

面向智能服务机器人任务规划的行动语言扩展*

靳国强, 陈小平

(中国科学技术大学 计算机科学与技术学院, 安徽 合肥 230027)

通讯作者: 靳国强, E-mail: jguoqiang@ustc.edu.cn

摘要: 针对智能服务机器人的任务规划, 引入复合行动对行动语言 C+ 进行了扩展, 并实现了其求解系统. 在扩展的行动语言 C+ 中, 复合行动被定义成一定条件下一系列基本行动的连续执行. 通过刻画扩展的行动描述和其对应的转移系统的关系, 证明了扩展行动语言相对于原始行动语言的可靠性和完备性. 在智能服务机器人的任务规划中, 复合行动可以看成是一种对于机器人能力的“高层”抽象. 这样的扩展使得对于机器人规划系统的建模更加直观, 具有更大的灵活性, 并且扩展有增量式的优点. 实验结果表明, 通过引入复合行动, 对于比较复杂的机器人任务规划问题, 可以很好地改进求解效率.

关键词: 非单调推理; 自动规划; 服务机器人

中图法分类号: TP242 **文献标识码:** A

中文引用格式: 靳国强, 陈小平. 面向智能服务机器人任务规划的行动语言扩展. 软件学报, 2013, 24(7): 1614-1625. <http://www.jos.org.cn/1000-9825/4331.htm>

英文引用格式: Jin GQ, Chen XP. Extending action languages for intelligent service robot task planning. Ruan Jian Xue Bao / Journal of Software, 2013, 24(7): 1614-1625 (in Chinese). <http://www.jos.org.cn/1000-9825/4331.htm>

Extending Action Languages for Intelligent Service Robot Task Planning

JIN Guo-Qiang, CHEN Xiao-Ping

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Corresponding author: JIN Guo-Qiang, E-mail: jguoqiang@ustc.edu.cn

Abstract: To improve the task planning module in intelligent service robots, an extension of the action language C+ is proposed and implemented by introducing composite actions as a sequential executions of other actions. The soundness and completeness of the extension is proved by relating the action description in the extended C+ to its corresponding transition system. In the domain of robotic task planning, a composite action can be treated as a “high-level” abstraction of the robot’s physical functions. Such an extension leads to a more intuitive and flexible representation of the robot’s task planning system, and the knowledge of composite actions can be added to the domain incrementally. The experimental results show that for large domains, the extension leads to a great improvement of the solving efficiency.

Key words: non-monotonic reasoning; automatic planning; service robot

由于广阔的应用前景, 服务机器人受到了广泛的关注和深入的研究. 近年来, 服务机器人各种功能的研究取得了很大的进展^[1-3], 因此, 国内外相关领域逐渐对服务机器人的集成及智能技术展开了研究^[4-7]. 同时, 随着服务机器人的执行环境和任务向复杂化和多样化发展, 对于服务机器人的任务规划也提出了挑战. 近年来, 国内外学者主要采用具有自动推理和规划能力的任务规划模块来处理这类问题. 文献[8]指出, 服务机器人通用控制软件应该含有一个独立的决策层, 即任务规划层, 不足的是, 其中提出的决策层不具备自动推理与规划的能力, 需要手工进行决策. 文献[4]提出了一个可以支持个性化协调的服务机器人的体系结构, 包括一个可以自动进行推

* 基金项目: 国家自然科学基金(60745002, 61105039, 61175057); 国家高技术研究发展计划(863)(2008AA01Z150)

收稿时间: 2012-05-12; 修改时间: 2012-07-26; 定稿时间: 2012-10-19

理和规划的任务规划模块,在此基础上,开发并实现了服务机器人可佳(KeJia)^[4,6],其中的任务规划模块采用回答集编程(answer set programming,简称 ASP)^[9]来实现.文献[5]则采用行动语言 C+^[10]来描述服务机器人的抓取任务规划,文献[11]采用行动语言 C+实现了多机器人之间的共同任务规划.在机器人任务规划模块的实现过程中,与回答集编程的方式相比,采用行动语言 C+的方式具有表达形式更加简洁、语义更加清晰、开发过程更加直观和方便的特点.在行动语言 C+的求解器方面,经过一段时间的发展,求解器 *cplus2asp* 采用将行动语言翻译到回答集程序的方式进行求解,比文献[5,11]中使用的求解器 *CCalc* 的效率有了进一步的提高^[12].

在使用行动语言或回答集编程的方式进行机器人任务规划时,首先需要根据机器人的硬件及底层感知-控制系统,将机器人的基本行动表达为对应的行动符号.例如,将移动能力表达为 *move*,抓取和放下物体的能力表达为 *pickup* 以及 *putdown* 等.在实际的机器人应用中,任务通过语音交互等方式指定之后,任务规划系统会自动规划,得出完成这个任务的行动序列.之后,机器人只需执行这个序列就可以完成指定的任务.但是,通过直接描述机器人的基本行动来进行推理和规划的方式也存在一些不足:首先,在具体规划过程中,这样的方式显得比较复杂,而且不太直观.例如,对于任务“帮我取来一杯热水”,任务规划系统会直接根据机器人基本能力的描述进行搜索,最终给出一个详细的基本行动序列,这个序列反映了采用基本的行动序列来完成的任务的完整方法.但是人类对这个任务的解决会基于一些“高层的”行动,例如首先获取(*fetch*)一杯水,然后对其进行加热(*heat*),最后将其送过来(*bring*).这样的方式更加直观,易于理解,同时也更加快捷;其次,与采用了高层行动的规划序列相比,一个完成任务的完整序列一般比较长,求解这样一个规划也会显著地增加求解时间.

为了在表达和求解效率上进一步提高机器人的任务规划能力,我们引入复合行动对行动语言 C+进行扩展.一个复合行动是在一定条件之下的基本行动的顺序执行.直观上,复合行动表达的正是能够用于解决规划问题的“高层行动”.例如,对于之前描述的任务,我们可以将机器人从一个位置将物体取到另一个位置定义成一个复合行动.其具体内容为:首先,如果机器人和目标物体不在同一个位置,那么机器人移动到这个位置,之后,依次执行抓取物体、移动到目标位置、放下物体的行动.在具体规划过程中,这些复合行动直接参与构成最终的规划序列,这些复合行动同时也包含了其构成的所有基本行动,使得机器人可以直接执行这个规划序列.

在经典规划领域,国内外学者对采用复合行动,如宏行动(*macro action*),来改进规划求解效率的方式进行了深入的研究^[13-15].这些工作中,描述行动的改变和推理都基于 STRIPS^[16],或者规划领域描述语言(PDDL)^[17].这些框架由于自身的限制,对行动的间接效果或惯性特性的表达和处理比较困难,难以直观地描述系统中的行动.并且在这些系统中的宏行动主要基于任务分解的方式,而不是作为类似基本行动的结构参与规划.基于情形演算(*situation calculus*)^[18]的编程语言 Golog^[19]引入了类似复合行动的结构,但是 Golog 中的结构描述的是一个问题的求解的整个框架.这个框架在求解中用于减小搜索空间,并不出现在最后的规划结果中.文献[20]中提出的“复杂行动”丢掉了详细执行细节,因此具体执行时还需要将其转换成基本行动的序列.更重要的是,上述工作都基于单调逻辑系统,不适用于机器人的规划领域.机器人在执行任务时,随时可以通过自身的感知和人机对话等手段获取新的知识.这些知识很可能与机器人的原始知识库有冲突,在单调的逻辑系统中,这些冲突会使得整个系统产生不一致,因此很难用于进一步规划.而在基于非单调逻辑的推理系统,例如行动语言 C+或回答集编程中,这些知识可以很方便地加入到知识库,进一步用于将来的规划,符合容变(*elaboration tolerance*)^[21]的要求.在非单调逻辑的框架下,文献[22,23]的工作都涉及到类似复合行动的结构.但文献[22]的工作只是将 Golog 中类似复合行动的结构在回答集编程下进行表达,作为规划过程的一个启发式信息;文献[23]中的复合行动只能是一个固定长度的基本行动的相继执行,不能通过条件来限定其是否执行,并且没有对复合行动扩展的可靠性及完备性加以论证.

对于采用复合行动进行扩展后的行动语言 C+,我们讨论了它的各种性质,并建立了扩展的行动语言及其所表达的转移系统的关系,进一步证明了扩展后系统的可靠性和完备性.同时,我们也扩展了求解系统 *cplus2asp*,增加了对复合行动的支持,并且在实验中进行了测试.实验结果表明,对于复杂的任务,采用复合行动的描述对求解效率有比较显著的改进.

1 背景知识

1.1 非单调因果理论(non-monotonic causal theory)

一个(多值的命题)基调(signature)是一个符号的集合 σ ,叫做常量集,以及一个与 σ 不同的非空有限符号集合 $Dom(c)$.集合 $Dom(c)$ 是 σ 的值域, σ 中的每个常量 c 都被指派到 $Dom(c)$ 中的一个值.原子是形如 $c=v$ 的表达式,其中, $c \in \sigma, v \in Dom(c)$ 表达“ c 的值是 v ”.公式是原子的命题组合.基调的解释 I 是将该基调中所有元素映射到其值域的函数,如果 $I(c)=v$,那么我们说解释 I 满足原子 $c=v$,或 I 将原子 $c=v$ 解释为真. I 对公式的可满足性的解释符合经典命题逻辑中对命题组合公式以及连接词的解释.

一条因果规则是形如 $F \Leftarrow G$ 的规则,其中, F 和 G 都是公式,分别叫做该因果规则的头和体.这条规则的直观解释为“当 G 为真时,有一个原因导致 F ”.一个因果理论是一个有限的因果规则的集合.

定义 1^[10]. 对于一个因果理论 T 和一个其基调的解释 I, T 相对于 I 的规约(reduct) T^I 是 T 中所有的规则体被 I 满足的那些规则的头部的集合.如果 I 是集合 T^I 的唯一模型,那么我们说 I 是 T 的模型.

1.2 行动语言C+

行动语言C+(的)多值基调 σ 中的常量分为行动常量(action constant)和变式常量(fluent constant).变式常量还可以分为简单变式常量(simple fluent constant)和静态确定变式常量(statically determined constant).

对于一个确定的基调 σ ,如果一个公式中所有出现的常量都是变式常量,那么这个公式被称为变式公式;否则,如果一个公式中含有至少一个行动常量并且没有变式常量,那么这个公式被称为行动公式.

一个行动描述由一个因果律的集合组成,因果律包含如下两种形式的表达式:

- 第1种形式的表达式形如

$$\text{caused } F \text{ if } G \quad (1)$$

其中, F 和 G 是公式.如果它们都是变式公式,那么这个因果律被称为静态律;如果 F 是行动公式,那么这个因果律被称为行动动态律.静态律通常用于描述变式之间的关系,表达行动的间接效果;行动动态律用于表达行动本身的性质,或者行动与其他行动以及变式之间的关系.

- 第2种形式的表达式形如

$$\text{caused } F \text{ if } G \text{ after } H \quad (2)$$

其中, F 和 G 是变式公式,并且 F 中不包含静态确定的变式常量; H 是一个公式.该因果律叫做变式动态律.

变式动态律通常用于描述规划系统中行动的直接效果.

由公式(1)和公式(2)可以定义很多缩略因果律,例如表达式

$$a \text{ causes } F \text{ if } G \quad (3)$$

其中, a 是行动常量,表示 **caused** F **if** True **after** $a \wedge G$.

其他有用的表达式包括:

$$\text{inertial } c \quad (4)$$

其中, c 是变式常量.该表达式叫做惯性律,表示 **caused** c **if** c **after** c ,通常用于表达变式的惯性性质,以及

$$\text{exogenous } a \quad (5)$$

其中, a 是行动常量.该表达式表示因果律 **caused** a **if** a , **caused** $\neg a$ **if** $\neg a$,用于表达规划系统中行动的性质,以及

$$\text{default } a \quad (6)$$

其中, a 是行动常量.该表达式表示因果律 **caused** a **if** a ,和表达式

$$\text{nonexecutable } H \text{ if } F \quad (7)$$

其中, H 是行动公式, F 是变式公式.该表达式表示 **caused** False **if** True **after** $H \wedge F$.

行动描述 D 可以转化成为一个有限的因果理论 $D_m(m \geq 0)$ 的集合,这个集合定义了 D 的语义. D_m 的基调是序对 $i:c$ 的集合,其中,当 c 为变式常量时,有 $i \in \{0, \dots, m\}$,表达了这个因果理论中所有的状态;当 c 为行动常量时,有 $i \in \{0, \dots, m-1\}$,表达了因果理论中所有的行动.

因果理论 D_m 中的因果规则包括如下 3 个部分:

- 1) 对每一条形如公式(1)的因果律,规则集 $i:F \Leftarrow i:G$,如果该因果律是 D 的静态律,那么 $i \in \{0, \dots, m\}$;如果它是行动动态律,那么 $i \in \{0, \dots, m-1\}$;
- 2) 对每一条变式动态律(2),规则集 $i+1:F \Leftarrow (i+1:G) \wedge (i:H)$,其中, $i \in \{0, \dots, m-1\}$;
- 3) 对每一个简单变式常量 c 及其值 $v \in \text{Dom}(c)$,规则集 $0:c=v \Leftarrow 0:c=v$.

因果理论 D_m 的模型对应于 D 所表示的转移图中一条长为 m 的路径(见文献[10]中的命题 8),这条路径中包含 $m+1$ 个状态和 m 个行动.当用行动语言来描述机器人的任务规划系统时,需要将机器人的基本功能表达为行动语言中的行动常量,并且通过因果律将这些基本行动常量的功能表达出来.

例 1:考虑一个具有移动和抓取能力的机器人,可以将物体移动到各个地方.我们采用如下的行动和变式常量对其进行建模:基本行动 $Move(l)$ 表达机器人可以移动到 l , $Pickup(s)$ 表示机器人可以将 s 拿起, $Putdown(s)$ 表示机器人可以将 s 放下;惯性变式 $Loc(o)$ 用于表达物体 o 的位置, $Hold(s)$ 表达机器人持有 s .

- **inertial** $Loc(o)=l$ **inertial** $Hold(s)$
- **exogenous** $Move(l)$ **exogenous** $Pickup(s)$ **exogenous** $Putdown(s)$
- **caused** $Loc(s)=l$ **if** $Hold(s) \wedge Loc(Robot)=l$
- $Move(l)$ **causes** $Loc(Robot)=l$ **nonexecutable** $Move(l)$ **if** $Loc(Robot) \neq l$
- $Pickup(s)$ **causes** $Hold(s)$ **nonexecutable** $Pickup(s)$ **if** $\neg Hold(Nothing)$
- **nonexecutable** $Pickup(s)$ **if** $Loc(Robot) \neq Loc(s)$
- $Putdown(s)$ **causes** $Hold(Nothing)$ **nonexecutable** $Putdown(s)$ **if** $\neg Hold(s)$

上述行动描述的第 1 行表达了两个变式 $Loc(o)$ 和 $Hold(s)$ 具有惯性性质,即如果没有外部行动改变,那么它们的值保持不变.第 2 行表达了对应的 3 个行动是外生的(exogenous),即在任何一个时刻,这些行动都可以发生或不发生.第 3 行是静态律,表达了变式 $Loc(l)$ 和变式 $Hold(s)$ 以及 $Loc(Robot)$ 的关系,实际上表达了行动的间接效果.例如,在机器人持有 s 时,机器人执行行动 $Move$ 除了会直接导致机器人本身的位置发生变化以外,还会间接导致 s 的位置发生变化.其余部分分别表达了对应行动的直接效果和执行条件.

设 $l \in \{L_1, L_2\}$, $o \in \{Robot, S\}$, $s \in \{S\}$, 我们得到了行动描述 D^0 .图 1 表达了 D^0 的一个模型.

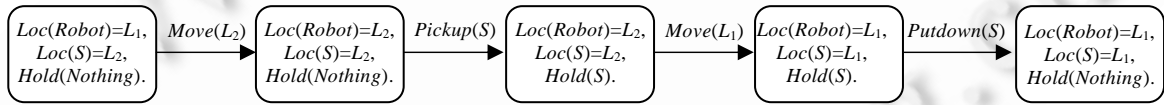


Fig.1 Part of the transition diagram of D^0

图 1 D^0 的部分转移图

当采用行动语言 C+ 进行具体的规划问题求解时,一个规划问题被分成了两部分:第 1 部分是针对整个规划问题的行动描述,例如 D^0 ;第 2 部分是针对这个规划问题初始状态以及目标状态的表达,主要是一些变式的值.例如,对于 D^0 的描述,给定初始状态机器人在 L_1 位置, S 在 L_2 位置,并且机器人手抓为空.目标状态为:机器人手抓为空, S 在 L_1 位置.经过对应的编码之后,这部分与行动描述部分一起经过 `cplus2asp` 求解,就可以得到一个规划的行动序列.对于上述的具体规划问题,求解出来的行动序列正如图 1 中的边所示,即机器人首先移动到 L_2 ,之后抓取 S ,然后移动到 L_1 ,最后在 L_1 放下 S .规划过程中整个系统的状态变化如图 1 中的顶点所示.

2 利用复合行动扩展行动语言

2.1 语法

考虑行动语言 C+ 的一个子集,包括 F 和 G 都是变式公式的静态律(1)、形如公式(5)的行动动态律、形如公式(3)、公式(4)和公式(7)的变式动态律.给定这个子集上的行动描述 D 和它的一个变式常量集合 σ^f ,以及行动常量集合 σ^{act} .一个扩展的行动描述 D^+ 是在 D 中引入一个复合行动常量集合 σ^{comp} ,以及如下形式的复合行动定

义规则:

$$b \text{ is } a_0 \text{ if } E_0; a_1 \text{ if } E_1; \dots; a_k \text{ if } E_k \quad (8)$$

其中, $b \in \sigma^{comp}$, 叫做复合行动常量; $a_0, \dots, a_k \in \sigma^{ac,t}$ 叫做复合行动 b 的子行动; a_j 是 b 的第 j 个子行动; E_0, \dots, E_k 是变式公式. 这个规则表达执行复合行动 b 等价于在 E_0 成立的时候首先执行 a_0 , 然后在 E_1 成立的时候接着执行 a_1, \dots , 最后在 E_k 成立的时候执行 a_k . 特别地, 如果相应的条件 E_i 不成立, 那么对应的 a_i 被跳过不执行.

例 2: 我们引入复合行动 $Fetch(s, l)$ 对例 1 进行扩展:

$$Fetch(s, l) \text{ is } Move(l_1) \text{ if } Loc(s) = l_1 \wedge Loc(Robot) \neq l_1; Pickup(s); Move(l); Putdown(s) \quad (9)$$

复合行动 $Fetch(s, l)$ 表示机器人将物品 s 直接运送到位置 l , 并且还完整定义了这个复合行动的执行细节: 如果机器人不在 s 的位置, 那么首先移动到 s 的位置; 如果机器人本身和 s 在同一个位置, 那么不需要移动. 之后, 机器人拿起 s , 移动到位置 l , 最后放下 s .

2.2 语义

我们规定, 与基本行动一样, 一个复合行动在规划的执行过程中也花费一个时间周期. 因此在规划过程中, 机器人会优先选择相应的复合行动, 从而减小规划步长, 提高规划效率. 那么, 由于复合行动本身包含了其中基本行动的执行细节, 我们需要将一个时间周期扩展为一个二维的结构. 对一个扩展的行动描述 D^+ , 令 k^* 是其中所有形如公式(8)的定义中最大的 k 值, σ_0 是所有复合行动定义中子行动 a_i 的集合. 与 D 类似, 我们也将 D^+ 转换到一个有限的因果理论序列 $D_m^+ (m \geq 0)$, 通过这个因果理论序列定义其语义. 转换的具体过程如下所述:

首先, 我们在一个时间周期 $(i, i+1)$ 中增加 k^* 个子时间点 $i, 1, \dots, i.k^*$, 将其划分成 k^*+1 个子周期, 即

$$(i, i.1), (i.1, i.2), \dots, (i.k^*, i+1).$$

这样, 复合行动中的所有执行细节都可以在一个周期内得到解释. 我们规定, 每个常量在增加的 k^* 个子时间点上也有取值. 那么, D_m^+ 的基调包括所有 D_m 的基调中出现的符号以及如下序对:

- 1) $i.j: a_i$, 其中, $i \in \{0, \dots, m-1\}, j \in \{0, \dots, k^*\}, a_i \in \sigma_0$;
- 2) $i.j: c$, 其中, $i \in \{0, \dots, m-1\}, j \in \{1, \dots, k^*\}, c$ 是 D 中的变式常元.

D_m^+ 中的规则包括:

1) D_m 中除了从公式(4)转换来的规则以外的所有规则. 这些规则表明, 对于 D_m^+ , 其中的基本行动的执行依然要花费一个时间周期.

- 2) 对每一条静态律(1)以及 $i \in \{0, \dots, m-1\}, j \in \{1, \dots, k^*\}$, 增加规则

$$i.j: F \Leftarrow i.j: G \quad (10)$$

该规则表示在每一个子时间点上, 由静态律规定的两个变式公式之间的关系依然成立, 因此, 在子时间点上, 行动产生的间接效果依然成立.

- 3) 对每条变式动态律(4)以及 $i \in \{0, \dots, m-1\}, j \in \{1, \dots, k^*-1\}, v \in Dom(c)$, 增加规则

$$\begin{aligned} i.1: c = v &\Leftarrow i.1: c = v \wedge i: c = v \\ i.j+1: c = v &\Leftarrow i.j+1: c = v \wedge i.j: c = v \\ i+1: c = v &\Leftarrow i+1: c = v \wedge i.k^*: c = v \end{aligned} \quad (11)$$

这些规则表示惯性规则被表达成一组在子周期上成立的惯性规则, 从而使得在增加的时间点上取值的变式也遵从惯性律.

- 4) 对每一条复合行动定义(8), 增加如下规则:

- 4.1) 对 $i \in \{0, \dots, m-1\}$, 增加

$$\begin{aligned} i.1: F &\Leftarrow i.1: G \wedge (i: H)^* \\ i.j+1: F &\Leftarrow i.j+1: G \wedge i.j: H \end{aligned} \quad (12)$$

如果 $k=k^*$, 那么还增加

$$i+1: F \Leftarrow i+1: G \wedge i.k^*: H \quad (13)$$

其中,行动常量 $a_j(0 \leq j \leq k)$ 都出现在公式(3)中,或者它们是 H 中唯一的行动常量; $(i:H)^*$ 是将 $i:H$ 中的行动常量 $i:a$ 替换成 $i.0:a$ 而得到的.这些规则表达当行动被复合行动触发时,可在子周期上产生与其作为基本行动一样的效果.另外,如果一个复合行动的子行动数目小于 k^*+1 个,那么在这个复合行动的 $k+1$ 个子行动都执行之后,就没有行动继续执行了.其中改变的变式的值,通过公式(3)中定义的子周期上的惯性定律传递到下一个时间周期.例如, $k^*=2$ 表示行动描述中的一个复合行动最多可能有 3 个子行动.复合行动 b 的定义为 $b \text{ is } a;a$.如果 b 在周期 $(i,i+1)$ 执行,那么,在 $(i,i.1)$ 和 $(i.1,i.2)$ 周期会执行 a ,产生相应的效果;而在 $(i.2,i+1)$ 的子周期没有任何基本行动执行.执行两次 a 产生的效果直接由 $i.2$ 时间点通过公式(3)定义的惯性规则传递到 $i+1$.

4.2) 对每一个动态变式规则(7),并且其中 H 包含不止 a_0 一个行动常量,那么对于 $i \in \{0, \dots, m-1\}$,增加

$$\text{False} \Leftarrow (i:H_{a_0}^b) \wedge i:F \quad (14)$$

其中, $H_{a_0}^b$ 是将 H 中所有出现的 a_0 全部替换成 b 而得到的.这条规则表明,一个复合行动,如果它的第 0 个子行动和其他的行动不能同时执行,那么这个行动与包含该子行动的复合行动也不能同时执行,这样可以避免复合行动与其他基本行动之间的冲突.例如,对复合行动 b 的第 0 个子行动 a ,有约束 a 和 c 不能同时执行,那么,同样地,复合行动 b 和 c 也不能同时执行,这样可以维护原始行动描述本身的约束.

4.3) 对每一个 $i \in \{0, \dots, m-1\}, j \in \{1, \dots, k\}, t \in \{0, \dots, k\}$,增加

$$i:b \Leftarrow i:b \quad i:\neg b \Leftarrow i:\neg b \quad (15)$$

$$i.0:a_0 \Leftarrow i:b \wedge i:E_0 \quad i.j:a_j \Leftarrow i:b \wedge i.j:E_j \quad (16)$$

$$\text{False} \Leftarrow i:a_t \wedge i:b \quad (17)$$

这些规则表达任何复合行动都是默认可执行的.在条件 E_j 成立时,相应的子行动被“触发”,并且复合行动不能与其子行动对应的基本行动同时执行.

5) 对于 $i \in \{0, \dots, m-1\}, j \in \{0, \dots, k^*\}$ 以及 $a_t \in \sigma_0$,增加

$$i.j:\neg a_t \Leftarrow i.j:\neg a_t \quad (18)$$

这条规则表明,任何复合行动的子行动默认是不执行的,它们只能被复合行动在相应条件成立的时候触发.

6) 对于 $i \in \{0, \dots, m\}, b_m, b_n \in \sigma^{comp}$,增加

$$\text{False} \Leftarrow i:b_m \wedge i:b_n \quad (19)$$

这条规则限定了任意两个复合行动不能同时执行.

例 3: 例 2 中的 $(D^0)^+$ 可以翻译成因果理论 $(D^0)_m^+$ 的集合(默认 $i \in \{0, \dots, m-1\}, j \in \{1, \dots, k^*-1=2\}$):

1) D_m^0 中除了惯性律转换成的规则之外的所有规则.

2) 静态律如 **caused** $Loc(s)=l$ **if** $Hold(s) \wedge Loc(Robot)=l$ 被翻译成如下规则集($t \in \{1, 2, 3\}$):

$$i.t:Loc(s)=l \Leftarrow i.t:Hold(s) \wedge i.t:Loc(Robot)=l.$$

3) 惯性律被翻译成公式(11).例如, **inertial** $Loc(o)=l$ 被翻译成如下规则集:

$$\begin{aligned} i.1:Loc(o)=l &\Leftarrow i:Loc(o)=l \wedge i.1:Loc(o)=l, \\ i.j+1:Loc(o)=l &\Leftarrow i.j+1:Loc(o)=l \wedge i.j:Loc(o)=l, \\ i+1:Loc(o)=l &\Leftarrow i+1:Loc(o)=l \wedge i.3:Loc(o)=l. \end{aligned}$$

4) $Move(l)$ **causes** $Loc(Robot)=l$ 被翻译成如下规则集($t \in \{0, 1, 2\}$):

$$\begin{aligned} i.t+1:Loc(Robot)=l &\Leftarrow i.t:Move(l), \\ i+1:Loc(Robot)=l &\Leftarrow i.3:Move(l). \end{aligned}$$

5) **nonexecutable** $Move(l)$ **if** $Loc(Robot)=l$ 被翻译成如下规则:

$$\text{False} \Leftarrow i.j:Move(l) \wedge i.j:Loc(Robot)=l.$$

6) 从公式(9)我们可以得到如下规则集(仅考虑前两个子行动):

$$\begin{aligned} i:Fetch(s,l) &\Leftarrow i:Fetch(s,l) \quad i:\neg Fetch(s,l) \Leftarrow i:\neg Fetch(s,l), \\ i.0:Move(l_1) &\Leftarrow i:Fetch(s,l) \wedge i:Loc(s)=l_1 \wedge i:Loc(Robot) \neq l_1 \quad i.1:Pickup(s) \Leftarrow i:Fetch(s,l), \\ \text{False} &\Leftarrow i:Move(l_1) \wedge i:Fetch(s,l) \quad \text{False} \Leftarrow i:Putdown(s) \wedge i:Fetch(s,l). \end{aligned}$$

7) 对于所有子行动,例如 $Move(l)$,需要增加如下规则($t \in \{0,1,2,3\}$):

$$i.t:\neg Move(l) \Leftarrow i.t:\neg Move(l).$$

3 扩展的行动语言的性质

本节研究扩展的行动语言的性质.与文献[10]对原始行动语言 C+性质的表达类似,我们主要通过扩展的行动语言 C+对应的转移系统来刻画其性质.一个基调的解释 I 可以表达为形如 $c=I(c)$ 的原子的集合,即将其中的原子都解释成真的集合.这样,行动描述 D_m^+ 的模型可以表达为

$$\bigcup_{i=0}^m i:s_i \cup \bigcup_{i=0}^{m-1} i:e_i \cup \bigcup_{i=0}^{m-1} \left(\bigcup_{j=1}^{k^*} i.j:s_{i,j} \cup \bigcup_{j=0}^{k^*} i.j:e'_{i,j} \right) \quad (20)$$

其中, e_0, \dots, e_{m-1} 是 σ^{act} 的解释, $s_0, \dots, s_m, s_{i,1}, \dots, s_{i,k^*}$ 是 σ^f 的解释,而 $e'_{i,0}, \dots, e'_{i,k^*}$ 是 σ_0 的解释.对任意 $m \geq 0$,我们将 D_m^+ 的模型都称为 D^+ 的模型.

一个状态是一个 σ^f 的解释 s ,使得 $0:s$ 是 D_0^+ 的模型.状态是 D_m^+ 所表达的转移图上的顶点.转移图的边(转移)通过 D_1^+ 的模型来定义.实际上, D_1^+ 的模型可以表达为

$$(0:s) \cup (0:e) \cup (1:s') \cup \bigcup_{0 \leq i \leq k^*} (0.i:e'_i) \cup \bigcup_{1 \leq i \leq k^*} (0.i:s_i) \quad (21)$$

例如,例 2 中的 $(D^0)^+$ 的一个模型可以表达为(略去解释为假的那些行动常量):

$$\begin{aligned} & (0: Loc(Robot) = L_1, Loc(S) = L_2, Hold(Nothing)) \cup (1: Loc(Robot) = L_1, Loc(S) = L_1, Hold(Nothing)) \cup \\ & (0.1: Loc(Robot) = L_2, Loc(S) = L_2, Hold(Nothing)) \cup (0.2: Loc(Robot) = L_2, Loc(S) = L_2, Hold(S)) \cup \\ & (0.3: Loc(Robot) = L_1, Loc(S) = L_1, Hold(S)) \cup (0: Fetch(S, L_1)) \cup (0.0: Move(L_2)) \cup \\ & (0.1: Pickup(S)) \cup (0.2: Move(L_1)) \cup (0.3: Putdown(S)) \end{aligned} \quad (22)$$

一个显式转移是一个三元组 $\langle s, e, s' \rangle$,其中, s 和 s' 是 σ^f 的解释, e 是 $\sigma^{act} \cup \sigma^{comp}$ 的解释,并且 $(0:s) \cup (0:e) \cup (1:s')$ 是 D_1^+ 的某个模型的子集.如果有某个 $b \in \sigma^{comp}$ 满足 $e(b)=t$,那么 $\langle s, e, s' \rangle$ 叫做复合转移,否则叫做简单转移.

一个转移扩展是一个元组 $\langle s, e'_0, s_1, \dots, s_{k^*}, e'_{k^*}, s' \rangle$,其中, e'_i 是 σ_0 的解释, s_i 是 σ^f 的解释,并且集合

$$(0:s) \cup (0.0:e'_0) \cup (0.1:s_1) \cup \dots \cup (0.k^*:s_{k^*}) \cup (0.k^*:e'_{k^*}) \cup (1:s')$$

是 D_1^+ 的某个模型的子集.一个转移扩展可以看做是 k^*+1 个如下形式的三元组列表:

$$\langle s, e'_0, s_1 \rangle, \dots, \langle s_{k^*}, e'_{k^*}, s' \rangle.$$

每一个这样的三元组称为一个隐式转移.

如果对于任意 b 的子行动 $a_j(0 \leq j \leq k)$, $e'_j(a_j) = \mathbf{f}$ 都成立,那么这个转移扩展叫做关于 b 的平凡转移扩展. D^+ 所表达的转移图中的边由 D_1^+ 的模型所表达的转移组成.

根据以上定义,对于扩展的行动语言描述,有如下性质成立:

命题 1. 对于任意显式转移 $\langle s, e, s' \rangle$ 或隐式转移 $\langle s, e', s' \rangle$, s 和 s' 都是一个状态.

该命题是文献[10]中命题 7 在 D^+ 中的增强.其有效性也依赖于如下事实:静态确定的变式不能出现在变式动态律(2)的头部.这条命题表明,显示转移或隐式转移都可以对应到转移系统中的一条长度为 1 的路径.

文献[10]的命题 8 描述了 D 与其对应的因果理论的模型之间的联系.对于扩展的行动描述 D^+ ,其与对应的因果理论的模型之间也有类似的关系.

命题 2. 对任意 $m > 0$,基于 D_m^+ 的基调的一个解释公式(20)是 D_m^+ 的模型,当且仅当对 $0 \leq i \leq m-1$,每个三元组 $\langle s_i, e, s_{i+1} \rangle$ 都是一个显式转移,每个元组 $\langle s_i, e'_{i,0}, s_{i,1}, \dots, s_{i,k^*}, e'_{i,k^*}, s_{i+1} \rangle$ 都是一个转移扩展.

根据命题 1 和命题 2,对于任意的 m , D_m^+ 的模型都可以表达成 D 所对应的转移系统中的一条长为 m 的路径,以及这条路径上的复合转移对应的隐式转移构成的路径集合.因此,一个扩展的行动描述可以用于表达一个转移系统.下面我们研究原始行动描述与扩展行动描述的转移之间的关系.

定义 1(相对转移). 对原始的行动描述 D 和采用复合行动扩展之后的行动描述 D^+ :

- (i) 对于 D^+ 中的一个转移 $t=\langle s, e, s' \rangle$, t 相对于 D 的转移是三元组 $t^-=\langle s, e^-, s' \rangle$, 其中, e^- 是 σ^{act} 的解释. 如果 t 是一个显式转移, 那么对任意 $a \in \sigma^{act}, e^-(a)=e(a)$; 如果 t 是一个隐式转移, 那么对于任意 $a \in \sigma_0, e^-(a)=e(a)$; 对于其他 $a \in \sigma^{act}, e^-(a)=f$;
- (ii) 对于 D 中的一个转移 $t=\langle s, e, s' \rangle$, t 相对于 D^+ 的转移是三元组 $t^+=\langle s, e^+, s' \rangle$, 其中, e^+ 是 $\sigma^{act} \cup \sigma^{comp}$ 的解释, 并且对于任意 $a \in \sigma^{act}, e^+(a)=e(a)$; 对于任意 $a \in \sigma^{comp}, e^+(a)=f$.

可以看到, 在原始的行动描述和扩展之后的行动描述之间, 相对转移都是一一对应的. 相对转移可以方便地刻画原始行动描述和扩展行动描述的转移之间的关系.

命题 3. 对原始行动描述 D 和采用复合行动扩展之后的行动描述 D^+ , 有:

- (i) 对 D 中的任何一个转移 t , 其相对于 D^+ 的转移 t^+ 都是 D^+ 中的一个简单转移;
- (ii) 对 D^+ 中的任何一个简单转移或隐式转移 t , 其相对于 D 的转移 t^- 是 D 中的转移.

命题 3 表明, 对于 D 的任意一个转移, D^+ 中都有一个转移与之对应, 反之亦然. 同时, 该命题还表明, D^+ 中的一个转移扩展可以对应到 D 的转移系统中的一条长为 k^*+1 的路径. 图 2 表达了 $(D^0)^+$ 的模型(22), 其中的虚线表示隐式转移. 可以看到, 所有的隐式转移都对应了图 1 中的转移, 模型(22)中的转移扩展对应了图 1 中一条长为 4 的路径.

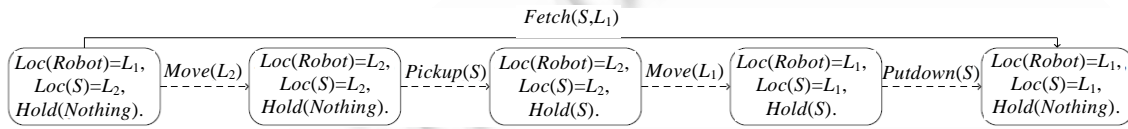


Fig.2 Transitions represented by the model (22) of $(D^0)^+$

图 2 $(D^0)^+$ 的模型(22)对应的转移

根据文献[24]的定义, 对行动描述 D 和 D' , D 的基调是 D' 基调的子集, 将 D' 的状态和转移限制在 D 的基调上, 如果所构成的新的转移系统与 D 的转移系统同构, 则称 D 是 D' 的子描述.

命题 4. 令 D 是一个基于基调 σ 的行动描述, b 是一个不在 σ 中的常量, 如果 D' 是基于基调 $\sigma \cup \{b\}$ 的行动描述, 并且是由在 D 中添加一条 b 的复合行动定义规则(8)构成的, 满足 b 的子行动都是 σ 中的行动常量, 那么 D 是 D' 的子描述.

例如, 对例 2 中的行动描述 $(D^0)^+$, 我们去掉其中与 $Fetch(s, l)$ 相关的描述规则之后, 得到的行动描述的转移系统与 D^0 对应的转移系统是同构的.

多次应用命题 4, 我们可以得出 D 是 D^+ 的子描述. 因此, 它们的模型是相互关联的. 下面我们根据 D 和 D^+ 的模型之间的关系, 讨论 D^+ 对于 D 的扩展的完备性和可靠性.

定义 2(模型扩张). 对任意 $m \geq 0$, 原始行动描述 D 的一个模型 $M=(0:s_0) \cup (0:e_0) \cup (1:s_1) \cup \dots \cup (m:s_m)$ 相对于 D^+ 的模型扩张是集合 $M'=(0:s_0) \cup (0:e_0^+) \cup (1:s_1) \cup \dots \cup (m:s_m)$, 其中, 任意一个三元组 $\langle s, e_i^+, s' \rangle$ 都是 D 中的转移 $\langle s, e_i, s' \rangle$ 相对于 D^+ 的转移.

我们可以通过模型扩张来描述 D 和 D^+ 的模型的关系, 如果对于 D 的任意模型, 其模型扩张是 D^+ 的某个模型子集, 那么我们称 D^+ 对 D 扩展是完备的.

定理 1. D^+ 对 D 的扩展是完备的.

完备性定理表明, 对于原始的行动描述, 采用复合行动进行扩展之后, 原来系统的任何模型都可以在新的系统中表达出来. 事实上, 新的系统中可能还有其他的模型. 因此, 对于实际应用中的机器人规划, 采用复合行动扩展之后, 进行任务规划时不会出现原始系统能够计算出规划结果而新的系统中无解的情况.

定义 3(模型约减). 对任意扩展行动描述 D^+ 的一个表达成公式(20)形式的模型 M , 其相对于 D 的模型约减是集合 $M'=(0:s_0) \cup (0:e_0^-) \cup (1:s_1) \cup \dots \cup (n:s_n) (n \geq m)$, 其中, 任意一个三元组 $\langle s, e_i^-, s' \rangle$ 都是 D^+ 中的简单转移或

隐式转移 $\langle s, e_i, s \rangle$ 相对于 D 的转移.

模型约减也可用来表达 D^+ 和 D 的模型之间的联系.如果 D^+ 的任意一个模型相对于 D 的模型约减都是 D 的模型,那么我们称 D^+ 对 D 的扩展是可靠的.

定理 2. D^+ 对 D 的扩展是可靠的.

可靠性定理表明,在使用复合行动对于行动描述的扩展中,并没有引入与原来的描述不一致的内容.因此,在实际的服务机器人规划中,我们可以保证在通过复合行动进行知识扩展和更新的过程中,所有的新知识都与原系统一致,可以进一步用于此后的任务规划.

4 实验设计与分析

Cplus2asp 系统的求解主要分为两部分,根据文献[12]中的描述,首先是将一个行动描述,包括其初始状态和目标状态的变式描述,一起翻译成一个回答集程序,之后,采用回答集编程的高效求解器,例如 clasp^[25]来进行求解,并将得到的回答集转化成相应的规划.在翻译成回答集程序的过程中,首先对给定的 m ,需要将行动描述 D 转化成对应的因果理论 D_m 的集合,并将这个集合中的每一个因果规则对应翻译成回答集程序中的语句.我们扩展的支持复合行动的 cplus2asp 程序也符合这个流程.采用第 2.2 节中的方法,我们将扩展的复合行动定义规则转化成因果理论规则,并且实现了这些因果理论规则向回答集的转换.最后,通过调用回答集编程的求解器,对整个回答集程序进行求解,将得到的结果转化成具体的规划行动序列.

为了研究扩展的行动语言 C+ 用于服务机器人的具体规划效率,我们采用服务机器人可佳(KeJia)^[4]中运行的实际规划问题进行实验.可佳的硬件系统中包括一个自由移动平台、一个机械手臂和手抓以及一个能够临时放置小物体的盘子.通过人机对话,可佳获取人的需求,根据需要完成一些简单的家务,例如移动一些小物体、送饮料等,并且可佳还能操作常用的家用电器,例如微波炉.

目前,基于可佳的硬件系统已经实现的基本行动包括移动到指定位置、抓取和放下指定物体等.我们将所有可佳可以实现的基本行动表示为如下符号: $Move(l), Pickup(s), Putdown(s), Open(m), Close(m), Putin(s, m), Takeout(s, m)$ 以及 $Start(m)$, 分别表示移动到指定位置 l 、抓起指定物体 m 、放下物体 s 、打开机器 m 的门(例如微波炉)、关闭 m 的门、将 s 放到 m 里面、将 s 从 m 中拿出来以及启动 m .此外,根据可佳的感知系统,在上述某些行动执行后,可佳也可以观察到运行的实际状态的变化.我们采用一些变式来描述这些状态的变化.目前,可以感知的变式包括描述物体以及自身位置的变式 $Loc(o)=l$ 、描述小物体是否在容器内部的变式 $Inside(c)=s$ 等.根据以上的行动和变式符号,行动 $Putin(s, m)$ 的效果和可执行条件可以描述如下:

$Putin(s, m) \text{ causes } Inside(m) = s \quad Putin(s, m) \text{ causes } Inside(Hand) = \text{Nothing},$
nonexecutable $Putin(s, m) \text{ if } Inside(Hand) \neq s \quad \text{nonexecutable } Putin(s, m) \text{ if } Loc(Robot) \neq Loc(m),$
nonexecutable $Putin(s, m) \text{ if } \neg Dooropen(m) \quad \text{nonexecutable } Putin(s, m) \text{ if } Inside(m) \neq \text{Nothing}.$

第 1 行的两条因果律表达执行行动 $Putin(s, m)$ 之后,物体 s 将置于 m 内部,并且机器人的手抓中不再有任何物体.后面两行表示,如果 s 不在机器人的手抓中、或者机器人和 m 不在同一个位置以及 m 的门没有打开、或 m 中已经不是空的,那么行动 $Putin(s, m)$ 都不能执行.我们可以类似地表达其他各个行动,以及采用静态律来表达变式之间的关系.我们将这样的行动描述记为描述 $KeJia_1$.

由于可佳也经常完成一些使用家用电器的任务,因此我们在描述 $KeJia_1$ 中引入操作电器的复合行动,使用某些家用电器的基本步骤有一些相似性,即“首先将物体放入机器中,之后将其门关上,然后启动机器,过一定时间之后,再打开门,将物体取出来”.根据这个模式,用微波炉(M)加热食物(f)可以表达如下:

$Heat(f, M) \text{ is } Putin(f, M); Close(M); Start(M); Open(M); Takeout(f, M).$

同样地,使用洗衣机(W)来洗衣服(c)也可以表达为

$Wash(c, W) \text{ is } Putin(c, W); Close(W); Start(W); Open(W); Takeout(c, W).$

实际上,这两类复合行动都是模式,根据相应的参数会生成不同的复合行动实例.例如,对于复合行动 $Heat(f, M)$,如果在场景中有一个可以加热的实物以及两台微波炉,那么实际上我们定义了 4 个不同的复合行动.

在具体的求解过程中,复合行动的模式会根据具体的参数自动地进行实例化.我们将这两类复合行动的描述直接加入到 $KeJia_1$ 中,形成行动描述 $KeJia_2$.根据机器人可以移动的位置的数目、可以抓取的物体的数目以及完成任务的类型(移动物体、加热食物、洗衣服等)的不同配置,我们随机生成了 240 组初始状态和目标状态的集合,即 240 个不同的具体任务配置.对于其中每一个配置,我们分别将行动描述 $KeJia_1$ 和 $KeJia_2$ 与其组合成一组具体的规划任务.对于每一组这样的规划任务,我们采用原始的 $cplus2asp$ 系统求解 $KeJia_1$ 描述的任务,采用我们扩展实现的 $cplus2asp$ 系统来求解 $KeJia_2$ 描述的任务,并且对结果进行对比.一个典型的任务配置,如在场景中有 5 件待洗衣物、4 样未加热食物、机器人、微波炉、洗衣机以及其他物体处于各自的初始位置.要求机器人通过规划完成清洗所有的衣物,并且将所有食物都加热好,送到对应的位置.在实验过程中,我们设置最大求解时间为 30 分钟,最大的求解步长为 50 步,即如果 50 个行动还不能完成这个任务,那么我们认为该任务无解.实验在一台 Intel i7 950 CPU,4G RAM 的 Linux 机器上进行.

对于每一个配置,我们分别记录了采用两种行动描述进行规划,得到一个规划解的时间和步长.有 34 个配置,采用两种描述都不能在规定的时间内得出结果.对于其他的任务配置,经过检验,所有得出的规划解都是正确的.我们根据 $KeJia_1$ 对具体规划任务成功求解时的规划步长进行分类,将所有规划任务分成了 5 类.对每一类,规划步长表示采用 $KeJia_1$ 求解这些问题得出的解的长度范围,超时数目表示在对应的类别中,使用 $KeJia_1$ 的描述在规定时间内没有求解出来结果、但是使用 $KeJia_2$ 求解出来的问题的数目. $KeJia_1$ 和 $KeJia_2$ 用时分别表示对同一类问题采用两种描述求解的平均时间,加速比表示对同一个任务配置分别采用 $KeJia_1$ 和 $KeJia_2$ 得到一个解的时间比值的平均值,对于超时问题,没有加入计算.具体结果见表 1.

Table 1 Results of the planning problem of KeJia

表 1 机器人可佳规划问题的实验结果

规划步长	问题数目	超时数目	KeJia ₁ 用时(s)	KeJia ₂ 用时(s)	加速比
<20	71	0	1.042	6.528	0.159
21~25	25	0	4.900	14.802	0.331
26~30	45	0	173.767	96.984	1.792
31~35	45	9	907.147	566.931	1.600
36~40	20	10	1 323.542	719.749	1.839

表 1 表明,当只要求解步长大于 25 时,采用复合行动的描述总是比原始行动的描述的效率要高.加速比在 1.5~2 左右.即在问题复杂到需要 25 步或以上的行动序列才能求解时,使用复合行动进行扩展的行动描述在解决同样的问题配置时具有更好的求解效率.但是,当求解步长小于 25 时,使用原始的行动描述比采用复合行动扩展的求解效率更高.不过,这两种描述求解时间都小于 15s.在我们测试的其他领域规划实验中也得到了类似的结果,即在问题复杂到一定程度之后,对于同样的问题配置,采用复合行动扩展的行动描述的求解效率总是比原始行动描述要高.

经过进一步分析我们发现,在问题求解步长小于 25 时,由于在复合行动的描述中增加了一些规则,这些规则经过处理,使得最后生成的回答集程序比原始行动描述的规模要大,回答集程序的求解器在求解时花费了大部分时间对这些规则进行常例化(grounding).同时,由于问题本身比较简单,因此求解器实际用于搜索得到解的时间比较少,导致使用复合行动求解上的优势没有完全发挥出来,因此,采用原始行动描述的方式效率反而更高.而对于规模较大的问题,虽然求解时常例化的时间有所增加,但是由于复合行动包含了一些行动的连续执行,这些行动都可以在一个求解步长中得到应用,一个复合行动的效果相当于一系列连续执行的基本行动,有助于求解器在求解的早期通过这些连续行动的累积效果,尽快地引导自身找到目标解,因此可以大大缩短实际用于求解的时间,使得效率有比较可观的提升.因此,通过实验我们可以得出,对于规模较大的问题,特别是那些基本行动在规划中常常相继出现、因而可以抽象成“高层行动”的问题,采用复合行动的方式可以很好地提高效率.

此外,在实验中我们扩展的那些复合行动的描述,即在 $KeJia_2$ 中使用的那些复合行动的知识,被写在了一个专门的文件中.在具体求解过程时,这些文件直接和 $KeJia_1$ 的描述文件一起进行计算,这个过程中没有对 $KeJia_1$ 的描述进行任何修改.这个事实表明,在原有的行动描述中增加复合行动描述是一个“增量式”的过程,这个特性

应用到机器人中,可以很方便地对机器人的原有知识库进行扩展.将对应的复合行动知识直接增加到机器人规划问题的行动描述,用于以后的规划过程,从而可以提高机器人的规划效率.同时,实验结果也表明,对于每个问题的解,复合行动与基本行动一样,都参与了规划结果的构建.并且,在出现复合行动的规划结果中,其所有细节也作为基本行动序列全部出现.这样,在机器人具体完成任务时就可以按照基本行动直接执行,不需要对复合行动作进一步的处理.

5 结论及展望

在本文中,我们首先通过引入复合行动的方式扩展了行动语言 C+,然后研究了扩展后的语言的一些性质,并证明了扩展后的语言与其描述的转移系统间的对应关系.根据这个对应关系,我们进一步证明了扩展后的行动语言的可靠性和完备性.最后,我们对求解系统 `cplus2asp` 进行了扩展,实现了其对复合行动的支持:一方面,在描述一个机器人相关领域的规划问题时,复合行动具有更加直观、更好的表达性和灵活性的特点;另一方面,加入复合行动之后的问题描述,对于规模较大的规划问题能够很好地提升求解效率.

在未来的研究中,一个直接的工作是将扩展的 `cplus2asp` 系统直接部署到服务机器人可佳上,与可佳的其他模块相结合来求解现实世界的各种问题.由于复合行动的知识可以增量式地加入到机器人的知识库中,因此,结合可佳的语音交互和自然语言理解模块,我们可以研究通过自然语言或者网络搜索的方式进行知识获取的问题.同时,我们可以进一步研究机器人任务规划的可重用性和模块化,通过改进已有的通用行动库,例如模块化行动描述 MAD^[24,26],将复合行动扩展成通用库中的行动连续执行,以增强复合行动描述的通用性.

致谢 在此,我们向对本文工作给予极大帮助的德州大学奥斯汀分校的杨方凯博士、对于文章提出建设性意见的 Vladimir Lifschitz 教授、Michael Gelfond 教授、Alfredo Gabaldon 博士和 Daniela Inclezan 博士表示感谢.

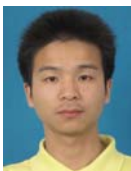
References:

- [1] Huang YT, Chen WD, Sun YX. Safety design and realization of an assistive robotic manipulator based on collision detection. *Robot*, 2011,33(1):40–45 (in Chinese with English abstract). [doi: 10.3724/SP.J.1218.2011.00040]
- [2] Rusu RB, Cousins S. 3D is here: Point cloud library (PCL). In: Proc. of the 2011 IEEE Int'l Conf. on Robotics and Automation (ICRA 2011). Shanghai: IEEE, 2011. [doi: 10.1109/ICRA.2011.5980567]
- [3] Cai ZX, He HG, Chen H. Control Theory and Methods of Mobile Robots' Navigation in Unknown Environments. Beijing: Science Press, 2009 (in Chinese).
- [4] Chen XP, Ji JM, Jiang JH, Jin GQ, Wang F, Xie JK. Developing high-level cognitive functions for service robots. In: van der Hoek W, Kaminka GA, Lesperance Y, Luck M, Sen S, eds. Proc. of the 9th Int'l Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2010). Toronto: IFAAMAS, 2010. 989–996. [doi: 10.1145/1838206.1838339]
- [5] Erdem E, Haspalanmutgil K, Palaz C, Patoglu V, Uras T. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA 2011). Shanghai: IEEE, 2011. 4575–4581. [doi: 10.1109/ICRA.2011.5980160]
- [6] Ji JM, Chen XP, Jiang JH, Jin GQ, Wang F. A service robot architecture supporting individualized coordination. *Journal of Nanjing University (Natural Sciences)*, 2010,(2):131–139 (in Chinese with English abstract).
- [7] Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Berger E, Wheeler R, Ng A. ROS: An open-source robot operating system. In: Proc. of the Open-Source Software Workshop of the IEEE Int'l Conf. on Robotics and Automation. Kobe, 2009. <http://www.willowgarage.com/papers/ros-open-source-robot-operating-system>
- [8] Wu B, He YY. Research and realization of universal control software of service robot. *Computer Measurement & Control*, 2010, 18(5):1186–1188 (in Chinese with English abstract).
- [9] Gelfond M, Lifschitz V. The stable model semantics for logic programming. In: Kowalski R, Bowen K, eds. Proc. of the Int'l Logic Programming Conf. and Symp. MIT Press, 1988. 1070–1080.
- [10] Giunchiglia E, Lee JH, Lifschitz V, McCain N, Turner H. Nonmonotonic causal theories. *Artificial Intelligence*, 2004,153(1-2): 49–104. [doi: 10.1016/j.artint.2002.12.001]
- [11] Aker E, Erdogan A, Erdem E, Patoglu V. Causal reasoning for planning and coordination of multiple housekeeping robots. In: Delgrande JP, Faber W, eds. Proc. of the 11th Int'l Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR 2011). Vancouver: Springer-Verlag, 2011. 311–316. [doi: 10.1007/978-3-642-20895-9_36]

- [12] Casolary M, Lee JH. Representing the language of the causal calculator in answer set programming. In: Gallagher JP, Gelfond M, eds. Proc. of the Technical Communications of the 27th Int'l Conf. on Logic Programming (ICLP 2011). Lexington: Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2011. 51–61. [doi: 10.4230/LIPIcs.ICLP.2011.51]
- [13] Korf RE. Planning as search: A quantitative approach. *Artificial Intelligence*, 1987,33(1):65–88. [doi: 10.1016/0004-3702(87)90051-8]
- [14] Erol K, Hendler J, Nau DS. HTN planning: Complexity and expressivity. In: Hayes-Roth B, Korf RE, eds. Proc. of the 12th National Conf. on Artificial Intelligence. Seattle: AAAI Press, 1994. 1123–1128.
- [15] Sacerdoti ED. Planning in a hierarchy of abstraction space. In: Nilsson NJ, ed. Proc. of the 3rd Int'l Joint Conf. on Artificial Intelligence (IJCAI'73). Stanford: William Kaufmann, 1973. 412–422.
- [16] Fikers RE, Nilsson NJ. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 1971,2(3/4):189–208. [doi: 10.1016/0004-3702(71)90010-5]
- [17] Ghallab M, Howe A, Knoblock C, Mcdermott D, Ram A, Veloso M, Weld D, Wilkins D. PDDL—The planning domain definition languages. Technical Report, CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.
- [18] Mccarthy J, Heyes PJ. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 1969,4: 463–502. <http://www-formal.stanford.edu/jmc/mcchay69/mcchay69.html>
- [19] Levesque HJ, Reiter R, Lesperance Y, Lin FZ, Scherl RB. Golog: A logic programming language for dynamic domains. *Journal of Logic Program*, 1997,31(1-3):59–83. [doi: 10.1016/S0743-1066(96)00121-5]
- [20] Mcilraith S, Fadel R. Planning with complex actions. In: Benferhat S, Guinchiglia E, eds. Proc. of the 9th Int'l Wrokshop on Non-Monotonic Reasoning (NMR 2002). Roulouse, 2002. 356–364. <http://www.informatik.uni-trier.de/~ley/db/conf/nmr/nmr2002.html#McIlraithF02>
- [21] Mccarthy J. Elaboration tolerance. In: Miller R, Shanahan M, eds. Proc. of 4th Symp. on Logical Formalizations of Commonsense Reasoning (Commonsense'98). Menlo Park: AAAI Press, 1998. <http://www-formal.stanford.edu/jmc/elaboration/>
- [22] Son TC, Baral C, Tran N, Mcilraith S. Domain-Depend knowledge in answer set planning. *ACM Trans. on Computational Logic*, 2006,7(4):613–657. [doi: 10.1145/1183278.1183279]
- [23] Inclezan D, Gelfond M. Representing biological processes in modular action language ALM. In: Proc. of the 2011 AAAI Spring Symp. on Formalizations of Commonsense Reasoning. AAAI Press, 2011. 49–55. <http://www.informatik.uni-trier.de/~ley/db/conf/aaais/aaais2011-6.html#InclezanG11>
- [24] Erdogan ST, Lifschitz V. Actions as special cases. In: Doherty P, Mylopoulos J, Welty CA, eds. Proc. of the 10th Int'l Conf. on Principles of Knowledge Representation and Reasoning (KR 2006). Lake District of the United Kingdom: AAAI Press, 2006. 377–388.
- [25] Gebser M, Kaufmann B, Neumann A, Schaub T. clasp: A conflict-driven answer set solver. In: Baral C, Brewka G, Schlipf JS, eds. Proc. of the 9th Int'l Conf. of Logic Programming and Nonmonotonic Reasoning (LPNMR 2007). Tempe: Springer-Verlag, 2007. 260–265. [doi: 10.1007/978-3-540-72200-7_23]
- [26] Lifschitz V, Ren WW. A modular action description language. In: Proc. of the 21st National Conf. on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conf. Boston: AAAI Press, 2006. 853–859. <http://www.informatik.uni-trier.de/~ley/db/conf/aaai/aaai2006.html#LifschitzR06>

附中参考文献:

- [1] 黄云天,陈卫东,孙逸翔.基于碰撞检测的护理型操作臂的安全性设计与实现. *机器人*,2011,33(1):40–45. [doi: 10.3724/SP.J.1218.2011.00040]
- [3] 蔡自兴,贺汉根,陈虹.未知环境中移动机器人导航控制理论与方法.北京:科学出版社,2009.
- [6] 吉建民,陈小平,姜节汇,靳国强,王锋.一种支持个性化协调的服务机器人体系结构. *南京大学学报(自然科学版)*,2010,(2): 131–139.
- [8] 吴斌,何永义.服务机器人通用控制软件的研究与实现. *计算机测量与控制*,2010,18(5):1186–1188.



靳国强(1985—),湖北当阳人,博士生,主要研究领域为智能服务机器人,知识表示与推理,自动规划.
E-mail: jguoqiang@ustc.edu.cn



陈小平(1955—),男,博士,教授,博士生导师,CCF会员,主要研究领域为人工智能逻辑,多主体系统,自主机器人系统关键技术.
E-mail: xpchen@ustc.edu.cn