述 PDDL问题实例。初始条件满足`move(psm1, ring, blue)`的前置条件。这会使相应效果`holding(psm1, ring, blue)`得到基化。接着，`transfer(psm1, ring, blue)` 的前置条件成立，其效果会使`holding(psm2, ring, blue)` 得到基化，最后执行`move(psm2, peg, blue)` 这一动作，从而完成该问题实例。

在详细说明不同逻辑程序框架如何针对表示和求解进行专门化之前，我们先概述本综述的主要成果，以便引导感兴趣的读者根据自身的具体需求，在相关章节中深入了解。

5 机器人任务规划的逻辑编程框架概述

表2总结了逻辑编程用于机器人任务规划的研究现状分析结果。任务规划所采用的逻辑程序框架取决于若干标准，这些标准将在第6至第8节末尾进一步阐述：

5.1 逻辑形式体系的表达能力

必须考虑底层逻辑形式体系的表达能力。我们的综述发现，逻辑编程在机器人任务规划中有多种应用，包括家庭/服务场景、导航、工业、搜索与救援、人机交互、多机器人协同、监视和操作任务。这些应用中的大多数

Table 2: Summary of the main reviewed frameworks for robotic task planning through logic programming. Legend for entries: $\mathscr{F}$ = formalism (*Pl* = Prolog); *ROS* = ROS support; *Impl.* = software implementation; *Integrated* specifies whether the framework is stand-alone or is integrated with other modules; *Online* = online adaptation / plan revision. Legend for applications: NAV=navigation; S&R=search&rescure; DOM=domestic environment; IND=industry, SRV=service; HRI=human-robot interaction, MRC=multi-robot coordination; MAN=manufacturing; SUR=surveillance; LOC=locomotion, MANIP=manipulation.

	$\mathscr{F}$	**Applications**	**Impl.**	**ROS**	**Integrated**	**Online**
SWI- Prolog (Wielemaker et al. 2012)	Pl	DOM, SRV, NAV, IND, S&R	C++, Java, Python	✓	In CRAM (Beetz et al. 2010b) (open-world), Tartarus (Semwal et al. 2015) (MRC)	✓
Readylog (Kirsch et al. 2020)	Pl	SRV, DOM	C++	✓		✓
Clingo (Gebser et al. 2019)	ASP	DOM, SRV, IND	C, Python	✓		✓
ROSPlan (Cashmore et al. 2015)	PDDL	NAV, HRI, MRC		✓		
LTLMoP (Finucane et al. 2010)	LTL	NAV, S&R, HRI, MRC	Python	✓		
TuLiP (Wongpiromsarn et al. 2011a)	LTL	SUR, LOC	Python		Hybrid controller design	
Problog (De Raedt et al. 2007)	DS	MANIP	Python, Java			✓
DC (Gutmann et al. 2011)	DS	MANIP				✓
ProbCog (Jain et al. 2009)	BLN	DOM, MANIP, NAV	Python, Java	✓	Statistical learning&reasoning	✓
PCTL (Rutten et al. 2004)	CTL	MAN, S&R, MRC				

标准逻辑规划器能够涵盖这些内容，它们支持指定条件律、约束和优化语句，以生成最优计划。基于 Prolog 的先进规划器 SWI-Prolog（Wielemaker 等人，2012 年）（Scopus 上有 > 250 篇引用）能够处理不同的数据类型，包括实变量，帮助设计者为处理传感器连续数据的真实机器人定义适当的规范。基于答案集语义的规划器在这方面仍然有限，不过，其主要代表 Clingo（Gebser 等人，2019年）（Scopus 上有 > 50 篇引用）允许定义算术约束。此外，Clingo 还支持多种计划计算方式，并支持用于最优规划的偏好推理。

当需要通过复杂的机器人智能体与环境之间的时序关系来恰当地描述场景时，应优先选择时序规划器。例如，需要长时间人机交互的任务，或对大型复杂环境进行（多机器人）探索，就属于这种情况。此外，时序逻辑能够轻松表达系统理论中的条件，例如集合的不变性和可达性（如用于定义系统稳定性）。因此，对于任务与运动规划紧密集成的应用（如多机器人探索），基于时序逻辑的规划器具有优势。为此，TuLiP工具箱（Wongpiromsarn 等人，2010年）（Scopus 上有 > 100 篇引用）是先进的解决方案。不过，值得指出的是，某些时序概念也可以用标准逻辑规划器表达，例如像 Son 等人（2006年）和 Meli 等人（2021a）那样，在答案集语义中使用事件演算，但这会增加表示的知识规模（例如，需要额外的时间变量或规范）。此外，答案集编程的时间扩展近来已经问世，例如 *telingo*（Cabalar 等人，2019a），但尚未应用于机器人案例。因此，逻辑形式体系的选择应根据应用领域的特点，以及最重要、最常见的待表示概念（例如时序概念或因果概念）来确定。

概率规划器适用于需要与环境持续交互的任务，这类任务要求对不确定的传感器信息进行推理，以完善对场景的认识。例如，操作和导航任务就属于这种情况。主要实现包括 Problog（De Raedt 等，2007年）（在 Scopus 上有 > 450 次引用），它采用类似 Prolog 的语法对离散概率分布进行推理；以及 Gutmann 等人（2011年）提出的分布式子句（DC）（在 Scopus 上有 ≈ 50 次引用），用于表示连续概率分布。

5.2 计算效率

一项重要要求是计划生成的计算效率 。这与底层逻辑形式体系的表达能力有关。事实上，一般而言，在经典规划中，判定一个问题是否可解是 PSPACE 完全问题（Bylander 1994）；考虑时序规划时，该问题则成为 EXPSPACE 完全问题（Rintanen et al. 2007）；而考虑连续变量时，甚至可能变得不可判定，例如概率规划（Helmert 2002）。因此，针对具体的机器人应用，谨慎选择最合适的逻辑编程框架十分重要，并需在表达能力与效率之间确定最佳权衡。一般来说，应优先选用标准逻辑编程框架，因为它们也实现了效率最高的求解算法，并能够表示具有实际规模的机器人智能体、动作和环境特征的复杂任务规划领域。尤其是，基于答案集语义的规划器相较于基于 Prolog 的规划器具有一些优势，例如它们

不要求预先对任务知识进行分层，而且其中语句和原子的顺序并不重要。

相反，只有在场景的复杂度（资源数量和规范数量）有限时，才应考虑时序和概率逻辑编程框架。事实上，由于状态爆炸问题，这些程序的求解效率无法随着问题规模的扩大而良好扩展，除非具体用例足以证明采用严格假设是合理的，例如限制计划的时间范围或忽略不重要的规范。需要更强的表达能力时，目前最佳方案是依赖将标准逻辑编程框架与特定工具相结合的框架，例如，将答案集语义与马尔可夫决策过程结合使用（Sridharan 等人，2019年），或使用 *telingo*（Cabalar 等人，2019a）在 Clingo 中表示时序算子。

5.3 软件实现

软件实现与编程语言支持 是选择逻辑编程框架、构建集成多种审议功能的机器人架构时的重要因素。标准逻辑编程在机器人领域已相当成熟，因此可与用于机器人研究的标准机器人操作系统（ROS）（Koubˆaa 等人，2017年）集成。ROSPlan（Cashmore 等人，2015年）（>150 Scopus 上的引用次数）是一个尤为灵活的框架，因为它对任务规范不要求采用任何特定的逻辑形式体系，而是采用通用的 PDDL 解释。Clingo 和 Prolog 也提供了用于与兼容 ROS 的语言集成的应用程序接口。

唯一提供与 ROS 直接集成的时序规划逻辑编程框架是 LTLMoP（Finucane 等人，2010）（>100 次 Scopus 引用）；而 ProbCog（Jain 等人，2009）（≈ 25 次 Scopus 引用）最初是作为 CRAM 项目的概率扩展而开发的（ Beetz 等人，2010b），因此内置了 ROS 接口。不过，TuLiP 和 Problog 等相关框架支持 Python 和 Java，因此可以轻松集成到 ROS 框架中。

5.4 对开放世界规划的支持

对开放世界规划的支持是复杂机器人应用的一项有趣特性，例如，在包含许多资源的大型环境中进行探索时，可以查询外部知识库，以获取有关领域的有用信息。一般而言，在这类场景中，基于 Prolog 的工具性能最佳，因为它们原生支持基于查询的推理。SWI-Prolog 是 CRAM 项目中用于开放世界规划的成熟组成部分；此外，新近推出的 DLVHEX（Eiter 等人，2018年）采用答案集语义，能够轻松集成外部知识库，不过截至本综述撰写时，尚无已知的机器人学应用。在概率规划器中，ProbCog 是 CRAM 项目自然而然的统计扩展。

5.5 计划修订机会

最后，计划修订是在线机器人任务规划器的一项关键能力，因为它能够应对许多机器人场景中出现的动态状况。基于 Prolog 的规划器

以及答案集语义都依赖于非单调逻辑形式体系，因此它们天然支持在线计划适应。

此外，一些概率规划器也支持计划修订，因为它们通常是基于 Prolog 的规划器的随机扩展。大多数时序规划器不提供此功能，因为它们基于单调的线性时序逻辑变体。这在一定程度上是由于时序规划器的适用范围不同：它们旨在生成构造即正确的计划。

6 标准逻辑编程

“标准逻辑 ”这一术语指基于经典布尔逻辑的逻辑编程框架，其中规范使用否定、合取、析取和蕴含等操作符来表示。为了表示动作和流值的时间序列，必须定义一个显式的时间变量 `t` 。因此，以插销转移领域为例，流值需增加一个参数`t` （例如，`on(R, C1, P, C2, t)`）。采用这种形式体系时，重要的是要在动作及其效果之间引入显式的时间延迟，例如，在插销转移任务中，对`move(A, P, C, t)` 而言：

$$\texttt{on(R, C1, P, C2, t+1)} \leftarrow \texttt{move(A, P, C2, t)}$$

这样便可生成动作时序。

此类逻辑编程框架可分为两大类：基于 Prolog 的框架（第 6.1 节）和基于答案集语义的框架（第 6.2 节），二者的主要区别在于求解策略。第 6.3 节还介绍了其他特定于应用的实现。

6.1 基于 Prolog 的规划器

最早用于规划的标准逻辑编程语言之一是*Prolog*，由 Van Emden 和 Kowalski（1976 年）提出。具体而言，Prolog 是一种用于非单调逻辑编程的语言，也就是说，其底层逻辑是非单调的（McDermott 和 Doyle，1980年）。要理解非单调性，可以考虑一个通用任务知识 K ，它表示为由目标、资源和规范组成的集合，以及一个布尔断言 A ，该断言由 K 推出，即 $K \models A$。如果添加新知识S （例如，更多规范或新证据）不会影响蕴含关系的真值，即 $K \cup S \models A$，那么该知识就称为单调的。反之，则称为非单调的。知识的非单调性使人工智能领域得以解决著名的资格问题；这一研究始于 McCarthy（1980年）对人类推理的 circumscription（界定）概念，并促成了多种用于专家系统决策的逻辑框架的发展，例如演绎数据库（Eiter 等，1997年）、溯因理论（Alferes 等，2000年）和稳定模型语义（Dimopoulos 等，1997年）。在机器人任务中，基于非单调性假设进行规划至关重要，因为环境通常并非完全已知，或会动态变化。以插销转移任务为例，为了模拟手术场景中的解剖结构变异，在任务执行期间，环可能会在场景中移动（例如被人移动），从而改变可达性条件。因此，求解器必须持续更新计划，并给出相应指令，例如要求在手臂间转移，而非直接放置到插销上。

Prolog 的语法通过失败即否定 （NAF）扩展了标准逻辑形式体系。为了说明 NAF 与经典逻辑否定之间的区别，考虑一个通用布尔变量 p。要在经典逻辑中断言 $\neg p$ ，必须证明其否定

p 成立；相反，要证明`not(p)`（Prolog 中的 NAF 语法），只需满足 p 不成立。NAF 在 Prolog 中有双重作用。首先，它引入了封闭世界假设，因为在求解过程中，NAF 被视为经典否定（换言之，未知事实均假定为假）。其次，NAF 支持非单调范式，因为可以指定某个原子不成立，并在获得新证据后对其进行修正。例如，插销转移任务的目标可以表示为以下约束：

```
:- reachable(A, R, C, t), not(on(R, C, P, C, t)).
```

其中 `:-` 表示 $\leftarrow$ 标准布尔逻辑中的 $\leftarrow$（在这个特定情况下，它等于 $\bot \leftarrow$，因为 $\bot$ 在标准 Prolog 语法中被省略）。只有当任何可达的环都不再满足基项 `on(R,C,P,C,t)` 时，才满足此规范（也就是说，场景中的所有环都放置在同色的插销上时）。`on(R,C,P,C,t)` 可以通过外部传感器信息或规范（例如`move(A,P,C,t)` 动作的效果）进行基项化。

从 Colmerauer（1990年）的 Prolog III 到 Narboni（1999年）的最新版本 Prolog IV，Prolog 进一步扩展了标准逻辑语法，加入了列表和树等结构，以表示原子之间更丰富的层次关系；它还能够管理实变量上的约束，这对于涉及连续环境变量和运动学变量的机器人应用十分有用（例如，插销转移任务中环和插销的三维位置）。

这些特性提升了 Prolog 在许多具有挑战性的机器人场景中的吸引力。值得关注的是，许多基于 Prolog 的实现采用了 Golog 动作语言（Levesque et al. 1997），例如：Janssen et al. (2016) 将其用于 RoboEarth 项目（Waibel et al. 2011）中的多机器人分层任务规划；Kirsch et al. (2020) 通过 C++ 绑定提出了将 Golog 方言 Readylog 与 ROS 有效集成的方法（Golog++ by Matar´e et al. (2018)），并将其应用于 Pepper服务机器人；Gierse et al. (2016) 介绍了 INTRGOLOG，用于在发生异常事件时实现任务中断与恢复执行；Schiffer et al. (2012) 使用 Readylog 在 Caesar机器人中进行家庭场景下的任务规划；Farinelli et al. (2007) 在部分可观测条件下，将 TEAMGOLOG 用于搜救应用中的多机器人协同。在 Prolog 的主要实现中，还应提及 SWI-Prolog（Wielemaker etal. 2012），它实现了基于偏好推理范式（Brewka et al. 2008）的最优查询应答和计划生成构造，并曾用于例如 Tartarus框架，以集成和协调信息物理系统中的多个机器人（Semwal et al. 2015）；Mu˜noz-Hernandez 和 Wiguna (2007) 使用模糊Prolog处理实时机器人足球中的信息缺失；Javia 和 Cimiano (2017) 使用 Prolog 实现家庭活动中的多机器人协同；Xu et al. (2014) 实现了定性Prolog演算与定量 C++ 处理之间高效的软件集成，用于辅助老年人和残障人士；Pineda et al. (2013) 提出了 Sit-Log，这是一种用于家庭场景下服务机器人任务的基于Prolog的规划器，应用背景为 RoboCup@Home 国际竞赛；Nevlyudov et al. (2008)；Fakhruldeen et al. (2016) 则研究了工业中的机器人装配。

正如 Narboni（1999年）所详述，Prolog 将任务知识表示为树结构，遵循变量-原子和原子-运算符层级。例如，考虑表示插销转移任务知识中`move(A,R,t)` 前置条件的语句：

```
move(A, R, C, t) :- reachable(A, R, C, t).
```

`A`, `R`, `C`, `t` 变量是相应谓词原子`move`, `reach-able`的叶节点。此外，在蕴含操作符之后，`reachable` 是`move`的叶节点。借助这样的树结构，Prolog 求解器能够通过实现回溯算法来响应用户查询。

从用户查询和可用的初始信息（例如来自传感器的信息）出发，Prolog 求解器沿着树的分支执行回溯搜索，以验证查询中的原子是否可以被实例化为基项。此外，树中最后探索的分支会保存在栈中，以便当某个分支没有返回解时回溯到根节点并重新开始搜索过程。有关该算法的更多详情，请参阅 Narboni（1999）及相关文献。研究人员一直在尝试提高这一求解过程的效率，例如采用基于生成器/消费者范式的智能回溯（Kumar 和 Lin，1988），或采用 Pereira 等人（1986）提出的并行与树验证。近期，Zhang 和 Hong（2019）提出了一种基于线性模型约简的方法。

回溯要求预先对程序进行分层，以确保求解过程能够终止。分层保证一个（命题）逻辑程序可以划分为若干递增的子句子集 $P_1 \cup ... \cup P_n$，使得：

– 如果谓词 p 在 P_i中以肯定形式出现，那么它就在 $\bigcup_{j \leq i} P_j$中定义；– 如果谓词 p 在 P_i 中以否定形式出现，那么它就在 $\bigcup_{j<i} P_j$ 中定义。

由于基于 Prolog 的程序以层次树结构表示知识，这种子集组合确保树搜索不会陷入不同层级定义的谓词之间的循环。然而，Drabent（1996）的研究表明，并非所有逻辑程序都可以分层。

6.2 基于答案集语义的规划器

一种语法与 Prolog 类似的非单调逻辑编程语言是答案集编程（*ASP*）（Calimeri 等，2020年）。答案集编程由 Marek 和 Truszczy´nski 于 1999年提出，Lifschitz（1999年）首次提出用它解决自主智能体的规划问题，例如 Dimopoulos 等人（2019年）的应用。答案集编程基于稳定模型语义（Gelfond 和 Lifschitz，1988年）。为了理解逻辑程序的稳定模型或答案集（Lifschitz，1999年）是什么，考虑一个经典的（即不含 NAF 的）命题逻辑公式 F，它取决于一组 X 布尔变量。F相对于 X的约化，表示为 F^X，是将所有以否定形式出现的变量替换为假 $\bot$后得到的公式。接着，我们将 F的稳定模型（或答案集）定义为满足 F^X 的最小模型，即一个没有任何真子集满足F^X的集合 $Y \subseteq X$。例如，令 F为：

$$(q \rightarrow p) \wedge (\neg r \rightarrow q) \wedge (p \rightarrow s)$$

$Y = \{s, p, q\} \subset X = \{s, p, q, r\}$是 F的一个答案集，因为它满足 F^X：$(q \rightarrow p) \wedge (\neg\bot \rightarrow q) \wedge (p \rightarrow s)$，且 Y没有任何真子集满足该条件。如果 F 包含默认否定，则会将其转换为经典否定（闭世界假设），然后计算稳定模型。

参考插销转移领域，考虑转移动作的前置条件：

```
transfer(A, R, C1, t) :- holding(A, R, C1, t), not on(R, C1, P, C2, t).
```

其中 并非 答案集编程的 NAF 默认否定语法，与 Prolog 类似。在传感器提供初始基项的情况下 {t = 1, A = psm1, C1 = red, holding(psm1, ring, red, 1)}， not on(R, C1,

P, C2, t) 被转换为 ¬on(ring, red, peg, , 1)，其中表示C2的任意可能赋值。因此，生成以下命题逻辑公式：

$$\texttt{transfer(psm1, ring, red, 1)} \leftarrow \texttt{holding(psm1, ring, red, 1)} \wedge \neg\bot$$

其稳定模型为{transfer(psm1, ring, red, 1), holding(psm1, ring, red,1)}。换言之，可以执行转移动作。从这个例子可以看出，一个答案集包含所有基原子，也就是说，不仅包含动作，还包含环境流值。这是与基于 Prolog 的系统的一个主要区别；基于 Prolog 的系统则是响应用户的特定查询。答案集的计算增强了任务规划器的可解释性，它会返回任务规划问题的输入（环境上下文）和输出（动作序列）。

一般而言，稳定模型并非唯一。例如，考虑描述插销转移任务的 ASP 程序，就能明显看出这一点。当插销转移领域中存在多个环时，它们的顺序是任意的，因此可以生成多个计划，每个计划中的环排列顺序都不同。在这种情况下，每个计划都属于该逻辑程序的一个独立答案集。

答案集编程相较于 Prolog 引入了额外的结构。从最早的求解器 Smodels（Syrj¨anen 和 Niemel¨a，2001年）和 DLV（Eiter 等人，2000年），到最新的扩展版本 DLV2（Adrian 等人，2018年）和 Clingo（Gebser 等人，2019年），都可以定义聚合，形式为l{a : b}u。聚合定义了一组原子a，并受前置条件b约束，使得该集合的基数（即答案集中基化原子a 的数量）介于l 和u之间。该结构可转换为选择规则，有助于在任务规划领域中生成不同的动作序列。例如，插针转移场景中可能存在多个环，从其中任意一个开始都可以完成任务。不过，也可以在聚合中为动作添加前置条件，例如move(A, R, C, t)：

```
0{move(A, R, C, t) : reachable(A, R, C, t)}1.
```

这样，每个时间步进部分的答案集中最多只会基项化一个动作，因此最终只会生成一个（随机）计划。类似地，也可以定义带有外部条件的聚合规则：

```
0{move(A, R, C, t) : reachable(A, R, C, t)}1 :- arm(A).
```

在这种情况下，求解器会为每个 *PSM*最多基项化一个动作，这样两个机械臂便可以同时在不同的环上作业，从而缩短完成任务所需的时间。

选择规则可以通过答案集编程中的偏好推理（Brewka 等，2008年）与弱约束来支持，也可以使用最先进的 Clingo 系统引入相应的优化语句（Gebser 等，2019年）。弱约束的表达语法如下：

$$\texttt{:\sim}\ b_{1..n}(V_{1..m}).[w@p, V_{1..m}]$$

其中 $b_{1..n}$ 是依赖于 $V_{1..m}$ 变量的体原子，w 是权重，p 是优先级。上述语法的含义是，对体原子同时进行基化会产生等于该权重的代价。答案集必须 确保满足采用常规语法 :- atom 定义的硬约束，此外还应尽可能 使弱约束的权重最小化。如果基化了多个弱约束，则必须最大化累积权重，但优先考虑p较低的约束。关于

以插销转移任务为例，假设要最大限度地提高运动经济性，例如先移至最近的环（对任意 PSM 而言）。这可以通过在领域描述中引入一个新的流 `distance(A, R, C, X, t)` 来轻松表达，该流将机械臂与环之间的距离定义为x[5]。然后，可以添加以下弱约束：

```
:~ move(A, R, C, t), distance(A, R, C, X, t).[X@1, A, R, C, X, t]
```

以最小化到目标环的距离。与弱约束表示法等效的 Clingo 语法是优化语句：

```
#minimize{X: move(A, R, C, t) : distance(A, R, C, X, t)}.
```

ASP 求解器采用继承自可满足性（SAT）检查领域的方法计算答案集（Biere 等，2009）。基本方法包括对初始已知的原子（例如来自传感器的环境信息）进行自底向上实例化，并逐步验证逻辑蕴含。在此需要特别提到 Clingo，它允许将 ASP 程序组织成若干部分或子程序，具体包括基础部分、步进部分 和检查部分。在任务规划领域中，基础部分通常包含资源的定义，规范位于步进部分，而目标则位于检查子程序中。求解器在每个时间步检查检查部分，并且仅当目标尚未满足时，才递增时间变量以对规范进行基化。这样，计划的计算会持续直到目标不再满足，从而实现时序逻辑形式体系的某些能力。

与主要基于带失败否定的选择线性确定子句（SLDNF）的 Prolog 树搜索（Apt 和 Doets，1994）不同，答案集编程求解器可以保证终止性（Lifschitz，2008年）。此外，SAT 求解在计算机科学的众多领域都有应用。因此，从广受欢迎的 MiniSAT（Sorensson 和 Een，2005年）开始，许多研究致力于提升现有 SAT 求解器（本质上是 NP 完全问题的求解器）的效率，即使是在复杂领域中也是如此（近期进展见 Manthey，2016年），例如在 Clingo 中对规范进行并行求值（Kaufmann 等，2016年）。此外，答案集编程语法要求用户定义安全语句，也就是说，头部中的所有变量都必须同时出现在体部中，且不能带有 NAF。这样一来，将体部中的基化原子传播到头部会更加高效，因为头部中的变量会继承体部中的特定赋值。

尽管答案集编程在人工智能规划中有大量应用（近期综述见 Erdem 等人（2016年）），但其在机器人学中的应用仍少于基于 Prolog 的系统；后者已经成熟，并且发展历史长得多。这也与高效 SAT 求解算法的研究进展有关，这类研究直到近几年才取得显著进展。目前，答案集编程在机器人场景中最受欢迎的实现是 Clingo；从最新版本（Clingo 5）起，它也支持实数约束。

Pianpak 等人（2019）提出了 Clingo 的相关应用，利用其面向多机器人物流和仓库自动化的变体 Asprilo（Gebser 等人，2018a）；Opfer 等人（2018）将 Clingo 系统与 ALICA 语言相集成，以支持家用及通用环境中的团队协作（Opfer 等人，2019）；Cui 等人（2021）将其用于通用服务机器人学；Lu 等人（2017）提出了一种任务规划框架，使家用服务机器人能够在 RoboCup@Home 挑战赛中接收人类以自然语言提供的信息和目标描述；Ginesi 等人（2020）；Meli

[5] X 变量在答案集编程中只能取整数值。因此，需要将领域中物体之间的实际距离转换为整数，例如按照 Ginesi 等人（2020年）所示，对所有距离进行排序。

等人（2021b）；Tagliabue 等人（2022）提出了一个可解释的外科任务—运动规划集成框架；Andres 等人（2015）展示了将 Clingo 集成到 ROS 中的 ROSoClingo 软件包。Wang 等人（2009）提出了一个基于 DLV 系统的近期应用，旨在解决工业中的机械装配序列问题。

Tu 等人（2007年）提出了其他几种答案集编程的实现，其中介绍了基于答案集的条件规划器（ASCP），并将其应用于机器人示例；Chen 等人（2013年）在 OK-KeJia 服务机器人原型上实现了答案集编程，用于在人机交互情境下基于多源信息进行规划与决策；Baral 和 Son（2015年）证明了使用答案集编程进行任务规划，以及在人机交互中进行故障分析和解释的优势；Ji（2009年）将其用于家庭服务场景；Sridharan 等人（2019年）将答案集编程与马尔可夫决策过程相结合，应用于家庭场景；Bertolucci 等人（2021年）将其用于双臂操作；Falkner 等人（2018年）回顾了答案集编程在工业领域的近期应用，而 Erdem 和 Patoglu（2018年）则综述了其在各种机器人领域的近期应用。最后，我们提到，答案集编程在调度问题中的应用近来日益受到关注，包括运输／路径规划（Gebser 等人，2018b；Abels 等人，2019）和空间分配（Dodaro 等人，2021、2022）场景。尽管目前尚无机器人应用实例，我们认为，答案集编程应用于日益热门的机器人问题颇具潜力，例如自动驾驶车辆以及工业／服务自动化设计。

6.3 其他规划器

虽然基于 Prolog 和 ASP 的规划器最为流行，且可应用于广泛的领域，但也存在其他基于标准逻辑程序的规划器实现。例如，约束处理规则（Fr¨uhwirth 1998）可通过在线简化任务特定约束来提高求解过程的性能。这种方法的一个实现是 fluent executor（Thielscher 2005），它曾用于办公室机器人的自动化，例如一项涉及人机交互的应用（Wu 和 Liu 2008）。

类似的、面向特定应用的方法直接依赖于用动作语言（例如 PDDL）对任务进行的通用描述，并针对具体用例实现定制的解决方案。这类规划器的一个重要示例是 ROSPlan（Cashmore 等，2015年），这是一个结合了基于 PDDL 的任务规划、运动规划、感知和开放世界规划的框架。基于 ROS 基础设施为适用于大多数机器人系统带来了显著的实现优势。ROSPlan 已应用于多个场景，例如 Miranda 等人（2018年）将其用于多机器人导航；Sanelli 等人（2017年）将其与 Petri 网集成，用于人机交互；Escudero-Rodrigo 和 Alquezar（2016年）则用它解决锚定问题，即将高层知识中的抽象符号与感知结果及已执行的动作相匹配，以便自动生成 ROS 节点代码。

其他基于 PDDL 衍生而来的定制实现示例，则建立在动作图之上（Harman 和 Simoens, 2020年）。Gragera 等人（2019年）提出了一种基于 PDDL 的社交机器人学规划器；Moon 和 Lee（2019年）研究了无人地面车辆与空中车辆的协调；Mu˜noz 等人（2019年）实现了一个集成框架，用于机器人机械臂、监视和漫游车探索；Ma 等人（2016年）专注于 ROS 框架中的移动机器人；SHOP 是 Nau 等人（1999年）提出的，用于实现一种先进的

分层任务网络（HTN）（Erol，1996年）方法。它是一种流行且高效的形式化方法，可用于带有子目标的任务规划。

这里需要强调的是，从任务的通用 PDDL 描述衍生出的特定任务方法并非唯一选择；也可以采用其他定制实现，例如基于 ASP 和类似 Prolog 的语法，并根据具体应用场景的需求进行调整。例如，在 Dix 等人（2003）的研究中，AnsProlog（用于答案集语义的语言）被用于实现 HTN，其效率与 SHOP 相当（更全面的比较见 Son 等人（2006））。此外，Jiang 等人（2019）最近比较了基于 PDDL 和基于 ASP 的规划器。作者在不同实验场景中（包括机器人导航）将作为 ASP 最先进实现的 Clingo，与两种最突出的基于 PDDL 的 FastDownward 规划器 FDSS-1（Helmert 等人，2011年）和 LAMA-2011（Richter 等人，2011年）进行了比较。比较结果表明，当任务所需的计划由较长的动作序列构成时，基于 PDDL 的系统通常效率更高。然而，对于较短的计划，基于 ASP 的系统在计算性能方面表现更好，并且随着领域描述规模（即资源和具体规范的数量）的增大，其扩展性显著更佳。

6.4 讨论

基于 Prolog 和答案集编程的系统是机器人学中解决任务规划问题的主要方案。它们的主要优势在于对任务知识进行非单调表示，这对于机器人应用至关重要，因为当环境条件发生变化时，这种表示能够支持计划修订。基于 Prolog 的规划器，例如 SWI-Prolog（Wielemaker 等，2012年），属于基于查询的规划器，还支持对连续变量进行高级推理，从而弥合机器人场景的离散逻辑表示与对环境的连续感知及交互之间的差距。然而，这可能会降低在线求解规划问题的计算效率（Helmert，2002年）。相比之下，基于答案集编程的框架，例如 Gebser 等人（2019年）提出的 Clingo，会直接计算出更易于解释且信息更丰富的答案集，即由基项构成的集合，这些基项表示当前的环境条件以及完成给定任务的可能计划。此外，这些框架还支持生成多个计划，并进行偏好推理，以实现最优规划。

所有标准逻辑编程框架都允许通过引入一个额外变量来进行时间索引，从而表示时序概念（例如，时间序列或事件驱动属性）；本节开头以及 Son 等人（2006）和 Meli 等人（2021a）对此均有说明。特别是，Abadi 和 Manna（1989）中介绍了时序 Prolog 扩展的示例；近期，基于 Clingo 求解器和时序均衡逻辑（Aguado 等人，2013年）的 ASP 语法时间扩展 *telingo*（Cabalar 等人，2019a）和 *tasplan*（Cabalar 等人，2019b）也已被提出。尽管作者尚不了解这些范式在机器人应用中的相关案例，但 ASP 和 Prolog 的时间扩展未来很可能会有广泛的应用，因为它们具有不可估量的优势，例如在任务知识表示中的非单调性。

基于这些原因，研究社区投入了大量精力来实现时间和内存高效求解器，尤其是在面向基于 ASP 的规划器的 SAT 求解领域。这有助于平衡表示的知识规模增长与计算成本，例如，加入用于时序推理的额外变量和规范时便是如此。此外，Clingo 和 SWI-Prolog 等先进框架提供 C / C++、Java 和 Python API，并且能够轻松集成 ROS。

此外，一些标准逻辑编程语言支持开放世界假设，可集成外部知识库，以便在计划执行过程中按需查询，从而在非常复杂的领域中实现模块化且更高效的知识表示。实际上，知识库对场景进行详细描述，而逻辑程序仅编码任务规范，并查询知识库以进行选择性基化。这种方法最著名的例子是 Beetz等人（2010b）提出的 CRAM框架，用于高级机器人操作；该框架基于 Knowrob（Tenorth 和 Beetz，2017）本体，适用于通用服务机器人任务。知识通过 SWI-Prolog 查询。例如，Yazdani等人（2019）验证了这一方法，完成了一项户外搜索与救援机器人任务，其中涉及环境的几何推理和语义建模；Crespo等人（2018）研究了多目标导航任务；Beßler等人（2018）研究了工业中的不完整装配任务；Gonc.alves 和 Torres（2015）将骨科专用知识库集成到骨科手术任务规划中；Papadimitriou等人（2015）则研究了自主水下航行器执行反水雷任务这一安全关键场景，解决了硬件故障恢复及优先级变化带来的挑战。另一种基于答案集语义的实现，是近期对 DLV 的扩展 DLVHEX（Eiter等人，2018），它提供了用于查询外部知识库的专用结构。虽然基于查询的方法无法提供与描述逻辑相同的表达能力，但这种集成逻辑编程与本体的方法通常效率更高，并保留了非单调假设，而这对大多数应用至关重要。

7 时序逻辑编程

标准逻辑编程具有足够的表达能力，可以描述机器人学中的许多任务规划问题，其中操作员表示原子之间的因果关系。然而，在若干场景中，要描述资源之间复杂的时序关系，还需要更强的表达能力。本节旨在介绍基于时序逻辑形式体系的逻辑编程解决方案。机器人应用中的主要时序算子包括：

– 最终 $\Diamond a$，规定陈述或原子 a 应在未来某个时刻成立；– 下一步 $\bigcirc a$，指定a 在随后的时间步成立；– 始终 $\Box a$，规定a 永远不会为假；– 直到 $a \mathcal{U} b$，表示a 必须至少持续成立，直到b 变为真；– 释放 $a \mathcal{R} b$，指定b 持续成立，直到a 变为真（从这个意义上说，a 释放 b）。

如上一节所述，其中一些时序关系（例如具有时间延迟的动作的效果）甚至可以用标准逻辑形式体系表示。不过，这需要定义一个显式的时间变量t。此外，若没有特定的时序算子，也很难表达复杂的关系。例如，考虑插销转移任务，并假设环的质量较大，因此会对抓取工具施加较大的力。在这种情况下，一种合理的做法是缓慢提起一个环（例如，将其从插销上取下），同时持续监控末端执行器感测到的力。如果超过某个安全阈值，就必须中断取环操作，并采取恢复策略。使用标准布尔逻辑（例如答案集编程），可以将其表达为

以下一组陈述：

```
extract(A, R, C1, t) :- holding(A, R, C1, t), on(R, C1, P, C2, t)
extract(A, R, C1, t) :- extract(A, R, C1, t-1), not max_force(A, t),
                        on(R, C1, P, C2, t)
```

其中，`max_force(A, t)` 是一个流，当力传感器检测到工具处的高力时，该流便被基项化。上述两条语句可以用时序逻辑中的一条语句替代，并使用 $\mathcal{R}$ 算子：

$$\texttt{not on(R, C1, P, C2)}\ \mathcal{R}\ (\texttt{extract(A, R, C)} \leftarrow \texttt{holding(A, R, C), not max_force(A)})$$

本节中的逻辑编程框架主要依赖于 Pnueli（1977）提出的线性时序逻辑（LTL），假设事件构成线性序列，而非更复杂（且在计算上不便）的非线性时序逻辑（例如 Emerson 和 Clarke（1982）提出的 CTL）中的多种实现。具体而言，LTL 的解释是在有限轨迹（LTL$_f$）上定义的，即在有限时间范围内定义，这一做法可见于 Bienvenu 等人（2006）、Gabaldon（2004）和 Wilke（1999）等人的研究。这一假设对于机器人任务而言是合理的，因为这类任务通常涉及需要执行的有限动作序列。选择聚焦于 LTL 有两个理由。首先，Diekert 和 Gastin（2008）已证明，LTL$_f$ 的表达能力与一阶逻辑相同。LTL 的扩展形式 LDL$_f$（De Giacomo 和 Vardi，2013）包含来自命题动态逻辑的正则表达式（Fischer 和 Ladner，1979）；Trakhtenbrot（1962）已证明，对于有限自动机而言，其表达能力与单子二阶逻辑相同。LDL$_f$ 公式的综合问题（即为变量赋予真值，使给定公式成立）已被证明等价于完全可观测性下的条件规划问题，且为 2EXPTIME 完全问题（De Giacomo 和 Vardi，2015）。因此，可以将高级动作语言（例如 PDDL）中的通用任务规划问题编码为 LTL$_f$/LDL$_f$ 公式，再通过基于 SAT 检验的方法求解。为求简便，本节余下部分将省略下标 f，因此 LTL 均指 LTL$_f$。

机器人领域中一个用于线性时序逻辑的相关工具是 Finucane 等人（2010）开发的 LTLMoP，它可在 ROS 框架中实现时序任务规范。LTLMoP 已用于多目标导航（Kumar 和 Kala，2016）；使用 Aldebaran Nao 机器人在杂货店进行任务规划（Jing 等，2012）；多机器人协作任务规划（Kim 和 Kwon，2012、2014）；使用 Pi-oneer 3-DX 机器人在部分已知环境中执行搜救任务（Sarid 等，2013）；模块化机器人的任务规划（Jing 等，2016）；巡逻机器人的多任务规划（Liu 等，2017）；以及基于博弈论的反应式任务规划（Schmuck 等，2017）。此外，LTLMoP 还被应用于机器人学中不可综合计划的解释问题（Raman 和 Kress-Gazit，2012），这有助于提升机器人系统的可解释性。LTLMoP 通过 SAT 检验，根据时序规格求解任务规划问题。正如 Kress-Gazit 等人（2011）所指出的，这种方法的一个局限是状态爆炸问题：在规划问题中引入时间维度后，求解阶段需要评估不同的时间范围，从而导致这一问题。当在线性时序逻辑应用于真实机器人时，这一问题会更加突出；真实机器人通常采用混合控制（连接离散时间控制与连续时间控制），以使底层运动行为符合任务层面的规范。在这种场景下，基于线性时序逻辑的计划综合往往变得难以处理，因为无法分析所有未来场景。为应对计算问题，人们已提出多种解决方案。

例如，Lahijanian 等（2015）提出了可满足性量化这一概念，以确保满足主要的线性时序逻辑约束，同时忽略不那么重要的约束。例如，Tumova 等（2014）采用了这一方案，保证一台能够抓取、放下球并行走的 Nao 机器人在线性时序逻辑任务规范方面达到最大满足度，同时处理故障情况。

另一种广泛使用的方法是滚动时域（Wongpiromsarn 等，2010）。该方法基于这样的假设：环境的动态特性使得无法制定一个延伸到遥远未来的计划。因此，一个好的设计选择是，在较短的时间范围内验证线性时序逻辑公式，并在执行过程中根据当前状态随时细化所得计划。这样一来，只需分析最有可能出现的未来轨迹，状态爆炸也能得到显著缓解。与 LTLMoP 不同，实现滚动时域方法的工具是 TuLiP（Wongpiromsarn 等，2011a）。尽管该工具主要用于为集成任务级（离散）规划与运动规划（连续）问题构造可证明正确的混合控制器，但 Lin（2014）中的应用也值得一提：该研究通过 TuLiP 规划机器人监视任务，同时利用 μ-微积分（一种可用于表达时序公式的模态逻辑）（Bradfield 和 Stirling，2007）保证局部反应能力（如 Bloem 等（2012）所示）和细化；此外，Zhao 等（2016）还将其用于非结构化环境中的全身双足运动。

遗憾的是，上述两种解决方案并非总能应用于机器人学。事实上，在安全性至关重要的场景中，量化求解过程中需要考虑的约束相关性是危险的，因为这可能导致不希望出现的行为。至于限制最大时间范围，这对于插销转移这类简单任务是可行的，但无法妥善扩展到具有长期目标的复杂任务，例如探索广阔且可能未知的环境。

不过，也存在其他用于机器人任务规划的基于线性时序逻辑的范式，它们同样面临上述问题。这些范式主要应用于多机器人系统协同（Kloetzer 和 Belta 2006；Wong 和 Kress-Gazit 2016；Lacerda 和 Lima 2012），采用佩特里网任务表示；Guo 等（2014）研究了通信约束下的动态领导者选择；Guo 等（2016）研究了相互距离约束；Kress-Gazit 等（2009）面向搜救和覆盖场景；He 等（2015）针对操作任务；Schillinger 等（2017）研究了 ROS 框架下的时间最优性；Gundana 和 Kress-Gazit（2021）则将信号时序逻辑（Donz´e 和 Maler 2010）——一种为运动规划设计的时序逻辑——与标准线性时序逻辑相结合，以处理多机器人领域中的基于事件的任务规划。还应提及其他应用，包括 Xu 等（2016）对机器人消防场景中的动态规划进行研究；Colledanchise 等（2017）利用从线性时序逻辑规范构建行为树的方法，对对抗性环境中的移动机器人进行仿真任务规划；Bank 等（2018）研究自主柔性制造，并在 Siemens NX Mechatronics Concept Designer 仿真器中对龙门机器人进行了实验评估；Hao 等（2017）研究仓储机器人技术；Guo 等（2013）研究机器人导航，并通过混合控制与基于低层模型检验的运动规划相结合；Camacho 等（2017）将由 PDDL 编码导出的有限与无限轨迹线性时序逻辑规范转换为非确定自动机来求解；Choi 等（2021）则使用线性时序逻辑设计了一种承担患者接诊与分诊工作的医疗机器人。

7.1 讨论

时序逻辑程序能够以优雅的形式化方式表示复杂的时序任务规范，无需使用专门用于时序索引的变量。这对于涉及较长规划时域的应用尤为有用；当必须保证机器人系统及其环境的形式化属性时（例如，在集成任务与运动规划中确保安全状态的稳定性或可达性），这一点也十分重要。许多采用线性时序逻辑语法实现的 PDDL 时间扩展应用，充分证明了对较强时序表达能力的需求，例如工业自动化（Kootbally 2016）或多机器人任务规划（Wurm 等 2010、2013；Coles 等 2019；Carreno 等 2020）。

时序逻辑编程的一个主要局限在于，大多数形式化方法都具有非单调性（少数例外不适用于机器人学，例如 Baral 和 Zhao（2007）提出的方法），这妨碍了动态环境条件下的在线可重配置性和计划修订。此外，表达能力的增强也带来了额外的计算开销，这是由时序规划可判定性问题的固有复杂度所导致的（Rintanen 等，2007）。因此，存在若干标准逻辑编程范式的时间扩展，例如类似 Prolog 的扩展（Abadi 和 Manna，1989），以及 Cabalar 等（2019a）提出的较新型、基于答案集编程的 *telingo*，该工具已在第 6.4 节中提及。

不过，近年来人们投入了大量精力来实现高效的 LTL可满足性检查，例如 Dureja 和 Rozier（2018）以及 Geatti 等（2021）的工作，以弥合时序逻辑编程与实际机器人应用案例之间的差距。本节介绍的主要范式，即 TuLiP（Wongpiromsarn 等，2011b）和 LTLMoP（Finucane 等，2010），也通过 Python API 等方式提供实现支持；后者已经集成到 ROS 框架中。

8 概率逻辑编程

前几节介绍的逻辑编程解决方案能够描述具有复杂约束的任务规划问题，其中涉及时间和因果关系。然而，对于与未知环境交互的审慎型机器人而言，一个重要挑战是感知的质量。这要求实时规划系统不仅要利用传感器提供的实时信息来更新环境模型，还要考虑所感知模型的不确定性，并生成一系列最有可能的动作，以便在获得新证据后进一步完善。在本节中，将分析一些逻辑编程语言；这些语言通过引入概率论概念来丰富标准逻辑和时序逻辑语义，从而表达具有一定不确定性的规范。概率逻辑定义可见于 Ng 和 Subrahmanian（1992），Sato（1995）中的分布语义（DS），以及相关扩展（Lukasiewicz 1998，2001）。

概率逻辑程序的语法包括概率注释原子和规则。给定$\mu \subseteq [0,1]$，概率注释布尔变量 $a:\ \mu$ 是一个布尔变量，其为真的概率在 μ中。同样，在逻辑编程形式体系中，可以定义概率原子，并以有界概率将其实例化为基项。概率语句则是包含概率注释原子的语句，或带有自身注释的语句（即（$a \leftarrow b$）：μ）。例如，在插销转移领域中，环的位置可能只能以 50% 到 80% 的准确度确定，具体取决于 RGBD相机生成的含噪点云。这会影响所有与环的位置相关的原子的概率，例如，`reachable(A, R, C, t)` 会变成概率原子 `reachable(A, R, C, t)`:$[0.5, 0.8]$。因此，所有包含

可达性原子的语句都会变成概率语句。例如，移动到环所在位置的前置条件为：

$$\texttt{move(A, R, C, t)} \leftarrow \texttt{reachable(A, R, C, t)} : [0.5, 0.8]$$

此外，`move(A, R, t)` 也会成为一个概率注释原子，并继承概率原子前置条件的相同概率范围。如果某个原子有多个成立的前置条件，则这些前置条件的注释共同决定其概率范围。

因此，概率逻辑程序的求解过程会遭遇与时序逻辑类似的状态爆炸问题。事实上，除非概率范围缩小为 $\{0\}$，否则无法将原子基化为真或假（假设采用封闭世界假设）。因此，根据规范对原子进行基化时，需要合并体原子的不同概率范围，再将其传播至头部原子，并评估所有可能的场景。

已有若干初步方案被提出，以提高概率逻辑求解的效率。例如，可以将逻辑程序的求解限制为仅推导紧逻辑后果。考虑一组概率命题逻辑子句（由概率语句基化得到）$\mathcal{P}$，若子句 $\texttt{c}:\mu_c$ 满足 $sup\mu_c = max\mu, inf\mu_c = min\mu$，则称其为 $\mathcal{P}$ 的**紧逻辑后果**，其中 $\mu \subseteq [0, 1]$ 是这样一组实数：对于 $\mathcal{P}$ 中的子句，总存在一种实现。换言之，如果一个子句可以从一组子句的所有可能实现中推导出来，那么它就是该子句集的紧逻辑后果。以插销转移领域为例。根据规范，我们知道，可以通过直接将环移动到插销上，或先将其转移到另一只机械臂上，再将其放置到插销上；具体取决于可达性条件。假设环境流值因传感器不确定性而属于概率原子，我们可以用以下概率子句对这些知识进行编码：

$$\begin{aligned}
&(\texttt{on(ring, red, peg, red, 2)} \leftarrow \texttt{move(psm1, peg, red, 1)}) : [0.7, 1] \\
&(\texttt{on(ring, red, peg, red, 3)} \leftarrow \texttt{move(psm1, peg, red, 2)}, \\
&\qquad\qquad\qquad\qquad\qquad \texttt{transfer(psm2, ring, red, 1)}) : [0.8, 1]
\end{aligned}$$

并且 $\texttt{on(ring, red, peg, red,)}$：$[0.7, 1]$ 是紧逻辑后果。

Lukasiewicz（1998）指出，在不涉及否定的情况下，紧逻辑后果的推导可以归结为求解两个线性规划。然而，研究表明，一般情况下该问题属于 NP，具体取决于可能实现的数量。因此，当概率子句可与确定性子句分离时，作者还提出了一种算法，以进一步降低该问题的复杂度。

另一种解决方案是对随机系统使用近似采样方法，例如蒙特卡洛。这种方法用于 Problog（De Raedt 等，2007年）等系统。Problog 是一种流行的 Prolog 概率扩展，允许定义一种权重，为原子和规范赋予离散概率。这样，概率原子和概率语句被赋予离散概率，而非实数范围，并且基化原子的传播效率更高（请注意，Helmert（2002）已证明，对连续变量进行推理通常是不可判定的）。Problog 及其变体已应用于多个机器人任务规划场景，例如开门（Moldovan 等，2013年）和物体操作（Moldovan 等，2012年；Tan 等，2019年；Antanas 等，2019年）。Problog 的另一种替代方法是 Gutmann 等人（2011年）提出的分布式子句（DC）框架，它扩展了 Problog，允许在规则和原子上定义连续概率分布。Moldovan 和 De Raedt（2014年）、Nitti 等人（2015年）以及 Moldovan 等人（2018年）将 DC 用于机器人操作和抓取。

上述方法仍然是近似方法，或者只能应用于有限类别的逻辑程序，因此不适用于大多数实时机器人领域。尽管如此，概率逻辑编程有望弥合经典逻辑确定性与审慎型机器人在规划—行动—感知循环中固有的不确定性之间的鸿沟。

近期研究实际上着重于在开放世界领域中，将基于查询的规划器与概率知识相结合；随着获取新证据，这些知识可以随之更新（类似于标准逻辑中 CRAM框架所采用的方法）。Jain等人（2009年）提出了一个相关实例：在包含一台 B21服务机器人的家庭场景中，ProbCog系统可以对由多实体贝叶斯网络的概率逻辑（MEBN）（Laskey 2008年）导出的贝叶斯逻辑网络（BLN）进行查询和推理。基本上，多实体贝叶斯网络是一种知识表示形式体系，它通过一阶逻辑语句连接贝叶斯网络片段。这一框架已被用于多种场景，例如，Beetz等人（2012年）将其用于机器人处理家务时的复杂任务分解与规划；Beetz等人（2010a年）和 Nyga和Beetz（2012年）将其用于操作；Maurelli等人（2014年）则将其用于认知型水下航行器。

Nyga等人（2018年）针对家庭任务提出了一种类似的方法，使用马尔可夫逻辑网络（MLN）（Richardson和Domingos 2006），而非贝叶斯逻辑网络。马尔可夫逻辑网络基于概率加权逻辑语句。Al-Moadhen等人（2016年）通过使用基于 STRIPS 的 Metric-FF 规划器（Hoffmann和Nebel 2001）查询马尔可夫逻辑网络，实现了操作与导航场景中的任务规划；Lisca等人（2015年）则提出了一种用于化学实验自动化的机器人，通过查询基于马尔可夫逻辑网络的概率动作核心（PRACs）（Nyga和Beetz 2012年）来完成任务。此外，Lee和Wang（2016年、2018年）提出的 LP^{MLN} 也值得一提：它在用于迁移系统的动作语言 $\mathcal{BC}+$（Babb和Lee 2015年）下，采用答案集语义对马尔可夫逻辑网络进行推理，不过其在机器人场景中的实际应用仍在研究之中。

还应强调，研究正致力于将概率论扩展到答案集语义中。主要例子是 P-log（Baral 等人，2004年），它与 Problog 一样基于加权规范，并被用于例如机器人序列规划的 LCORPP框架（Amiri等人，2019年）。这样一来，不久的将来或许就能在概率逻辑领域利用 SAT求解策略的最新进展。

在其他应用于机器人场景的概率逻辑编程实现中，我们还提到 PREGO（Belle 和 Levesque，2014年）；Rutten 等人（2004年）提出的概率计算树逻辑（PCTL）是一种概率非线性时序逻辑，可用于定义任务规范，例如制造业中的人机交互（Zhang 等人，2016年）和机器人群体协调（Brambilla 等人，2012年）；Gurzoni 等人（2014年）则将基于马尔可夫逻辑网络的事件演算应用于搜索与救援场景。

8.1 讨论

基于概率逻辑编程的规划器能够优雅地将现实机器人场景中的不确定性建模到任务规范中：既可以像 De Raedt 等人（2007）的 Problog 那样，为规则和原子扩展离散概率值，也可以像 DC（Gutmann 等人，2011）那样扩展为连续分布。这使其能够丰富纯概率马尔可夫网络或贝叶斯网络的表达能力，并增强可解释性。其主要代表性工具 Problog 是 Prolog 的扩展，因此具备 Prolog 的大部分优点，包括 Python /

Java API、高级构造，以及可能与外部知识库集成以进行开放世界推理（如 CRAM，Beetz 等人，2010b）。同样，研究者也提出了时序逻辑编程的概率扩展，例如 Rutten 等人（2004）提出的 PCTL，以及答案集语义（Baral 等人，2004）。

尽管近期文献中已有多种概率逻辑编程范式在机器人领域的应用，但当前表示与推理工具的主要局限在于：在不确定性下求解规划问题时，问题复杂性会导致计算效率低下（Helmert 2002）；当模型采用连续（更为现实）的概率分布时，这一问题尤为突出。因此，目前结合形式逻辑与概率论优势的最高效方案，是将标准逻辑（例如，Sridharan 等人（2019）以及 Leonetti 等人（2016）提出的用于自主导航与探索的 ASP）或时序逻辑编程（Leonetti 等人，2012）与马尔可夫模型进行分层组合，以约束概率推理，并生成效率更高、可解释性更强的鲁棒计划。

9 结论

目前，标准逻辑编程框架是基于逻辑的方法中最突出的审慎型机器人任务规划问题解决方案，因为它们能够满足大多数实际应用场景对表达能力的要求，提供足够的软件支持以集成到机器人应用中，并实现高效的求解算法。基于 Prolog 的框架历史最悠久、应用最广泛，而基于 ASP 的框架正迅速成为有力的竞争者。在答案集语义方面，最先进的解决方案是 Clingo，它通过偏好推理支持最优规划生成。一般而言，基于 Prolog 的工具旨在响应外部查询，因此最适合在大型知识库上进行推理（例如，在 CRAM 项目中），或用于紧密的人机交互。相反，在复杂场景中进行高级策略搜索时，应优先采用基于 ASP 的规划器。

时序逻辑为集成任务与运动规划（例如使用 TuLiP）提供了一种实用的形式化方法，因此有望推动下一代审慎型机器人的发展。类似地，概率逻辑有望弥合规划与感知之间的差距。遗憾的是，这两种形式化方法在现实场景中都会因状态爆炸问题而面临计算效率低下的困境。因此，机器人领域尚未出现通用框架，定制的特定应用实现或其他框架（例如贝叶斯框架）仍代表着研究现状。

我们认为，未来的研究应推动不同框架的整合，以找到效率与表达能力之间的最佳权衡。例如，Cabalar 等人（2019a）提出了一种采用时序答案集编程的近期框架，即 *telingo*；而 Sridharan 等人（2019）则提出了逻辑规划与马尔可夫决策过程的层次化组合。

10 声明

10.1 资金支持

本研究部分由欧盟资助，资助项目为 ERC-ADG N. 742671、ARS（自主机器人手术）项目。网址：https://www.ars-project.eu/?lang=it

10.2 利益冲突

作者声明，他们与任何在本稿件所讨论的主题或材料方面具有经济或非经济利益的组织或实体均无关联，也未参与其中。

10.3 数据和材料的可用性

不适用。

10.4 代码可用性

不适用。

参考文献

Abadi M, Manna Z（1989）时序逻辑编程。《符号计算杂志》8(3):277–295 Abels D, Jordi J, Ostrowski M, Schaub T, Toletti A, Wanko P（2019）基于混合 ASP 的列车调度。载于：逻辑程序设计与非单调推理国际会议，Springer，第 3–17 页 Adrian WT, Alviano M, Calimeri F, Cuteri B, Dodaro C, Faber W, Fusc`a D, Leone N, Manna M, Perri S, 等（2018）ASP 系统 DLV：进展与应用。《KI-K¨unstliche Intelligenz》32(2):177–179 Aeronautiques C, Howe A, Knoblock C, McDermott ID, Ram A, Veloso M, Weld D, SRI DW, Barrett A, Christianson D, 等（1998）PDDL——规划领域定义语言。技术报告，Tech Rep Aguado F, Cabalar P, Di´eguez M, P´erez G, Vidal C（2013）时序均衡逻辑：综述。《应用非经典逻辑杂志》23(1-2):2–24 Al-Moadhen AA, Packianather M, Setchi R, Qiu R（2016）利用语义知识库在确定性和概率条件下进行机器人任务规划。《国际知识与系统科学杂志》7(1):56–77 Alami R, Clodic A, Montreuil V, Sisbot EA, Chatila R（2005）面向人机交互的任务规划。载于：2005 年智能物体与环境智能联合会议论文集：创新的情境感知服务、用途与技术，第 81–85 页 Alferes JJ, Leite JA, Pereira LM, Quaresma P（2000）作为溯因更新的规划。载于：人工智能规划与智能体研讨会论文集，第 1–8 页 Amiri S, Shirazi MS, Zhang S（2019）不确定性下机器人序贯决策的学习与推理。arXiv 预印本 arXiv:190105322 Andres B, Rajaratnam D, Sabuncu O, Schaub T（2015）将 ASP 集成到 ROS 中以实现机器人推理。载于：逻辑程序设计与非单调推理国际会议，Springer，第 69–82 页 Antanas L, Moreno P, Neumann M, de Figueiredo RP, Kersting K, Santos-Victor J, De Raedt L（2019）用于机器人抓取的语义与几何推理：一种概率逻辑方法。《自主机器人》43(6):1393–1418 Apt KR, Doets K（1994）SLDNF 归结的新定义。《逻辑程序设计杂志》18(2):177–190

Armando A, Castellini C, Giunchiglia E（1999）用于时序推理的基于 SAT 的过程。载于：欧洲规划会议，Springer，第 97–108 页 Baader F, Calvanese D, McGuinness D, Patel-Schneider P, Nardi D（2003a）《描述逻辑手册：理论、实现与应用》。剑桥大学出版社 Baader F, Calvanese D, McGuinness D, Patel-Schneider P, Nardi D, 等（2003b）《描述逻辑手册：理论、实现与应用》。剑桥大学出版社 Babb J, Lee J（2015）动作语言 BC+。《逻辑与计算杂志》 B¨ackstr¨om C, Klein I（1991）多项式时间规划：SAS-PUBS 类。《计算智能》7(3):181–197 Bank HS, D'Souza S, Rasam A（2018）基于时序逻辑（TL）的智能制造系统自主性。载于：Procedia Manufacturing，第 26 卷，第 1221–1229 页 Baral C, Son TC（2015）"再来一摞同样高度的蓝色物品！"：基于 ASP 的规划与规划失败分析。载于：逻辑程序设计与非单调推理国际会议，Springer，第 127–133 页 Baral C, Zhao J（2007）用于目标规约的非单调时间逻辑。载于：国际人工智能联合会议，第 7 卷，第 236–242 页 Baral C, Gelfond M, Rushton N（2004）基于答案集的概率推理。载于：逻辑程序设计与非单调推理国际会议，Springer，第 21–33 页 Beetz M, Jain D, M¨osenlechner L, Tenorth M（2010a）迈向日常操作活动的执行。《机器人学与自主系统》58(9):1085–1095 Beetz M, M¨osenlechner L, Tenorth M（2010b）CRAM——用于人类环境中日常操作的认知机器人抽象机。载于：2010年IEEE/RSJ智能机器人与系统国际会议，电气电子工程师学会，第 1012–1017 页 Beetz M, Jain D, Mosenlechner L, Tenorth M, Kunze L, Blodow N, Pangercic D（2012）认知增强型自主机器人控制，实现家庭杂务任务智能。《IEEE 会议录》100(8):2454–2471 Belle V, Levesque H（2014）PREGO：用于连续领域中基于信念的认知机器人学的动作语言。载于：第二十八届 AAAI 人工智能会议 Bertolucci R, Capitanelli A, Dodaro C, Leone N, Maratea M, Mastrogiovanni F, Vallati M（2021）利用答案集编程，通过双臂机器人操作铰接物体。《逻辑编程理论与实践》21(3):372–401 Beßler D, Pomarlan M, Beetz M（2018）面向机器人智能体的 OWL 支持的装配规划。载于：自主智能体与多智能体系统国际联合会议论文集，AAMAS，第 3 卷，第 1684–1692 页 Bienvenu M, Fritz C, McIlraith SA（2006）具有定性时间偏好的规划。知识表示与推理原理国际会议 6:134–144 Biere A, Heule M, van Maaren H（2009）《可满足性手册》，第 185 卷。IOS Press Bloem R, Jobstmann B, Piterman N, Pnueli A, Sa'ar Y（2012）反应式（1）设计的综合。《计算机与系统科学杂志》78(3):911–938 Bogue R（2011）核工业中的机器人：技术与应用综述。《Industrial Robot: An International Journal》Bogue R（2018）建筑业中机器人的前景如何？《Industrial Robot: An International Journal》Bolisani E, Bratianu C（2017）知识战略规划：管理不确定性、动荡与动态的综合方法。《知识管理杂志》 Bradfield J, Stirling C（2007）模态 μ 演算。《模态逻辑手册》3:721–756 Brambilla M, Pinciroli C, Birattari M, Dorigo M（2012）群体机器人学的性质驱动设计。载于：2012 年第 11 届自主智能体与多智能体系统国际会议，AAMAS：创新应用专题，第 2 卷，第 664–671 页 Bratman M（1987）《意图、计划与实践理性》。剑桥：马萨诸塞州剑桥：哈佛大学出版社 Bratman ME, 等（1988）计划与资源受限实践推理。《计算智能》4(3):349–355 Brewka G, Niemela I, Truszczynski M（2008）偏好与非单调推理。《AI 杂志》29(4):69–69 Bruno B, Chong NY, Kamide H, Kanoria S, Lee J, Lim Y, Pandey AK, Papadopoulos C, Papadopoulos I, Pecora F, 等（2017）CARESSES 欧日项目：打造具有文化适应能力的辅助机器人。载于：意大利环境辅助生活论坛，Springer，第 151–169 页 Bylander T（1994）命题 STRIPS 规划的计算复杂性。《人工智能》69(1-2):165–204

Cabalar P, Kaminski R, Morkisch P, Schaub T (2019a) telingo= asp+ time. 载于：逻辑程序设计与非单调推理国际会议，Springer，第256–269页。Cabalar P, Rey M, Vidal C (2019b) 时序答案集程序设计的完备规划器。载于：EPIA 人工智能会议，Springer，第520–525页。Calimeri F, Faber W, Gebser M, Ianni G, Kaminski R, Krennwallner T, Leone N, Maratea M, Ricca F, Schaub T (2020) ASP-Core-2 输入语言格式。《逻辑编程理论与实践》20(2):294–309。Camacho A, Triantafillou E, Muise C, Baier JA, McIlraith SA (2017) 带时态扩展目标的非确定性规划：有限与无限轨迹上的线性时序逻辑。载于：第31届 AAAI 人工智能会议，AAAI 2017，第3716–3724页。Campbell M, Egerstedt M, How JP, Murray RM (2010) 城市环境中的自动驾驶：方法、经验与挑战。《英国皇家学会哲学汇刊 A 辑：数学、物理与工程科学》368(1928):4649–4672。Carreno Y, Pairet `E, Petillot Y, Petrick RP (2020) 通过目标分配与时序规划实现异构自主水下航行器任务的去中心化策略。载于：国际自动规划与调度会议论文集，第30卷，第431–439页。Cashmore M, Fox M, Long D, Magazzeni D, Ridder B, Carrera A, Palomeras N, Hurtos N, Carreras M (2015) ROSPlan：在机器人操作系统中进行规划。载于：第二十五届国际自动规划与调度会议。Chandrasekaran B, Conrad JM (2015) 人机协作：综述。载于：SoutheastCon 2015，电气电子工程师学会，第1–8页。Chen X, Xie J, Ji J, Sui Z (2013) 推进开放知识赋能人机交互。《人机交互期刊》1(2):100–117。Choi B, Park J, Park C (2021) 医疗环境中人机交互的形式化验证。载于：ACM/IEEE 人机交互国际会议，第181–185页。Coles AJ, Coles AI, Munoz MM, Savas OE, Keller T, Pommerening F, Helmert M (2019) 使用时序 PDDL 为机器人太空任务进行机载规划。载于：第11届太空规划与调度国际研讨会。Colledanchise M, Murray RM, Ogren P (2017) 正确性构造行为树的综合。载于：IEEE 智能机器人与系统国际会议，第2017年9月卷，第6039–6046页。Colmerauer A (1990) Prolog III 导论。载于：计算逻辑，Springer，第37–79页。Crespo J, Barber R, Mozos OM, Beßler D, Beetz M (2018) 移动机器人语义导航的推理系统。载于：IEEE 智能机器人与系统国际会议，第5654–5659页。Cui G, Shuai W, Chen X (2021) 开放世界中服务机器人的语义任务规划。《未来互联网》13(2):1–19，网址：www.scopus.com。De Giacomo G, Vardi M (2015) 有限轨迹上的线性时序逻辑与线性动态逻辑综合。载于：第二十四届人工智能国际联合会议。De Giacomo G, Vardi MY (2013) 有限轨迹上的线性时序逻辑与线性动态逻辑。载于：第二十三届人工智能国际联合会议。De Raedt L, Kimmig A, Toivonen H (2007) Problog：概率 Prolog 及其在链接发现中的应用。载于：人工智能国际联合会议，海得拉巴，第7卷，第2462–2467页。Diab M, Akbari A, Ud Din M, Rosell J (2019) PMK——面向自主机器人感知与操作的知识处理框架。《传感器》19(5):1166。Diekert V, Gastin P (2008) 一阶可定义语言。《逻辑与自动机》2:261–306。DiMaio S, Hanuschik M, Kreaden U (2011) 达芬奇手术系统。载于：《手术机器人学》，Springer，第199–217页。Dimopoulos Y, Nebel B, Koehler J (1997) 在非单调逻辑程序中编码规划问题。载于：欧洲规划会议，Springer，第169–181页。Dimopoulos Y, Gebser M, L¨uhne P, Romero J, Schaub T (2019) plasp 3：迈向高效的答案集程序规划。《逻辑编程理论与实践》19(3):477–504。Dix J, Kuter U, Nau D (2003) 使用有序任务分解进行答案集程序规划。载于：人工智能年会，Springer，第490–504页。Dodaro C, Galat`a G, Grioni A, Maratea M, Mochi M, Porro I (2021) 基于答案集编程的化疗治疗调度问题解决方案。《逻辑编程理论与实践》21(6):835–851。Dodaro C, Galat`a G, Khan MK, Maratea M, Porro I (2022) 通过床位管理进行手术室（重新）排程。载于：《逻辑编程理论与实践》22(2):229–253。Donz´e A, Maler O (2010) 实值信号上时序逻辑的鲁棒满足性。载于：定时系统形式化建模与分析国际会议，Springer，第92–106页。Dornhege C, Eyerich P, Keller T, Tr¨ug S, Brenner M, Nebel B (2009) 面向领域无关规划系统的语义附加。载于：第十九届国际自动规划与调度会议。

Drabent W (1996) 非悬置查询的 SLDNF 归结完备性。《逻辑程序设计杂志》27(2):89–106。Dureja R, Rozier KY (2018) 通过发现设计空间依赖提升线性时态逻辑模型检测的可扩展性（d^3）。载于：系统构造与分析工具和算法国际会议，Springer，第309–327页。Eiter T, Leone N, Mateis C, Pfeifer G, Scarcello F (1997) 非单调推理的演绎系统。载于：逻辑程序设计与非单调推理国际会议，Springer，第363–374页。Eiter T, Faber W, Leone N, Pfeifer G (2000) 使用 DLV 系统进行声明式问题求解。载于：基于逻辑的人工智能，Springer，第79–103页。Eiter T, Germano S, Ianni G, Kaminski T, Redl C, Sch¨uller P, Weinzierl A (2018) DLVHEX 系统。《KI-K¨unstliche Intelligenz》32(2-3):187–189。Emerson EA, Clarke EM (1982) 使用分支时态逻辑综合同步框架。《计算机编程科学》2(3):241–266。Erdem E, Patoglu V (2012) 动作语言在认知机器人学中的应用。载于：《正确推理》，Springer，第229–246页。Erdem E, Patoglu V (2018) 答案集编程在机器人学中的应用。《KI-K¨unstliche Intelligenz》32(2-3):143–149。Erdem E, Gelfond M, Leone N (2016) 答案集编程的应用。《AI 杂志》37(3):53–68。Erol K (1996) 分层任务网络规划：形式化、分析与实现。博士论文。Escudero-Rodrigo D, Alquezar R (2016) 基于 ROSPlan 的通用型机器人锚定问题研究，《人工智能与应用前沿》，第288卷。Fakhruldeen H, Maheshwari P, Lenz A, Dailami F, Pipe AG (2016) 利用嵌入对象中的计划进行人机协作的规划器。IFAC-PapersOnLine 49(21):668–674，网址：www.scopus.com，引用次数：1。Falkner A, Friedrich G, Schekotihin K, Taupe R, Teppan EC (2018) 答案集编程的工业应用。《KI-K¨unstliche Intelligenz》32(2-3):165–176。Farinelli A, Finzi A, Lukasiewicz T (2007) 部分可观测条件下基于 Golog 的团队编程。载于：人工智能国际联合会议，第2097–2102页。Fiazza MC, Fiorini P (2021) 面向可解释性的设计：应对手术机器人的认证挑战。载于：2021年IEEE机器人智能与安全国际会议，电气电子工程师学会，第264–267页。Fikes RE, Nilsson NJ (1971) STRIPS：将定理证明应用于问题求解的新方法。《人工智能》2(3-4):189–208。Finucane C, Jing G, Kress-Gazit H (2010) LTLMoP：探索语言、时序逻辑与机器人控制。载于：2010年IEEE/RSJ智能机器人与系统国际会议，电气电子工程师学会，第1988–1993页。Fischer MJ, Ladner RE (1979) 正则程序的命题动态逻辑。《计算机与系统科学杂志》18(2):194–211。Fox M, Long D (2003) PDDL2.1：用于表达时态规划领域的 PDDL 扩展。《人工智能研究杂志》20:61–124。Fox M, Long D, Magazzeni D (2017) 可解释规划。载于：人工智能国际联合会议第17届可解释规划研讨会论文集。Frank J, J´onsson A (2003) 基于约束的属性与区间规划。《Constraints》8(4):339–364。Fratini S, Cesta A, De Benedictis R, Orlandini A, Rasconi R (2011) 面向目标的自主控制器中的基于APSI的审议。ASTRA 11。Fr¨uhwirth T (1998) 约束处理规则的理论与实践。《逻辑程序设计杂志》37(1-3):95–138。Gabaldon A (2004) 前置条件控制与推进算法。载于：国际自动规划与调度会议，第23–32页。Geatti L, Gigante N, Montanari A (2021) Black：快速、灵活且可靠的线性时态逻辑可满足性检查器。载于：第3届人工智能与形式化验证、逻辑、自动机和综合研讨会论文集，CEUR-WS，第2987卷，第7–12页。Gebser M, Obermeier P, Otto T, Schaub T, Sabuncu O, Nguyen V, Son TC (2018a) 在机器人内部物流领域中的实验。《逻辑编程理论与实践》18(3-4):502–519。Gebser M, Obermeier P, Schaub T, Ratsch-Heitmann M, Runge M (2018b) 使用答案集编程为汽车装配中的无人驾驶运输车辆规划路径。《逻辑编程理论与实践》18(3-4):520–534。Gebser M, Kaminski R, Kaufmann B, Schaub T (2019) 使用 clingo 进行多轮答案集编程求解。《逻辑编程理论与实践》19(1):27–82。

Geffner H, Bonet B（2013）《自动规划模型与方法简明导论》。人工智能与机器学习综合讲座 8(1):1–141。Gelfond M, Lifschitz V（1988）逻辑编程的稳定模型语义。载于：逻辑编程国际会议，第 88 卷，页 1070–1080。Ghallab M, Nau D, Traverso P（2004）《自动规划：理论与实践》。爱思唯尔。Ghallab M, Nau D, Traverso P（2016）《自动规划与行动》。剑桥大学出版社。Gibaud B, Forestier G, Feldmann C, Ferrigno G, Gonc¸alves P, Haidegger T, Julliard C, Kati´c D, Kenngott H, Maier-Hein L 等（2018）迈向手术流程模型的标准本体。《国际计算机辅助放射学与外科学杂志》13(9):1397–1408。Gierse G, Niemueller T, Claßen J, Lakemeyer G（2016）GOLOG 中可中断并可恢复执行的任务，《人工智能与应用前沿》，第 285 卷。Ginesi M, Meli D, Nakawala HC, Roberti A, Fiorini P（2019）用于手术任务自动化的知识框架。载于：2019年第19届国际先进机器人会议，页 37–42，DOI 10.1109/ICAR46387.2019.8981619。Ginesi M, Meli D, Roberti A, Sansonetto N, Fiorini P（2020）机器人辅助手术中的自主任务规划与情境感知。载于：IEEE/RSJ国际智能机器人与系统会议，电气电子工程师学会，页 3144–3150。Goebel R, Chander A, Holzinger K, Lecue F, Akata Z, Stumpf S, Kieseberg P, Holzinger A（2018）可解释人工智能：新的 42？载于：国际跨领域机器学习与知识提取会议，Springer，页 295–303。Gonz´alez D, P´erez J, Milan´es V, Nashashibi F（2015）自动驾驶车辆运动规划技术综述。《IEEE智能交通系统汇刊》17(4):1135–1145。Gonc¸alves PJS, Torres PMB（2015）应用于机器人骨科手术的知识表示。《机器人与计算机集成制造》33:90–99。Goodrich MA, Schultz AC（2008）人机交互：综述。Now出版社。Gragera A, Garc´ıa AM, Fern´andez F（2019）用于社会自主机器人用例设计的建模与形式化工具。载于：伊比利亚机器人会议，Springer，页 656–667。Gundana D, Kress-Gazit H（2021）面向单机器人与多机器人任务的基于事件的信号时序逻辑综合。《IEEE机器人与自动化快报》6(2):3687–3694。Guo M, Johansson KH, Dimarogonas DV（2013）使用导航函数和动作描述语言，在 LTL 规范下进行运动与动作规划。载于：IEEE 智能机器人与系统国际会议，页 240–245。Guo M, Tumova J, Dimarogonas DV（2014）在局部 LTL 任务与连通性约束下的协作式分布式多智能体控制。载于：IEEE决策与控制会议论文集，第 2015-February 卷，页 75–80。Guo M, Tumova J, Dimarogonas DV（2016）在局部时序任务与相对距离约束下的无通信多智能体控制。《IEEE自动控制汇刊》61(12):3948–3962。Gurzoni J J A, Cozman FG, Martins MF, Santos PE（2014）用于机器人搜救场景中事件识别的逻辑概率模型。载于：IEEE国际系统、人类与控制论会议论文集，第 2014-January 卷，页 1726–1731。Gutmann B, Thon I, Kimmig A, Bruynooghe M, De Raedt L（2011）概率编程中的逻辑推理之妙。《逻辑编程理论与实践》11(4-5):663–680。Hao S, Huang Z, Wang L, Zhang R, Zhang X, Peng J, Yu W（2017）一种基于线性时序逻辑的仓库机器人多任务最优任务决策方法。载于：2017年IEEE国际系统、人类与控制论会议（SMC 2017），第 2017-January 卷，页 1453–1458。Harman H, Simoens P（2020）智能环境中用于主动式机器人辅助的动作图。《环境智能与智能环境杂志》12(2):79–99。Harmo P, Taipalus T, Knuuttila J, Vallet J, Halme A（2005）需求与解决方案——面向老年人与残障人士的家庭自动化与服务机器人。载于：2005年IEEE/RSJ国际智能机器人与系统会议，电气电子工程师学会，页 3201–3206。He K, Lahijanian M, Kavraki LE, Vardi MY（2015）迈向具有时序逻辑规范的操作规划。载于：IEEE国际机器人与自动化会议论文集，第 2015-June 卷，页 346–352。Helmert M（2002）含数值状态变量的规划问题的可判定性与不可判定性结果。载于：AIPS，页 44–53。Helmert M, R¨oger G, Karpas E（2011）Fast Downward Stone Soup：构建规划器组合的基线。载于：2011年国际自动规划与调度会议规划与学习研讨会，页 28–35。Hoffmann J, Nebel B（2001）FF规划系统：通过启发式搜索快速生成计划。《人工智能研究杂志》14:253–302

Ingrand F, Ghallab M（2017）自主机器人的审议：综述。《人工智能》247:10–44。Jain D, Mosenlechner L, Beetz M（2009）为机器人控制程序赋予一阶概率推理能力。载于：2009年IEEE国际机器人与自动化会议，电气电子工程师学会，页 3626–3631。Janssen R, van de Molengraft R, Bruyninckx H, Steinbuch M（2016）面向人类领域多机器人操作的云端集中式任务控制。《智能服务机器人》9(1):63–77。Javia B, Cimiano P（2017）协作式自主机器人技术的逻辑编程方法，《计算机科学讲义》（含《人工智能讲义》和《生物信息学讲义》子系列），第 10505 卷 LNAI。Ji J（2009）服务机器人的认知架构：一种答案集编程方法。载于：第25届逻辑编程国际会议论文集，第 5649 卷，页 532–533。Jiang Y, Zhang S, Khandelwal P, Stone P（2019）机器人任务规划：基于 PDDL 和答案集编程的系统的实证比较。《信息技术与电子工程前沿》20(3):363–373。Jim´enez S, De La Rosa T, Fern´andez S, Fern´andez F, Borrajo D（2012）自动规划中的机器学习综述。《知识工程评论》27(4):433–467。Jing G, Finucane C, Raman V, Kress-Gazit H（2012）根据结构化英语实现正确的高级机器人控制。载于：2012年IEEE国际机器人与自动化会议，电气电子工程师学会，页 3543–3544。Jing G, Tosun T, Yim M, Kress-Gazit H（2016）一个利用模块化机器人完成任务的端到端系统。载于：机器人学：科学与系统，页 1–5。Kadonoff MB（1987）用于移动机器人导航的基于 Prolog 的世界模型。载于：移动机器人 II，国际光学与光子学会，第 852 卷，页 305–310。Kamil F, Tang S, Khaksar W, Zulkifli N, Ahmad S（2015）动态环境中运动规划与避障方法综述。《机器人与自动化进展》4(2):134–142。Karpas E, Magazzeni D（2020）面向机器人的自动规划。《控制、机器人与自主系统年度评论》3:417–439。Kaufmann B, Leone N, Perri S, Schaub T（2016）答案集编程中的基化与求解。《AI 杂志》37(3):25–32。Kim S, Kwon G（2012）协作式多机器人的仿真。载于：混合信息技术国际会议，Springer，页 588–593。Kim S, Kwon G（2014）使用线性时序逻辑综合的多机器人协作仿真。《Information》17(5):1763–1769。Kirsch M, Matar´e V, Ferrein A, Schiffer S（2020）将 Golog++ 与 ROS 集成，以实现实用且可移植的高级控制。载于：智能体与人工智能国际会议（ICAART 2020）论文集，第 12 届会议，第 2 卷，页 692–699。Kloetzer M, Belta C（2006）面向机器人群体的 LTL 规划。载于：2006年IEEE网络、感知与控制国际会议论文集（ICNSC'06），页 578–583。Kolobov A（2012）马尔可夫决策过程规划：人工智能视角。《人工智能与机器学习综合讲座》6(1):1–210。Kootbally Z（2016）面向敏捷制造的工业机器人能力模型。《Industrial Robot: An International Journal》。Koubˆaa A 等（2017）《机器人操作系统（ROS）》，第 1 卷。Springer。Kovacs DL（2011）PDDL 3.1 的巴科斯-诺尔范式定义。IPC-2011 网站上的未发表手稿。Kress-Gazit H, Fainekos GE, Pappas GJ（2009）基于时序逻辑的反应式任务与运动规划。《IEEE机器人学汇刊》25(6):1370–1381。Kress-Gazit H, Wongpiromsarn T, Topcu U（2011）正确的反应式高级机器人控制。《IEEE机器人与自动化杂志》18(3):65–74。Kumar A, Kala R（2016）基于线性时序逻辑的任务规划。《IJIMAI》3(7):32–41。Kumar V, Lin YJ（1988）一种用于 Prolog 的基于数据依赖的智能回溯方案。《逻辑程序设计杂志》5(2):165–181。Lacerda B, Lima PU（2012）根据 LTL 规范设计 Petri 网监督器。载于：机器人学：科学与系统，第 7 卷，页 169–176。Lahijanian M, Almagor S, Fried D, Kavraki LE, Vardi MY（2015）这次，机器人选择了成本较低的方案：一种具有部分满足的定量时序逻辑规划方法。载于：第二十九届AAAI人工智能会议。Laskey KB（2008）多实体贝叶斯网络：一阶贝叶斯知识库的语言。《人工智能》172(2-3):140–178。Lasota PA, Fong T, Shah JA 等（2017）安全人机交互方法综述。Now出版社。Lee J, Wang Y（2016）稳定模型语义下的加权规则。载于：第十五届知识表示与推理原理国际会议

Lee J, Wang Y（2018）动作语言的一种概率扩展。《逻辑编程理论与实践》18(3-4):607–622 Lemaignan S, Ros R, M¨osenlechner L, Alami R, Beetz M（2010）Oro：面向机器人学认知架构的知识管理平台。载于：2010年IEEE/RSJ智能机器人与系统国际会议，IEEE，第3548–3553页 Leonetti M, Iocchi L, Patrizi F（2012）有限状态控制器的自动生成与学习。载于：人工智能：方法、系统与应用国际会议，Springer，第135–144页 Leonetti M, Iocchi L, Stone P（2016）自动规划与强化学习的综合：实现高效、稳健的决策。《人工智能》241:103–130 Levesque HJ, Reiter R, Lesp´erance Y, Lin F, Scherl RB（1997）Golog：一种面向动态领域的逻辑程序设计语言。《逻辑程序设计杂志》31(1-3):59–83 Levinson J, Askeland J, Becker J, Dolson J, Held D, Kammel S, Kolter JZ, Langer D, Pink O, Pratt V, 等（2011）迈向完全自动驾驶：系统与算法。载于：2011年IEEE智能汽车研讨会，IEEE，第163–168页 Lifschitz V（1999）答案集规划。载于：逻辑编程与非单调推理国际会议，Springer，第373–374页 Lifschitz V（2008）什么是答案集编程? 载于：AAAI，第8卷，第1594–1597页 Likhachev M, Thrun S, Gordon GJ（2005）针对稀疏随机性的马尔可夫决策过程规划。载于：神经信息处理系统进展，第785–792页 Lin H（2014）任务完成：移动机器人运动规划与控制中的形式化方法简介。《无人系统》2(02):201–216 Lisca G, Nyga D, B´alint-Bencz´edi F, Langer H, Beetz M（2015）迈向开展化学实验的机器人。载于：2015年IEEE/RSJ智能机器人与系统国际会议，IEEE，第5202–5208页 Liu L, Peng J, Zhang R, Chen B, Yang Y, Zhang X（2017）面向智能变电站环境的智能机器人时序逻辑任务与运动规划。载于：2017年IEEE国际系统、人类与控制论会议（SMC 2017），第2017-January卷，第1110–1115页 Lloyd JW（2012）《逻辑程序设计基础》。Springer Science & Business Media Lu D, Zhou Y, Wu F, Zhang Z, Chen X（2017）将答案集编程与语义词典相结合，实现机器人任务规划。载于：人工智能国际联合会议，第4361–4367页 Lukasiewicz T（1998）概率逻辑编程。载于：ECAI，第388–392页 Lukasiewicz T（2001）带条件约束的概率逻辑编程。《ACM计算逻辑汇刊》2(3):289–339 Ma C, Fang F, Ma X（2016）分布式服务框架中的移动机器人任务规划。载于：工业电子会议论文集，第342–347页 Mandel C, Huebner K, Vierhuff T（2005）迈向自主轮椅：服务机器人学中的认知方面。载于：迈向自主机器人系统论文集，第165–172页 Manthey N（2016）迈向新一代串行与并行SAT求解器。《KI-K¨unstliche Intelligenz》30(3-4):339–342 Marek VW, Truszczy´nski M（1999）稳定模型与一种替代性的逻辑程序设计范式。载于：《逻辑程序设计范式》，Springer，第375–398页 Matari´c MJ（2006）情境机器人学。《认知科学百科全书》 Matar´e V, Schiffer S, Ferrein A（2018）Golog++：一种集成系统设计。载于：CEUR研讨会论文集，第2325卷，第29–35页 Maurelli F, Saigol Z, Lane D（2014）面向自主水下航行器的不确定性认知知识表示。载于：IEEE ICRA'14论文集，IEEE，香港，水下机器人持续自主性研讨会 McCarthy J（1980）circumscription——一种非单调推理形式。《人工智能》13(1-2):27–39 McDermott D, Doyle J（1980）非单调逻辑 I。《人工智能》13(1-2):41–72 Meli D, Sridharan M, Fiorini P（2021a）面向自主手术任务规划的答案集程序归纳学习。《机器学习》110(7):1739–1763 Meli D, Tagliabue E, Dall'Alba D, Fiorini P（2021b）采用基于生物力学的逻辑框架实现自主组织牵拉。载于：国际医疗机器人研讨会，IEEE，第1–7页 Miranda DSS, de Souza LE, Bastos GS（2018）基于ROSPlan的多机器人导航系统。载于：2018年拉丁美洲机器人研讨会、2018年巴西机器人研讨会和2018年机器人教育研讨会，IEEE，第248–253页 Moldovan B, De Raedt L（2014）双臂机器人关系可供性模型学习。载于：2014年IEEE/RSJ智能机器人与系统国际会议，IEEE，第2916–2922页

Moldovan B, Moreno P, Van Otterlo M, Santos-Victor J, De Raedt L（2012）面向机器人多物体操作任务的关系可供性模型学习。载于：2012年IEEE国际机器人与自动化会议，IEEE，第4373–4378页 Moldovan B, Antanas L, Hoffmann M（2013）开门：一种初步的SRL方法，《计算机科学讲义》（含《人工智能讲义》和《生物信息学讲义》子系列），第7842卷，LNAI Moldovan B, Moreno P, Nitti D, Santos-Victor J, De Raedt L（2018）多物体操作的关系可供性。《自主机器人》42(1):19–44 Moon J, Lee B（2019）利用基于自然语言的场景理解进行无人机-无人地面车辆协同的PDDL规划。《应用科学》9(18) Mu˜noz P, R-Moreno MD, Barrero DF, Ropero F（2019）MOBAR：一种分层面向行动的自主控制架构。《智能与机器人系统杂志：理论与应用》94(3-4):745–760 Mu˜noz-Hernandez S, Wiguna WS（2007）RoboCupSoccer中的模糊认知层，《计算机科学讲义》（含《人工智能讲义》和《生物信息学讲义》子系列），第4529卷，LNAI Nagy D ´A, Nagy TD, Elek R, Rudas IJ, Haidegger T（2018）基于本体的手术子任务自动化：自动执行钝性分离。《医学机器人研究杂志》3(03n04):1841005 Nagy TD, Haidegger TP（2021）迈向自主手术子任务执行评估的标准方法。载于：2021年IEEE第25届智能工程系统国际会议，IEEE，第67–74页 Nakawala H, Goncalves PJ, Fiorini P, Ferringo G, De Momi E（2018）机器人学中的动作序列表示方法综述。载于：2018年IEEE/RSJ智能机器人与系统国际会议，IEEE，第5666–5671页 Nakawala H, Bianchi R, Pescatori LE, De Cobelli O, Ferrigno G, De Momi E（2019）用于手术工作流程与情境识别的“deep-onto”网络。《国际计算机辅助放射学与外科学杂志》14(4):685–696 Narboni GA（1999）从Prolog III到Prolog IV：重新审视约束编程的逻辑。《Constraints》4(4):313–335 Nau D, Cao Y, Lotem A, Munoz-Avila H（1999）SHOP：简单分层有序规划器。载于：第16届国际人工智能联合会议论文集（第2卷），第968–973页 Nevlyudov I, Tsymbal O, Milyutina S（2008）产品装配工艺流程设计的逻辑模型。载于：TCSET 2008——无线电工程、电信与计算机科学的现代问题——国际会议论文集，第500–501页 Ng R, Subrahmanian VS（1992）概率逻辑编程。《信息与计算》101(2):150–201 Nitti D, Belle V, De Raedt L（2015）通过概率编程，在离散与连续马尔可夫决策过程中进行规划。载于：欧洲机器学习与数据库知识发现联合会议，Springer，第327–342页 Nyga D, Beetz M（2012）机器人一直想了解的家务事（但不敢问）。载于：2012年IEEE/RSJ智能机器人与系统国际会议，IEEE，第243–250页 Nyga D, Roy S, Paul R, Park D, Pomarlan M, Beetz M, Roy N（2018）利用不完整的世界知识，根据自然语言指令为机器人计划进行基化。载于：机器人学习会议，第714–723页 Opfer S, Jakob S, Geihs K（2018）动态领域中自主智能体的推理：迈向自动满足模块性质，《计算机科学讲义》（含《人工智能讲义》和《生物信息学讲义》子系列），第10839卷，LNAI Opfer S, J akob S, Jahl A, Geihs K（2019）ALICA 2.0——领域无关的团队协作，《计算机科学讲义》（含《人工智能讲义》和《生物信息学讲义》子系列），第11793卷，LNAI Papadimitriou G, Saigol Z, Lane DM（2015）利用本体实现反水雷任务中的故障恢复与适应。载于：MTS/IEEE OCEANS 2015——热那亚：探索可持续海洋能源，开创崭新世界 Pereira LM, Monteiro L, Cunha J, Aparicio JN（1986）Delta Prolog：一种带事件的分布式回溯扩展。载于：逻辑编程国际会议，Springer，第69–83页 Pianpak P, Son TC, Toups ZO, Yeoh W（2019）用于多智能体路径规划问题的分布式求解器。载于：ACM国际会议论文集系列 Pineda LA, Salinas L, Meza IV, Rascon C, Fuentes G（2013）Sit Log：一种服务机器人任务编程语言。《国际先进机器人系统杂志》10:1–12 Pnueli A（1977）程序的时序逻辑。载于：第18届计算机科学基础年度研讨会（1977），I EEE，第46–57页

Poli R, Healy M, Kameas A (2010) 本体论：计算机应用的理论与应用。Springer Raja P, Pugazhenthi S (2012) 移动机器人最优路径规划：综述。《国际物理科学杂志》7(9):1314–1320 Rajan K, PyF, Barreiro J (2013) 面向海洋机器人中的审议式控制。载于：Marine Robot Autonomy，Springer，第91–175页 Raman V, Kress-Gazit H (2012) 解释不可能实现的机器人高级行为。IEEE机器人学汇刊 29(1):94–104 Richardson M, Domingos P (2006) 马尔可夫逻辑网络。《机器学习》62(1-2):107–136 Richter S, Westphal M, Helmert M (2011) Lama 2008和2011。载于：国际规划竞赛，第117–124页 Rintanen J, et al. (2007) 并发时序规划的复杂性。载于：国际自动规划与调度会议，第7卷，第280–287页 Rutten JJ, Kwiatkowska M, Norman G, Parker D (2004) 分析并发与概率系统的数学技术。23，美国数学学会 Sanelli V, Cashmore M, Magazzeni D, Iocchi L (2017) 通过条件规划与执行实现短期人机交互。载于：第二十七届国际自动规划与调度会议 Sarid S, Xu B, Kress-Gazit H (2013) 在探索部分已知地图时保证机器人高级行为。载于：机器人学，麻省理工学院出版社，第377–384页 Sato T (1995) 一种基于分布语义的逻辑程序统计学习方法。载于：第十二届逻辑编程国际会议论文集 Schiffer S, Ferrein A, Lakemeyer G (2012) Caesar：一种智能家用服务机器人。智能服务机器人 5(4):259–273 Schillinger P, Burger M, Dimarogonas DV (2017) 针对有限线性时序逻辑规范和资源约束的最优多机器人规划进行多目标搜索。载于：IEEE国际机器人与自动化会议论文集，第768–774页 Schmuck A, Majumdar R, Leva A (2017) 动态分层反应式控制器综合。离散事件动态系统：理论与应用 27(2):261–299 Semwal T, Bode M, Singh V, Jha SS, Nair SB (2015) Tartarus：用于集成信息物理系统与机器人的多智能体平台。载于：ACM国际会议论文集系列，第02-04-July-2015卷 Shivashankar V, Alford R, Kuter U, Nau D (2013) Godel规划系统：领域无关规划与分层规划的更完美结合。载于：第二十三届国际人工智能联合会议 Smullyan RM (1995) 一阶逻辑。Courier Corporation Son TC, Baral C, Tran N, Mcilraith S (2006) 答案集规划中的领域相关知识。ACM计算逻辑汇刊 7(4):613–657 Soper NJ, Fried GM (2008) 腹腔镜手术基础：它的时代已经到来。Bull Am Coll Surg 93(9):30–32 Sorensson N, Een N (2005) Minisat v1. 13——一种具有冲突子句最小化功能的SAT求解器。SAT 2005(53):1–2 Sridharan M, Gelfond M, Zhang S, Wyatt J (2019) Reba：用于机器人领域知识表示与推理的基于细化的架构。《人工智能研究杂志》65:87–180 Syrj¨anen T, Niemel¨a I (2001) smodels系统。载于：逻辑程序设计与非单调推理国际会议，Springer，第434–438页 Tagliabue E, Meli D, Dall'Alba D, Fiorini P (2022) 自主机器人辅助手术中的审议：处理解剖结构不确定性的框架。载于：IEEE国际机器人与自动化会议 Tan ZX, Brawer J, Scassellati B (2019) 那是我的！让机器人学习所有权关系与规范。载于：AAAI 人工智能会议论文集，第33卷，第8058–8065页 Teichteil-K¨onigsbuch F, Kuter U, Infantes G (2010) 通过增量式计划聚合在马尔可夫决策过程中生成策略。载于：第九届自主智能体与多智能体系统国际会议论文集：第1卷，国际自主智能体与多智能体系统基金会，第1231–1238页 Tenorth M, Beetz M (2013) Knowrob：面向具备认知能力的机器人的知识处理基础设施。国际机器人研究杂志 32(5):566–590 Tenorth M, Beetz M (2017) KnowRob框架中的机器人知识表示。《人工智能》247:151–169 Thielscher M (2005) 在FLUX中处理蕴含与全称量化约束，《计算机科学讲义》（包括《人工智能讲义》和《生物信息学讲义》子系列），第3709卷 LNCS Trakhtenbrot BA (1962) 有限自动机与单子二阶逻辑。Siberian Math J 3:101–131

Tu PH, Son TC, Baral C (2007) 使用答案集编程，基于感知动作、不完整信息和静态因果律进行推理与规划。《逻辑编程理论与实践》7(4):377–450 Tumova J, Marzinotto A, Dimarogonas DV, Kragic D (2014) 最大限度满足线性时序逻辑动作规划。载于：IEEE 智能机器人与系统国际会议，第1503–1510页 Tzafestas S, Plouzennec A (1989) 一种基于Prolog的机器人任务规划专家系统原型。IFAC论文集 22(6):225–230 Van Emden MH, Kowalski RA (1976) 谓词逻辑作为编程语言的语义。《ACM学报》23(4):733–742 Vidal T (1999) 处理时间约束网络中的偶然性：从一致性到可控性。《实验与理论人工智能杂志》11(1):23–45 Vidal T, Fargier H (1997) 时间约束满足问题中的偶然持续时间：从一致性到可控性。载于：TIME'97：第四届时间表示与推理国际研讨会论文集，电气电子工程师学会，第78–85页 Vidal T, Ghallab M (1996) 处理专用于规划的时间约束网络中的不确定持续时间。载于：ECAI，PITMAN，第48–54页 Vitucci N, Gini G (2019) 使用描述逻辑对物体和抓取进行推理。《先进机器人学》33(13):616–635 Waibel M, Beetz M, Civera J, d'Andrea R, Elfring J, Galvez-Lopez D, H¨aussermann K, Janssen R, Montiel J, Perzylo A, et al. (2011) Roboearth。《IEEE机器人与自动化杂志》18(2):69–82 Wang X, Zhao L, Zeng C, Qian J, Gu T (2009) 一种基于答案集编程的机械装配序列规划解决方案。载于：第三届遗传与进化计算国际会议，WGEC 2009，第205–208页 Wielemaker J, Schrijvers T, Triska M, Lager T (2012) S WI-Prolog。《逻辑编程理论与实践》12(1-2):67–96 Wilke T (1999) 离散时间属性分类。载于：理论计算机科学年度研讨会，Springer，第32–46页 Wilkins DE (1984) 领域无关规划表示与计划生成。《人工智能》22(3):269–301 Wong KW, Kress-Gazit H (2016) 面向分布式高级机器人控制的基于需求的协调。载于：IEEE 智能机器人与系统国际会议，第2016年11月卷，第2209–2216页 Wongpiromsarn T, Topcu U, Murray RM (2010) 面向时序逻辑规范的滚动时域控制。载于：第十三届ACM混合系统：计算与控制国际会议论文集，第101–110页 Wongpiromsarn T, Topcu U, Ozay N, Xu H, Murray RM (2011a) TuLiP：用于滚动时域时序逻辑规划的软件工具箱。载于：第十四届混合系统：计算与控制国际会议论文集，ACM，第313–314页 Wongpiromsarn T, Topcu U, Ozay N, Xu H, Murray RM (2011b) TuLiP：用于滚动时域时序逻辑规划的软件工具箱。载于：HSCC'11——2011年ACM/SIGBED混合系统：计算与控制论文集，第313–314页 Wu Wh, Liu Ys (2008) 基于流演算和FLUX [j]的办公室机器人控制。《计算机工程与设计》11 Wurm KM, Dornhege C, Eyerich P, Stachniss C, Nebel B, Burgard W (2010) 使用时间符号规划，由有袋式机器人团队协同探索。载于：2010年IEEE/RSJ智能机器人与系统国际会议，电气电子工程师学会，第5014–5019页 Wurm KM, Dornhege C, Nebel B, Burgard W, Stachniss C (2013) 使用时间符号规划协调异构机器人团队。《自主机器人》34(4):277–294 Xu N, Li J, Niu Y, Shen L (2016) 一种面向自主机器人的基于线性时序逻辑的运动与动作动态规划方法。IFAC-PapersOnLine 49(5):91–96 Xu S, Wu G, Zhao Q, Tu D, Zheng H (2014) 支持独立生活的人机环境信息物理系统与分层任务规划。载于：2014年IEEE生物信息学与生物医学国际会议论文集，IEEE BIBM 2014，第568–573页 Yazdani F, Blumenthal S, Huebel N, Bozcuo˘glu AK, Beetz M, Bruyninckx H (2019) 面向搜救任务的异构知识库基于查询的集成。《机器人学与自主系统》117 Younes HL, Littman ML (2004) Ppddl1. 0：一种用于表达具有概率效应的规划领域的PDDL扩展。技术报告 CMU-CS-04-162 2:99 Zacharaki A, Kostavelis I, Gasteratos A, Dokas I (2020) 人机交互中的安全界限：综述。《安全科学》127:104667 Zhang W, Hong S (2019) 并行Prolog系统的线性归约模型。载于：2019年IEEE第四届云计算与大数据分析国际会议，IEEE，第650–654页

Zhang X、Zhu Y、Lin H（2016年）通过正确性设计实现有性能保证的人机协作。载于：美国控制会议论文集，第2016年7月卷，第6183–6188页。Zhang Y、Sreedharan S、Kulkarni A、Chakraborti T、Zhuo HH、Kambhampati S（2017年）机器人任务规划中的计划可解释性与可预测性。载于：2017年IEEE国际机器人与自动化会议，电气电子工程师学会，第1313–1320页。Zhao Y、Topcu U、Sentis L（2016年）非结构化环境中全身运动的高级规划器综合。载于：2016年 IEEE第五十五届决策与控制会议，电气电子工程师学会，第6557–6564页