

一种基于规划知识图的规划识别算法*

姜云飞¹, 马宁²

¹(中山大学 软件研究所, 广东 广州 510275);

²(吉林大学 计算机科学系, 吉林 长春 130023)

E-mail: lncsri05@zsu.edu.cn

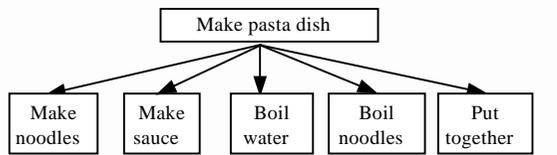
http://www.zsu.edu.cn; http://www.zsusoft.com

摘要: 提出了一种新的规划表示方法及规划知识图的概念.与目前规划识别领域广泛使用的 Kautz 方法相比,新的规划表示方法更加简便与直观.由于在规划知识图中增加了支持程度的概念,使得规划的识别可以随着收集到的新证据而合理地加以改变.所提出的基于规划知识图的规划识别算法把规划的识别问题转化为图搜索问题,因而其效率更高,但其识别结果与 Kautz 方法是一致的.

关键词: 规划识别;事件;时序约束

中图法分类号: TP18 文献标识码: A

规划识别^[1]是根据观察到的片段、琐碎的现象,推出具有合理的因果联系的、完整而全面的规划描述的过程.一个规划识别器推出的规划既能补充一些我们未观察到而又实际发生的现象,同时还可以预测未来.合理地推出 Agent 未来可能采取的动作.假设我们观察到 John 做好了调味汁,现在正在烧水.那么,基于如图 1 所示的规划,我们有理由相信 John 的目标是做 pasta,同时我们可以预测他下一步是煮面条,而且已经做好面条.



做意大利面条, 做面条, 做调味汁, 烧水, 煮面条, 混合.

Fig.1 A simple plan
图 1 一个简单的规划

规划识别在很多领域都具有广泛的应用.例如:故事理解^[2]、心理学模型^[3]、自然语言片段^[4]以及智能计算机系统接口^[5]等.在这些特定领域中的规划识别模型大多数都是先确定一个规划库,然后用搜索及匹配机制去识别里面的规划.其中,包含观察到的动作的规划称为候选规划.通常无二义性地确定一个 Agent 的规划是很困难的.

因为对 Agent 动作的观察经常是不完整的,而且往往在多个规划中可能包含有相同的动作.

由于存在着这些规划识别问题求解上的困难,很多研究人员提出了各式各样的解决办法.有基于 Bayesian 概率理论的规划识别^[2]、有采用语法分析的方法分析规划描述^[6]、有利用溯因理论进行识别过程的推导^[7]、更有基于限定理论的规划识别^[1].其中采用 Bayesian 网络的方法在自然语言理解中有其天然的优势,但这种方法由于其表示形式的特殊性,因而在大多数场合并不适合.另外,采用语法分析的方法虽然有其搜寻速度快的优点,但是它不可能像 Kautz 的方法具有很强的表达能力.例如,它无法表示和处理 Agent 可能附带的时序及其他约束,它也不可能处理抽象类型和具体类型之间的特征共享等问题.Kautz 在文献[1]中认为,溯因方法与他提出的限定方法有相似之处,而且他解决了溯因方法不能处理的抽象继承等问题.总的说来,Kautz 的方法具有非常丰富的表达能力,是目前最著名、也最具广泛影响的一种表示及处理规划识别的方法.

* 收稿日期: 2000-05-27; 修改日期: 2001-02-16

基金项目: 国家自然科学基金资助项目(69873047,60173039);广东省自然科学基金资助项目(980260)

作者简介: 姜云飞(1945 -),男,吉林长春人,教授,博士生导师,主要研究领域为计算机自动推理,基于模型的诊断,智能规划;马宁(1963 -),女,吉林长春人,博士生,主要研究领域为智能规划.

然而,采用 Kautz 的识别框架通常又是很困难的.为了简化 Kautz 的识别算法,很多人做了大量努力.我们在这里提出一种利用知识图进行规划识别搜索的方法.这种方法在表示上基本遵循 Kautz 的表示.我们可以证明利用知识图搜索方法获得的识别结果与 Kautz 方法是一致的.由于知识图搜索方法把规划识别问题转化为了图搜索问题,因而其效率更高.

在规划识别中,通常根据时序约束检查来排除大部分不满足条件的规划.Kautz 的时序约束检查,采用的是 Allen^[8]的时间片逻辑的表示方法.这种方法虽然有表达能力强的特点,但它的计算复杂程度却是 NP-完全的.这使得在 Kautz 的方法中时序相互作用既不能直接地用于规划的动作表示,也不能用于识别推理,而只是在识别出规划以后,再根据动作间的时序关系进行约束检查,以排除不可满足的规划.为了降低计算的复杂度,我们用整数标记时间片的办法,它虽然简单,但用在规划的搜索及匹配过程中却很有效.因为它可以直接根据现象发生的时间顺序直接进行检查.从而在很多情况下,在搜索过程的初期就可以排除大量不满足条件的规划,这使得用知识图搜索方法访问节点数更少,更节约时间和空间.

根据观察到的现象,有时可能会找到多个满足条件的规划.为了求得最优解,通常都采用信任度量的方法.Kautz^[1]也肯定了此种方法.但由于计算复杂,他还是采用与信任度量方法相似的、选择具有最少规划的规划集作为最优解.这种方法虽然节省了时间,但在很多情况下并不都能行得通.因为最少并不意味着最优.通常,包含了很多子事件的大型规划出现的机会总是比这些子事件单独出现的机会小得多.所以,为了获得一个满意解,还是应该采用信任度量的方法.这种方法由于考虑到了规划在客观条件下出现的可能性及现象对规划的支持程度,即根据观察到的现象推测出规划出现的可能性,所以它比 Kautz 的方法所求得解应该是更可信的.

1 Kautz 的规划识别表示

Kautz^[1]把规划识别中的事件(规划)表示为一个层次结构.这个结构中主要有两类表达式:抽象和分解.分解表达式用来说明一个事件是由几个子事件组成的.如图 2 所示:在 Kautz^[1]的 Make-Pasta-Dish 规划中,Make-Pasta-Dish 可以分解为 3 个子事件:Make-Noodles,Boil-Water 和 Make-Sauce.抽象表达式说明事件(规划)间的 IS-A 关系,例如:Make-Fettucini-Marinara 抽象于 Make-Pasta-Dish.这表明 Make-Fettucini-Marinara 是 Make-Pasta-Dish 的一个具体实例,换言之,Make-Fettucini-Marinara 应当继承 Make-Pasta-Dish 的所有特征,即它也可分解为以上说明的 3 个子类型事件.这两种形式用一阶逻辑可以分别表示为

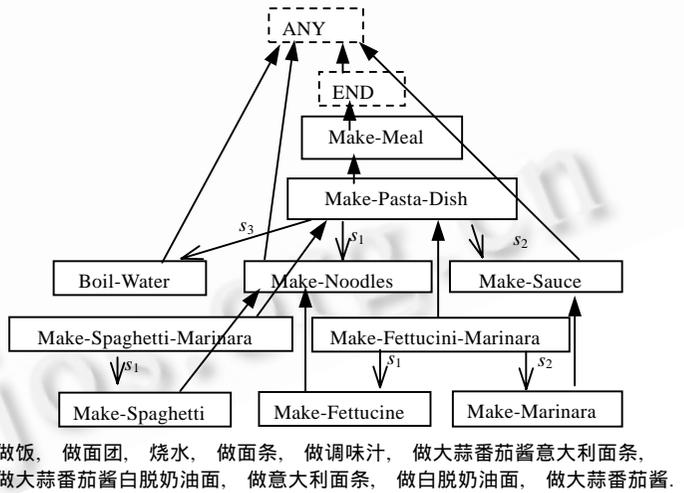


Fig.2 A hierarchical plan
图 2 一个分层规划

$$\forall x \text{ Make-Pasta-Dish}(x) \supset \text{Boil-Water}(s_3(x)) \wedge \text{Make-Noodles}(s_1(x)) \wedge \text{Make-Sauce}(s_2(x))$$

$$\forall x \text{ Make-Fettucini-Marinara}(x) \supset \text{Make-Pasta-Dish}(x)$$

其中 s_1, s_2, s_3 是步骤函数,表明整体和部分之间的映射关系.

Kautz 把所有的规划分为两类:一类是不具有独立意义的规划,它们总是表示成其他规划的一部分,也是需要进一步推导的规划,例如图 2 中的(Make-Noodles).另一类是具有独立意义的规划,即规划推导的终端,例如图 2 中的(Make-Meal).这两种类型的规划都抽象于一个特殊的规划 ANY.为了区分这两类规划,Kautz 引进了一个特殊的符号 END,凡是抽象于 END 的规划都是具有独立意义的规划,也就不需要进一步的推导,以此作为推导的控制机制.对于这种表示,很多人提出了不同意见^[2],主要原因是很难区分具有独立意义的规划的界限.我们在

这里也不采用这种形式来控制推导的过程.

Kautz 认为规划识别的解集应当是能够推出观察到的现象的最小规划集.这与 McCarthy 限定理论的思想很相似.据此,Kautz^[1]提出一个基于限定求解规划识别的框架.这个框架的核心是 Kautz 提出的 4 个假设,其中第 4 个假设是选择具有最少规划的规划集作为解集.而其他 3 个假设,Kautz^[1]证明了根据它们所求的解满足限定理论.这 3 个假设是:

穷尽(exhaustiveness)假设:设 $\Psi(x)$ 是一个抽象事件, $\varphi_1(x) \dots \varphi_n(x)$ 是已知属于事件 $\Psi(x)$ 的下一层的具体事件, 则 $\varphi_1(x) \dots \varphi_n(x)$ 就是 $\Psi(x)$ 的所有具体事件.即: $\forall x \Psi(x) \supset \varphi_1(x) \vee \dots \vee \varphi_n(x)$.

互斥(disjointness)假设:设 $\varphi_1(x)$ 和 $\varphi_2(x)$ 是两个互不相容事件, 则不存在同时是 $\varphi_1(x)$ 和 $\varphi_2(x)$ 的下层具体事件的事件 $\varphi_3(x)$.即: $\forall x \neg(\varphi_1(x) \vee \neg \varphi_2(x))$.

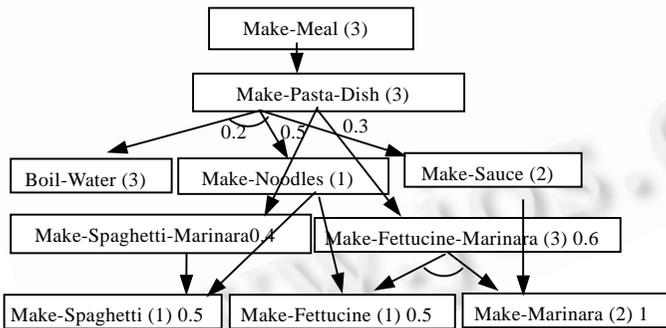
使用部件(component/use)假设:设 φ_{ij} 是一个事件, $\Psi_1 \dots \Psi_m$ 是把 φ_{ij} 作为其部件的所有规划.描述为 $\forall x \Psi_i(x) \supset \varphi_{i1}(f_{i1}(x)) \wedge \dots \wedge \varphi_{ij}(f_{ij}(x)) \wedge \dots \wedge \varphi_{in_i}(f_{in_i}(x)), i=1, \dots, m, n_i$ 为 Ψ_i 的部件个数, $\varphi_{i1} \dots \varphi_{in_i}$ 是 Ψ_i 的部件, $f_{i1} \dots f_{in_i}$ 分别为各部件在 Ψ_i 中的位置或映像函数, 那么 $\forall x \varphi_{ij}(x) \supset \text{END}(x) \vee \exists y_1, \Psi_1(y_1) \wedge (f_{1j}(y_1)=x) \vee \dots \vee \exists y_m, \Psi_m(y_m) \wedge (f_{mj}(y_m)=x)$.

以这 3 个假设为基础所建立的规划识别框架(如图 2 所示)是以层次结构的形式来组织的.不同的规划安置在结构中的不同位置.Kautz 提出的算法就是在这个结构上进行搜索,找出满足条件的规划集,然后再借助第 4 个假设最终确定规划识别的解集.

2 规划知识图

规划知识图是一个非循环的与或图.这个与或图由节点的集合组成.这些节点代表规划(事件).节点间均由连接符连接,连接符用来连接一个父节点和它的一组后继节点.每一个 K -连接符是从一个父节点指向一组共 K 个后继节点,用 K -连接符可以表示事件之间的整体与部分、具体与抽象的关系.这两种关系在规划识别中用两种表达式表示:

(1) 抽象表达式.所有: $\forall x \varphi_1(x) \supset \varphi(x), i=1 \dots n, \varphi, \varphi_1 \dots \varphi_n$ 是规划, $\varphi_1 \dots \varphi_n$ 抽象于 φ .在知识图中都表示为: $\varphi \varphi_1, \varphi$ 有一个 1-连接符指向后继节点 φ_i .例如图 3 中节点 Make-Meal 有一个 1-连接符指向后继节点 Make-Pasta-Dish, Make-Meal 与 Make-Pasta-Dish 之间是抽象与具体的关系.



做饭(3), 做面团(3), 烧水(3), 做面条(1), 做调味汁(2), 做大蒜番茄酱意大利面条 0.4, 做大蒜番茄酱白脱奶油面(3)0.6, 做意大利面条(1)0.5, 做白脱奶油面(1)0.5, 做大蒜番茄酱(2)1.

Fig.3 Changed to plan knowledge graph
图 3 转换为规划知识图

的后继是此父节点的具体化,即当此父节点与其后继节点之间是抽象与具体的关系时,这些具体化节点称为或节点.如果一个父节点的后继是一组组成部分节点,即当此父节点与其后继节点之间是整体与部分的关系时,这些组成部分节点称为与节点,在图中用圆弧标出.由上述方法把 Kautz 的 Make-Pasta-Dish 层次规划图改写成规划知识图(如图 3 所示).

知识图的规划识别算法通过计算各事件在现实生活中出现的可能性来判别候选规划.这种可能性值的计

(2) 分解表达式. $\forall x \varphi(x) \supset \varphi_1(f_1(x)) \wedge \dots \wedge \varphi_n(f_n(x)), i=1 \dots n, \varphi, \varphi_1 \dots \varphi_n$ 为规划, $f_1 \dots f_n$ 分别为各部件 φ_i 在整体规划 φ 中的位置或映像函数.在知识图中表示成: $\varphi \varphi_1, \varphi$ 有一个 n -连接符指向后继节点 $\varphi_1 \dots \varphi_n$.例如图 3 中节点 Make-Meal 有一个 3-连接符指向后继节点 {Boil-Water, Make-Noodles, Make-Sauce}, Make-Meal 与这 3 个后继节点之间是整体与部分的关系.

规划知识图中的节点可以分为两类:与节点和或节点.每一个节点是与节点还是或节点是相对于其父节点而言的,具体地说就是:如果一个父节点

算需要两种数据,其一,各种事件在现实生活中出现的可能性.Kautz 在他的规划识别中认为所有事件出现的可能性都是一样的,而我们知道,情况并非如此.假设一个人走进商店,只可能是去买东西或者是去抢东西,那么按照 Kautz 的方法,观察到一个人走进商店,则这两种事件都可能发生,换言之,出现的可能性是一样的.但实际情况是抢东西的可能性很小.为了使求得的结果更可信,我们在知识图中给出每个事件在现实生活中出现的可能性(或节点内数字).如果我们给出买东西出现的可能性是 0.99,抢东西的可能性是 0.01.当观察到一个人走进商店时,我们认为他去买东西这件事发生的可能性是 0.99,去抢东西这件事发生的可能性是 0.01.其二,一个事件的出现使另一个事件出现的可能性(支持程度).事件之间只有抽象与具体或整体与部分这两种关系,对这两种关系下的规划(事件)之间的支持程度作如下规定:

抽象与具体关系:由抽象表达式可知,具体类型规划的出现蕴涵抽象类型规划肯定出现,所以具体类型规划的出现对抽象类型规划出现的支持程度为 1(或节点与父节点间连接符上的数字,未标出).例如:观察到 Make-Pasta-Dish 可以确定观察到了 Make-Meal,因为 Make-Pasta-Dish 只是 Make-Meal 的一个具体形式.

整体与部分关系:规划的组成部件出现对规划出现的支持程度.如果一个规划是由 n 个组成部分规划组成的,那么这全部 n 个组成部分规划都出现,这个规划肯定出现.换言之,组成一个规划的所有组成部分规划的出现对此规划出现的支持程度的和为 1(与节点与父节点间连接符上的数字).例如:如果观察到 Boil-Water,Make-Noodles,Make-Sauce 这 3 个事件,则我们有理由相信 Make-Meal 肯定发生.

最后,知识图上节点内的整数代表时间片.我们用整数标记时间片的方法表示规划的各组成部分在规划中出现的先后顺序.例如,Make-Noodles 的时间片为 1,Boil-Water 的时间片为 3,说明 Make-Noodles 在 Boil-Water 之前发生.为每个事件规定一个时间片,是为了进行时序约束检查.

3 支持程度

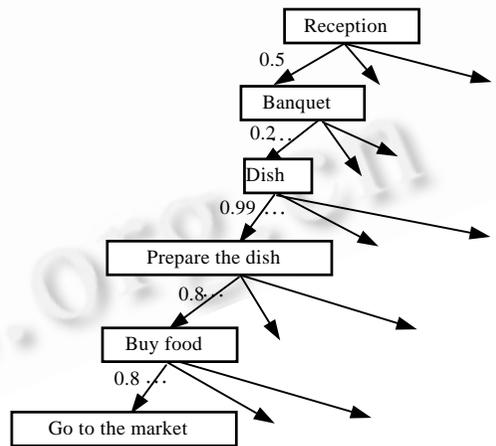
根据观察到的现象,可能会找到多个满足条件的规划集,但它们在现实世界中出现的可能性是不一样的.这主要是由于观察到的现象在满足条件的规划内的重要程度不同所致.由于重要程度不同,所以观察到的现象对规划出现可能性的支持程度也不同.例如图 4,去商店这个事件在买菜这个规划中很关键,所以当观察到去商店这个事件推断是买菜的可能性为 0.8,而同样是这个事件,由于在迎宾规划中无足轻重,所以当观察到去商店这个事件时,迎宾规划发生的可能性只有 0.06.利用支持程度的方法来估算各个规划出现的可能性就可以找到最可能出现的规划——最优解.在这里,支持程度是指一个规划(事件)的出现使另一个规划(事件)出现的可能性.为了使获得的结果更符合客观情况,我们在计算规划出现的可能性时规定:

- $P'(B/A)$ 代表规划 A 对规划 B 的支持程度,并且下面的等式成立:

(1) 规划 A_i 是规划 B 的组成部分规划,则 A_i 对 B 的支持程度 $P'(B/A_i) \leq 1$,规划 B 共有 m 个组成部分规划 $A_i, i=1, \dots, m$,则这 m 个组成部分规划一起对 B 的支持程度为 $P'(B/\bigwedge_{j=1}^m A_j) = \sum_{j=1}^m P'(B/A_j) = 1$;

(2) 具体规划 A 对它的抽象规划 B 的支持程度 $P'(B/A)=1$,即出现一个具体类型的规划也意味着出现其抽象类型的规划,因为具体规划是抽象规划的一种特例.例如,Make-Spaghetti-Marinara 是 Make-Pasta-Dish 的具体化,出现 Make-Spaghetti-Marinara 也就表示出现了 Make-Pasta-Dish.

- $P(A)$ 代表一个规划 A 出现的可能性;
- $P(B/A)$ 代表在规划 A 出现的条件下规划 B 出现的可能性;



接待, 宴会, 菜肴, 准备菜肴, 买菜, 去市场.

Fig.4 Structure of a large plan

图 4 一个庞大的规划

· $P''(B_i/\bigvee_{j=1}^n B_j)$ 表示在 n 个规划 $B_j, j=1, \dots, n$ 都有可能出现的情况下, B_i 出现的可能性. 规划知识图中规划出现的可能性根据下面的规定计算得出:

(1) 观察到的现象 A 出现的可能性 $P(A)=1$, 其他所有规划出现的可能性初始化为 0;

(2) 规划 A 是 n 个规划 $B_i, i=1, \dots, n$ 的组成部分规划, A 出现则所有 B_i 都有可能出现. 在这种情况下, $\sum_{i=1}^n P''(B_i/\bigvee_{j=1}^n B_j)=1$. 当 $n=1$ 时, $P''(B_i/\bigvee_{j=1}^n B_j)=1$;

(3) 规划 A 是 n 个规划 $B_i, i=1, \dots, n$ 的组成部分规划, 在 A 出现的条件下 B_i 出现的可能性,

$$P(B_i/A)=P(B_i)+P'(B_i/A)*P''(B_i/\bigvee_{j=1}^n B_j),$$

当 $n=1$ 时,

$$P(B_i/A)=P(B_i)+P'(B_i/A);$$

(4) 具体规划 A 的出现使抽象规划 B 出现的可能性为 $P(B/A)=P(B)+P(A)$;

(5) 如果识别出的多个规划无关, 则构成的规划集出现的可能性为各规划在观察下出现的可能性的乘积. 这个假设同 Kautz 的第 4 个假设——选择具有最少规划的规划集作为解集的目的是一样的, 都是希望获得的规划集中的规划尽可能的少.

4 算法

在 Kautz 层次结构图中定义了两个特殊的规划类型 ANY 和 END, 并且规定只有抽象于 END 的规划才具有独立意义, 以此来控制整个搜索算法. 但是对一个系统来说, 什么样的规划可以抽象于 END 却是一个棘手的问题. 例如如图 4, 对于一个用户来说, 做饭可能就是他要寻找的 END 了, 而对另一个用户来说, 这可能只是要寻找的一个庞大活动(迎宾)的一小部分, 一个部分规划. 而且如图 4 所示, 如果观察到一个人去商场就推断他是要迎宾客, 在实际生活中就过分牵强了, 如果认为他是去买菜或准备做饭可能更合理. 所以我们在知识图中没有用 ANY 和 END 这两个特殊的规划类型, 而是采用为每个规划提供一个支持程度的方法来确定可识别的候选规划. 通过这种方式, 我们可以把求解的过程控制在任何层次上, 从而找到可能性比较高的一个, 而不是最抽象或最庞大的一个规划.

知识图采用了与或图的宽度优先搜索策略, 并且一次性地产生整个解图. 知识图的解图 SG(solution graph) 是指由观察到的现象——规划知识图中的节点出发, 找到的一个包含这些节点的子图. 原来的规划知识图为 G , 那么解图 SG 如下规定:

(1) 观察到的现象节点 m 在 SG 中;

(2) n 在 SG 中, 如果 n 有一组 OR 后继节点, 那么有且只有一个 OR 后继节点在 SG 中;

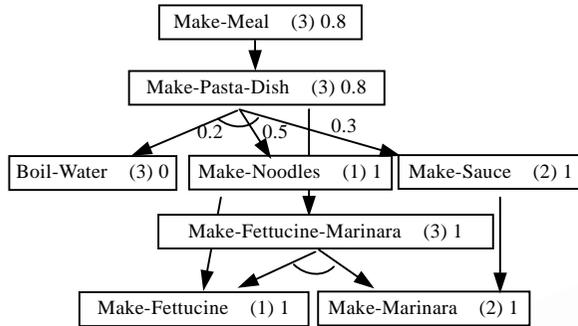
(3) n 在 SG 中, 如果 n 有一组 AND 后继节点, 则所有的 n 的 AND 后继节点都在 SG 中;

(4) 由观察到的现象节点 m 出发, 可能性最大的 OR 节点在 SG 中;

(5) 无其他的在 G 中的节点 m 在 SG 中.

采用这种方式获得的解图不但可以解释观察到的现象, 而且由于在知识图中用支持程度的方法确定一个规划出现的可能性更符合客观情况. 这种支持程度是根据实际情况给出的, 也可以随着实际情况的改变而改变, 所以用这种方式应该比用 Kautz 选择最少规划集的方法所确定的规划集更可信.

而且采用这种方式访问的节点更少, 效率更高. 我们举一个小例子, 在图 3 中, 如果观察到: Make-Fettucine (1)、Make-Marinara(2), 产生的解图如图 5 所示. 但如果采用 Kautz 的方法, 他会首先产生两个与图 5 非常相似的规划层次图, 其中一个不包含 Make-Fettucine 节点, 另一个不包含 Make-Marinara 节点. 然后再把这两个图组合成与图 5 一致的解图. 这样会产生将近一倍的冗余节点. 下面的情况其差别更明显, 观察: Make-Marinara (1)、Make-Fettucine(2), 这时由于不满足时序约束, 在第 1 步我们就不会产生 Make-Fettucine-Marinara 节点, 整个图 5 根本就不会产生, 而 Kautz 的方法却只能依旧如上面观察那样, 在最后进行合并时才能删除产生的所有节点. 如果我们观察到的现象非常多, 或者这些现象可能出现在多个规划中, 那么这种冗余就更多了.



做饭, 做面团, 烧水, 做面条, 做调味汁, 做大蒜
番茄酱白脱奶油, 做白脱奶油, 做大蒜番茄酱。

Fig.5 A solution of plan knowledge graph
图5 一个规划知识图的解

5 结 语

以上我们给出了用知识图来解决规划识别问题的方法.这种方法与 Kautz 方法相比,从形式上看:都是采用抽象和分解两种表达式来表示规划识别问题.但除此之外,Kautz 还引进了两个特殊的规划:END 和 ANY,这导致两种方法在规划图的构造上有很大的差别.在 Kautz 的规划层次结构图中抽象和分解两种表达式用两种方向和形状都不同的连接线表示,而且为了控制规划识别过程,他规定所有的规划都抽象于 ANY,所有可识别的规划都抽象于 END.这些特点都使得他的规划层次结构图的构造更加复杂.而在知识图中,这两种表达式用节点间的与或关系就能很明确地表示出来,所以节点间的连接线只需一种.由于与或图本身就有非常成熟的算法,也不需要 END 和 ANY 这些特殊的节点来控制算法的进程.总之,规划知识图与 Kautz 的规划层次结构图相比,结构更简单,表达更清楚.

从算法上看:知识图方法也能处理抽象继承和时序约束问题,两种方法所获得的结果是一样的.所不同的是,Kautz 的方法是先根据每个观察分别产生一个只满足它的子图,然后再对这些子图进行约束检查,以试图合并成一个解图.由于大多数观察到的现象都存在着或多或少的联系,这些子图内肯定存在着大量重复节点.采用与或图的宽度优先搜索方式,在知识图中一次性产生整个解图,而且相同的规划只会产生一个节点;另外,多个观察之间可能会存在破坏时序约束的情况,在 Kautz 的方法中,这种情况只能在最后合并的时候才能被发现;而知识图的宽度优先搜索方式在搜索的过程中就对观察到的所有现象进行约束检查,这样在算法的初期就能裁掉很多不满足条件的规划;在最后结果的处理上,Kautz 提出了一个假设——选择具有最少规划的规划集作为解集,而我们知道,最少并不意味着最可能发生,与或图的宽度优先搜索中采用支持程度的方法,这种方法获得的结果不但更合理,而且由于在搜索的过程中能直接求出一个规划出现的可能性,这使得那些可能性很小的规划在算法进行的过程中也会逐渐被裁掉.

总之,这种方法同 Kautz 的方法比较,不但求解的结果更合理,而且由于知识图的与或结构和宽度优先搜索方式,使得它的结构更简单、更清楚,求解过程更简捷,求解过程中产生的冗余节点更少.

References:

- [1] Henry, A., Kautz, A. Formal theory of plan recognition [Ph. D. Thesis]. Rochester: University of Rochester, 1987.
- [2] Charniak, E., Goldman, R.P. A Bayesian model of plan recognition. *Artificial Intelligence*, 1993,64(1):53~79.
- [3] Schmidt, C.F., Sridharan, N.S., Goodson, J.L. The plan recognition problem: an intersection of psychology and artificial intelligence. *Artificial Intelligence*, 1978,11(1):45~83.
- [4] Litman, D., Allen, J. A plan recognition model for subdialogues in conversation. *Cognitive Science*, 1987,11(1):163~200.
- [5] Karen, H., Victor, L. Knowledge-Based command understanding: an example for the software development environment. Technical Report TR 82-6, Department of Computer and Information Science, University of Massachusetts, Amherst, 1982.

- [6] Vilain, M. Getting serious about parsing plans: a grammatical analysis of plan recognition. In: Proceedings of the 8th National Conference on Artificial Intelligence. Boston: AAAI Press, 1990. 190~197.
- [7] Poole, D. Probabilistic horn abduction and Bayesian networks. Artificial Intelligence, 1993,64(2):81~129.
- [8] Allen, J.F., Koomen, J.A. Planning using a temporal world model. In: Amarel, S., ed. Proceedings of the IJCAI-83. Karlsruhe: Morgan Kaufmann Publishers, Inc., 1983. 741~747.

A Plan Recognition Algorithm Based on Plan Knowledge Graph*

JIANG Yun-fei¹, MA Ning²

¹(Computer Software Institute, Zhongshan University, Guangzhou 510275, China);

²(Department of Computer Science, Jilin University, Changchun 130023, China)

E-mail: lncsr05@zsu.edu.cn

<http://www.zsu.edu.cn>; <http://www.zsusoft.com>

Abstract: A new plan formalism method and a plan knowledge graph concept. Compared with Kautz's formalism used widely in plan recognition, this method is simpler and more direct. Because a concept of supporting degree is introduced into plan knowledge graph, the recognized plans are changed reasonably as new evidences collected. A plan recognition algorithm based on knowledge graph presented in this paper can change the plan recognition problem into the graph searching one. This method not only prompts efficiency but also gives the same result as Kautz's.

Key words: plan recognition; event; temporal constraint

* Received May 27, 2000; accepted February 16, 2001

Supported by the National Natural Science Foundation of China under Grant Nos.69873047, 60173039; the Natural Science Foundation of Guangdong Province of China under Grant No.980260