

关于 CTL 与 $E_A G_{LE}$ 两种规划扩展目标 表示语言的语义比较

黄巍¹⁾ 姜云飞²⁾ 文中华³⁾ 彭宏¹⁾

¹⁾(华南理工大学计算机科学与工程学院 广州 510641)

²⁾(中山大学软件研究所 广州 510275)

³⁾(湘潭大学信息工程学院 湖南 湘潭 411105)

摘 要 在不确定的智能规划领域中,CTL 和 $E_A G_{LE}$ 是两种重要的扩展目标表示语言.虽然与 CTL 相比 $E_A G_{LE}$ 具有可以表示规划意图和失败处理机制的特点,但是有关严格比较这两种目标表示语言语义的研究工作还不多.文章在规划的执行结构这一语义层次上对这两种语言做了严格的比较,证明了对于许多包括原来曾被认为无法用 CTL 表示的 $E_A G_{LE}$ 规划目标而言,都存在着一个与之语义等价的 CTL 规划目标,并且进一步分析了这两种语言在表示规划目标和指导规划求解这两个层次上的优缺点.

关键词 不确定的智能规划;扩展的规划目标;执行结构;CTL; $E_A G_{LE}$

中图法分类号 TP18 **DOI 号:** 10.3724/SP.J.1016.2009.00086

A Comparison of Semantics of Extended Goals Expressed in CTL and $E_A G_{LE}$

HUANG Wei¹⁾ JIANG Yun-Fei²⁾ WEN Zhong-Hua³⁾ PENG Hong¹⁾

¹⁾(School of Computer Science & Engineering, South China University of Technology, Guangzhou 510641)

²⁾(Software Research Institute, Sun Yat-Sen University, Guangzhou 510275)

³⁾(College of Information Engineering, Xiangtan University, Xiangtan, Hunan 411105)

Abstract In automated planning under uncertainty, extended goals are often expressed as temporal logic formulas, particularly as CTL (computation tree logic) formulas and $E_A G_{LE}$ formulas. Though it is said that the capability of representing the “intentional” aspects of goals and the possibility of dealing with failure are the main new features of $E_A G_{LE}$ with respect to CTL, there is not enough work devoted to formal comparison of semantics of these two languages. According to the formal semantics of $E_A G_{LE}$ and CTL goals on Kripke structures (i. e., execution structures induced by plans from initial states), it is proven that many $E_A G_{LE}$ goals including some having been deemed to be unexpressive in CTL (e. g., some $E_A G_{LE}$ goals in which TryReach operator and Fail operator appear) can be expressed in CTL in fact, and an analysis of the characters of these two languages in expressing extended goals and guiding the search for plans that satisfy them is given.

Keywords automated planning under uncertainty; extended goal; execution structure; CTL (computation tree logic); $E_A G_{LE}$

收稿日期:2007-12-05;最终修改稿收到日期:2008-10-20. 本课题得到国家自然科学基金(60773201)、广东省自然科学基金(07006474)和广东省科技攻关项目基金(2007B01020044)资助. 黄巍,男,1979年生,博士,主要研究方向为智能规划、自动推理和基于模型诊断. E-mail: huangbodao@yahoo.com.cn. 姜云飞,男,1945年生,硕士,教授,博士生导师,主要研究领域为智能规划、自动推理和基于模型诊断等. 文中华,男,1966年生,博士,副教授,主要研究方向为智能规划和图论. 彭宏,男,1956年生,博士,教授,博士生导师,主要研究领域为数据挖掘和智能网络技术.

1 引言

与经典规划中的可达性目标不同, 扩展的规划目标往往包含了对规划执行过程的时态要求, 所以它们往往被表示成时态逻辑公式^[1-2]. 而在不确定的规划领域中, 具有分支计算结构的 CTL^[3] 和 $E_A G_{LE}$ ^[4] 是目前两种重要的扩展目标表示语言.

CTL 由于已被广泛应用于模型检测的领域, 所以它也在基于模型检测的规划方法^[5-7] 中被用于表示扩展的规划目标; 而 $E_A G_{LE}$ 是一种专门描述扩展的规划目标的语言, 它长于表示规划者的规划意图 (如其语法中的 TryReach 和 DoReach 算子分别表示“尽量试着到达”和“必须保证到达”, 而 TryMaint 和 DoMaint 算子则分别表示“尽量保持”和“必须保持”) 以及失败处理机制 (如其语法中的 Fail 算子表示“若…失败则…”)^[4,7-8].

如何选择合适的规划目标表示语言并将目标公式转化为用于指导搜索规划解的控制自动机是一个很重要的问题. 因为目标语言的表达能力决定了对规划解的搜索是否可能, 而由目标公式产生的控制自动机的状态数目部分决定了整个搜索空间的大小^[4-5]. 虽然在文献[4]和文献[7]中, 一些用 $E_A G_{LE}$ 公式 (例如一些含有 TryReach 算子和 Fail 算子的 $E_A G_{LE}$ 公式) 表示的扩展目标被认为无法用 CTL 表示, 但是至今有关 CTL 和 $E_A G_{LE}$ 在语义层次上比较的研究还很少. 本文在规划的执行结构这一语义层次上对这两种语言做了严格的比较, 证明了对于许多 $E_A G_{LE}$ 目标公式 (包括了文献[4]和文献[7]所列举的所有不可被 CTL 表示的 $E_A G_{LE}$ 目标公式) 而言, 都存在着一个与之语义等价的 CTL 目标公式. 并进一步分析指出了 $E_A G_{LE}$ 与 CTL 相比在表示规划目标时更简洁, 相应的控制自动机构造过程更简单, 但是它无法表示一些基本的 CTL 语义特征.

本文第 2 节介绍有关不确定规划的基本概念; 第 3 节则是 CTL 和 $E_A G_{LE}$ 目标公式严格的语法和语义定义; 第 4 节即本文的核心是 CTL 和 $E_A G_{LE}$ 之间的语义比较; 最后的第 5 节则给出了结论以及下一步的研究展望.

2 不确定的规划领域与带有上下文信息的规划

所谓规划领域是指现实动态系统的抽象模型.

在本文中, 规划领域是指一个不确定的状态转换系统, 它由系统的状态集合、动作集合以及状态转换函数构成. 其中, 各系统状态可以用命题公式描述, 状态转换函数则描述了如何通过执行某个动作而使系统从一个状态不确定地转换到若干个不同的可能状态.

定义 1 (规划领域). 一个规划领域 \mathcal{D} 是一个四元组 $\langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$, 其中, \mathcal{P} 是一个有限非空的原子命题集合, $\mathcal{S} \subseteq 2^{\mathcal{P}}$ 是系统的状态集合, \mathcal{A} 是一个有限的动作集合, 而 $\mathcal{R}: \mathcal{S} \times \mathcal{A} \rightarrow 2^{\mathcal{S}}$ 是一个状态转换函数.

在后面的内容里, $\mathcal{F}(\mathcal{P})$ 表示建立在原子命题集 \mathcal{P} 上的命题公式集合.

对扩展目标进行规划时, 规划的目的已不再仅仅限于令系统能够从初始状态到达目标状态 (即可达性目标), 当中还可能包括了对系统的状态转换历史的具有时态性质和强度差别的要求. 例如, 要求一个可移动的机器人尽量试着到达某一指定位置并要求它永远避开那些危险的区域; 又或者, 要求机器人不断地在位置 A 与位置 B (如果可能的话) 之间来回移动. 像上述这些扩展目标往往被表示成时态逻辑公式, 特别是 CTL 公式或 $E_A G_{LE}$ 公式.

为了满足扩展的规划目标, 规划中须包含上下文信息 (即应考虑系统的状态转换历史).

定义 2 (规划). 对于一个规划领域 $\mathcal{D} = \langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ 而言, 它的一个规划 π 是一个四元组 $\langle \mathcal{C}, c_0, act, ctxt \rangle$, 其中, \mathcal{C} 是一个上下文集合, $c_0 \in \mathcal{C}$ 是 \mathcal{D} 的初始上下文, $act: \mathcal{S} \times \mathcal{C} \rightarrow \mathcal{A}$ 为动作选择函数, 而 $ctxt: \mathcal{S} \times \mathcal{C} \times \mathcal{S} \rightarrow \mathcal{C}$ 为上下文更新函数.

在定义 2 中, act 和 $ctxt$ 的含义是: 若 \mathcal{D} 处于状态 s 下并且当前上下文为 c , 那么 $act(s, c)$ 将返回此时规划 π 所指定的待执行动作, 而 $ctxt(s, c, s')$ 将为某一可能的下一状态 s' 指定其相应的上下文. act 和 $ctxt$ 都是部分函数, 因为某些状态——上下文的组合在规划的执行过程中也许不可能出现. 如果当 $act(s, c) = a$ 且 $ctxt(s, c, s') = c'$ 时都有 $s' \in \mathcal{R}(s, a)$, 那么称 π 是可执行的; 如果当 $act(s, c) = a$ 且 $s' \in \mathcal{R}(s, a)$ 时都有 $(\exists c' \in \mathcal{C})(ctxt(s, c, s') = c')$, 那么称 π 是完备的. 在后面内容里, 我们所讨论的规划都是可执行的且完备的.

其实, 一个带有上下文信息的规划 π 的执行过程就是一个当前状态——上下文序对不断转换的过程. 我们可以用下面定义的执行结构来表示 π 的所有可能的执行情况.

定义 3(执行结构). 设 $\pi = \langle \mathcal{C}, c_0, act, ctxt \rangle$ 是规划领域 $\mathcal{D} = \langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ 的一个规划, $\emptyset \subset I \subseteq \mathcal{S}$ 为 \mathcal{D} 初始的可能状态的集合. 从 I 所导出的 π 的执行结构为 $K = \langle \mathcal{Q}, T, L \rangle$, 其中 $\mathcal{Q} \subseteq \mathcal{S} \times \mathcal{C}$ 和 $T \subseteq \mathcal{Q} \times \mathcal{Q}$ 是满足以下条件的最小集合:

(1) 若 $s \in I$, 那么 $\langle s, c_0 \rangle \in \mathcal{Q}$;

(2) 若 $\langle s, c \rangle \in \mathcal{Q}$, $act(s, c) = a$, $s' \in \mathcal{R}(s, a)$ 且 $ctxt(s, c, s') = c'$, 则 $\langle s', c' \rangle \in \mathcal{Q}$ 且 $(\langle s, c \rangle, \langle s', c' \rangle) \in T$. 另外, 函数 $L: \mathcal{S} \times \mathcal{C} \rightarrow \mathcal{S}$ 的定义为 $L(s, c) = s$. $q \in \mathcal{Q}$ 是 K 的一个终止状态——上下文序对当且仅当不存在 $q' \in \mathcal{Q}$ 使得 $(q, q') \in T$.

直观地讲, 执行结构 K 就是一个有向图, 其中的节点集 \mathcal{Q} 是系统在以 I 为初始状态集执行规划 π 时所可能到达的所有状态——上下文序对的集合, 而 T 则表示了相应的所有可能的状态——上下文序对的转换. K 的各个终止状态——上下文序对则代表了规划执行的终止. 其实, 定义 3 里的执行结构就是一个 Kripke 结构^[8].

一个执行结构描述了执行一个给定的规划所可能出现的各种情况. 而执行结构中的一条执行路径则表示了相应规划的一种可能的执行历史情况, 它是执行结构中的一个状态——上下文序对序列.

定义 4(执行路径). 设 $K = \langle \mathcal{Q}, T, L \rangle$ 是规划领域 \mathcal{D} 的规划 $\pi = \langle \mathcal{C}, c_0, act, ctxt \rangle$ 从初始状态集 I 所导出的执行结构. K 里的一条从 $q_0 \in \mathcal{Q}$ 出发的执行路径是一个状态——上下文序对序列 $\{q_0, q_1, \dots\}$, 其中各 q_i 都属于 \mathcal{Q} , 而且

(1) 若 q_i 是该序列的最后一个状态——上下文序对, 那么 q_i 是 K 里的一个终止状态——上下文序对; 否则

(2) $(q_i, q_{i+1}) \in T$.

从上面的定义里可以发现执行路径既可以是有限的也可以无限的. 若一个状态——上下文序对序列是某个执行路径的有限子序列, 那么我们称之为相应执行结构中的一条路径. 若有一条从 q 到 q' 的路径, 那么就称 q' 是 q 可达的.

为了便于后面的叙述, 这里规定了一些有关执行路径和路径的标记方法. ϵ 表示空路径. 设 K 是一个执行结构, 那么 $HISTORIESOF(K)$ 和 $PATHSOF(K)$ 分别表示 K 的执行路径集合与路径集合. 若 $\rho, \sigma \in PATHSOF(K)$ 且 $\rho \neq \epsilon$, 则 $\rho \geq \sigma$ (或 $\sigma \leq \rho$) 表示 σ 是 ρ 的一个前缀; $\rho > \sigma$ (或 $\sigma < \rho$) 则表示 $\rho \geq \sigma$ 且 $\sigma \neq \rho$; $LAST(\rho)$ 表示 ρ 的最终状态——上下

文序对; $|\rho|$ 表示在 ρ 中出现的状态——上下文序对的数目; 若 $\sigma \neq \epsilon$ 且 $LAST(\rho) = FIRST(\sigma)$, 则 $\rho \cdot \sigma$ 表示 ρ 与 σ 连接之后所得到的路径 (例如 $\{q_0, q_1\} \cdot \{q_1, q_2\} = \{q_0, q_1, q_2\}$). 若 $\rho \in PATHSOF(K) \cup HISTORIESOF(K)$, 则 $FIRST(\rho)$ 表示 ρ 的最初状态——上下文序对; $q \in \rho$ 表示状态——上下文序对 q 在 ρ 中出现. 若 $FPaths \subseteq PATHSOF(K)$, 那么 $FPaths$ 的极小路径集合为 $\min(FPaths) = \{\sigma \in FPaths \mid (\forall \sigma' \in PATHSOF(K)) ((\sigma' < \sigma) \rightarrow (\sigma' \notin FPaths))\}$.

3 CTL 和 $E_A G_{LE}$ 扩展目标的语法和语义

在动作效果不确定的规划领域里, 具有分支计算结构的时态逻辑公式是扩展目标的一种主要表示形式. 而这其中, 又以 CTL 和 $E_A G_{LE}$ 公式为主, 它们的语法如下面的定义 5 所示.

定义 5(CTL 和 $E_A G_{LE}$ 目标公式的语法). 设 \mathcal{P} 是一个有限的非空原子命题集合, b 表示 \mathcal{P} 中的任一元素. 那么建立在 \mathcal{P} 上的 CTL 目标公式集 $\mathcal{F}_C(\mathcal{P})$ 里的任意一个公式 g 应满足以下语法:

$$g ::= T \mid \perp \mid b \mid \neg b \mid g \wedge g \mid g \vee g \mid AX \ g \mid EX \ g \mid$$

$$A \ (g \ U \ g) \mid E \ (g \ U \ g) \mid A \ (g \ W \ g) \mid E \ (g \ W \ g).$$

设 g 为 $\mathcal{F}_C(\mathcal{P})$ 里的任意一个公式, 则我们可以分别用缩写 $AF \ g$ 和 $EF \ g$ 来表示公式 $A \ (T \ U \ g)$ 和 $E \ (T \ U \ g)$, 分别用缩写 $AG \ g$ 和 $EG \ g$ 来表示公式 $A \ (g \ W \ \perp)$ 和 $E \ (g \ W \ \perp)$.

另外, 建立在 \mathcal{P} 上的 $E_A G_{LE}$ 目标公式集 $\mathcal{F}_E(\mathcal{P})$ 里的任意一个公式 g 应满足以下语法:

$$h ::= T \mid \perp \mid b \mid \neg b \mid h \wedge h \mid h \vee h$$

$$g ::= h \mid DoReach \ h \mid TryReach \ h \mid DoMaint \ h \mid$$

$$TryMaint \ h \mid g \ Then \ g \mid g \ Fail \ g \mid$$

$$g \ And \ g \mid Repeat \ g.$$

在上面的定义里, CTL 公式中的 A 和 E 分别为全称路径量词和存在路径量词, 而时态算子 X, U 和 W 则分别表示“下一状态”、“(强)总是…直到…”和“(弱)总是…直到…”. 与 CTL 公式相比, $E_A G_{LE}$ 公式更注重表示规划的“意图”和失败处理机制, 如其中的 $DoReach$ 、 $TryReach$ 和 $Fail$ 算子.

在规划的执行结构上, CTL 目标公式和 $E_A G_{LE}$ 目标公式的语义形式也不尽相同. 前者体现为执行结构中满足目标公式的状态——上下文序对的集

合,而后者则体现为各个状态——上下文序对所分别对应的成功路径集合和失败路径集合。

定义 6(CTL 目标公式的语义). 设 $K = \langle Q, T, L \rangle$ 是由规划领域 $\mathcal{D} = \langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ 的某个规划所导出的执行结构. 设 $q \in Q$, 而 $g, g' \in \mathcal{F}_C(\mathcal{P})$. 若公式 g 在 q 上成立, 则记为 $K, q \models g$, 其含义可以按 $\mathcal{F}_C(\mathcal{P})$ 的语法递归地给出:

- (1) $K, q \models T$, 并且 $K, q \not\models \perp$;
- (2) 若 $g \in \mathcal{P}$, 那么 $K, q \models g$ 当且仅当 $g \in L(q)$ 而 $K, q \models \neg g$ 当且仅当 $g \notin L(q)$;
- (3) $K, q \models g \wedge g'$ 当且仅当: $K, q \models g$ 并且 $K, q \models g'$;
- (4) $K, q \models g \vee g'$ 当且仅当: $K, q \models g$ 或者 $K, q \models g'$;
- (5) $K, q \models AX \ g$ 当且仅当: $(\forall \sigma = \langle q_0, q_1 \rangle \in PATHSOF(K))((q_0 = q) \rightarrow (K, q_1 \models g))$;
- (6) $K, q \models EX \ g$ 当且仅当: $(\exists \sigma = \langle q_0, q_1 \rangle \in PATHSOF(K))((q_0 = q) \wedge (K, q_1 \models g))$;
- (7) $K, q \models A \ (g \ U \ g')$ 当且仅当: $(\forall \sigma = \langle q_0, q_1, \dots \rangle \in HISTORIESOF(K))((q_0 = q) \rightarrow (\exists i \geq 0) (K, q_i \models g') \wedge (i > \forall j \geq 0) (K, q_j \models g))$;
- (8) $K, q \models E \ (g \ U \ g')$ 当且仅当: $(\exists \sigma = \langle q_0, q_1, \dots \rangle \in HISTORIESOF(K))((q_0 = q) \wedge (\exists i \geq 0) (K, q_i \models g') \wedge (i > \forall j \geq 0) (K, q_j \models g))$;
- (9) $K, q \models A \ (g \ W \ g')$ 当且仅当: $(\forall \sigma = \langle q_0, q_1, \dots \rangle \in HISTORIESOF(K))((q_0 = q) \rightarrow ((\exists i \geq 0) (K, q_i \models g') \wedge (i > \forall j \geq 0) (K, q_j \models g)) \vee ((\forall k \geq 0) (K, q_k \models g)))$;
- (10) $K, q \models E \ (g \ W \ g')$ 当且仅当: $(\exists \sigma = \langle q_0, q_1, \dots \rangle \in HISTORIESOF(K))((q_0 = q) \wedge (((\exists i \geq 0) (K, q_i \models g') \wedge (i > \forall j \geq 0) (K, q_j \models g)) \vee ((\forall k \geq 0) (K, q_k \models g))))$.

另外,为了书写方便,在后面的内容里:若 $g \in \mathcal{F}(\mathcal{P})$, 则将 $K, q \models g$ 简写成 $q \models g$, 而 $q \not\models g$ 则记为 $q \models \neg g$.

定义 7($E_A G_{LE}$ 目标公式的语义). 设 $K = \langle Q, T, L \rangle$ 是由规划领域 $\mathcal{D} = \langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ 的某个规划所导出的执行结构. 设 $q \in Q, g, g', g_1, g_2 \in \mathcal{F}_E(\mathcal{P})$, 而 $h \in \mathcal{F}(\mathcal{P})$. 那么

- (1) 若 $g = h$, 则如果 $q \models h$, 那么 $S_g(q) = \{\langle q \rangle\}$ 且 $F_g(q) = \emptyset$; 否则 $S_g(q) = \emptyset$ 且 $F_g(q) = \{\langle q \rangle\}$.
- (2) 若 $g = \text{DoMaint } h$, 则 $S_g(q) = \emptyset$; 若对于所有 q 可达的状态——上下文序对 q' 都有 $q' \models h$, 那

么 $F_g(q) = \emptyset$, 否则 $F_g(q) = \{\langle q \rangle\}$.

(3) 若 $g = \text{TryMaint } h$, 则 $S_g(q) = \emptyset, F_g(q) = \min(\{\rho \in PATHSOF(K) \mid (FIRST(\rho) = q) \wedge (LAST(\rho) \models h)\})$.

(4) 若 $g = \text{DoReach } h$, 则 (令 $q_0 = q$): 如果 $(\exists \sigma = \langle q_0, q_1, \dots \rangle \in HISTORIESOF(K))(\forall i \geq 0) (q_i \models h)$, 那么 $S_g(q_0) = \emptyset$ 且 $F_g(q_0) = \{\langle q_0 \rangle\}$; 否则 $S_g(q_0) = \min(\{\rho \in PATHSOF(K) \mid (FIRST(\rho) = q_0 \wedge (LAST(\rho) \models h))\})$ 且 $F_g(q_0) = \emptyset$.

(5) 若 $g = \text{TryReach } h$, 则 $S_g(q) = \min(\{\rho \in PATHSOF(K) \mid (FIRST(\rho) = q) \wedge (LAST(\rho) \models h)\})$, 而 $F_g(q) = \min(\{\rho \in PATHSOF(K) \mid (FIRST(\rho) = q) \wedge (\forall q' \in \rho) (q' \not\models h) \wedge (\forall \rho' \in PATHSOF(K))((\rho' \geq \rho) \rightarrow (LAST(\rho') \not\models h))\})$.

(6) 若 $g = g_1 \text{ Fail } g_2$, 则 $S_g(q) = S_{g_1}(q) \cup \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in F_{g_1}(q)) \wedge (\sigma_2 \in S_{g_2}(LAST(\sigma_1)))\}$, $F_g(q) = \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in F_{g_1}(q)) \wedge (\sigma_2 \in F_{g_2}(LAST(\sigma_1)))\}$.

(7) 若 $g = g_1 \text{ Then } g_2$, 则 $S_g(q) = \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in S_{g_1}(q)) \wedge (\sigma_2 \in S_{g_2}(LAST(\sigma_1)))\}$, $F_g(q) = F_{g_1}(q) \cup \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in S_{g_1}(q)) \wedge (\sigma_2 \in F_{g_2}(LAST(\sigma_1)))\}$.

(8) 若 $g = g_1 \text{ And } g_2$, 则 $S_g(q) = \min(\{\rho \in PATHSOF(K) \mid (\exists \rho_1 \leq \rho) (\rho_1 \in S_{g_1}(q)) \wedge (\exists \rho_2 \leq \rho) (\rho_2 \in S_{g_2}(q))\})$, $F_g(q) = \min(F_{g_1}(q) \cup F_{g_2}(q))$.

(9) 若 $g = \text{Repeat } g'$, 则 $S_g(q) = \emptyset, F_g(q) = \{\sigma_0 \cdot \sigma_1 \cdot \dots \cdot \sigma_n \cdot \sigma' \mid (\sigma' \in F_{g'}(FIRST(\sigma'))) \wedge (n \geq \forall i \geq 0) (\exists \sigma'_i \in S_{g'}(FIRST(\sigma'_i))) (\sigma_i = \sigma'_i \cdot \langle LAST(\sigma'_i), LAST(\sigma_i) \rangle)\}$.

现在,我们可以定义何为满足某 CTL 或 $E_A G_{LE}$ 目标公式的规划了.

定义 8(满足 CTL 或 $E_A G_{LE}$ 目标公式的规划).

设 $\pi = \langle \mathcal{C}, c_0, act, ctxt \rangle$ 是规划领域 $\mathcal{D} = \langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ 的一个规划, 而 $K = \langle Q, T, L \rangle$ 是 π 从初始状态集 I 所导出的执行结构. 那么

- (1) 若 $g \in \mathcal{F}_C(\mathcal{P})$ 且 $(\forall s_0 \in I) (K, \langle s_0, c_0 \rangle \models g)$, 则称 π 在 I 上满足 g , 并记为 $\pi, I \models_C g$;
- (2) 若 $g \in \mathcal{F}_E(\mathcal{P})$ 且 $(\forall s_0 \in I) (F_g(\langle s_0, c_0 \rangle) = \emptyset)$, 则称 π 在 I 上满足 g , 并记为 $\pi, I \models_E g$.

有关满足某些 CTL 或 $E_A G_{LE}$ 目标公式的规划的例子可以参考文献[4-5].

4 CTL 目标与 $E_A G_{LE}$ 目标之间的语义比较

根据定义 8, 我们可以发现: 一个规划在某个状

态集上是否满足某 CTL 或 $E_A G_{LE}$ 目标公式都取决于它所导出的执行结构 K . 所以, 我们可以通过 K 来定义 CTL 和 $E_A G_{LE}$ 目标公式间的语义等价关系.

为了叙述方便, 在后面的内容里, 如无特别说明, 则 $\mathcal{D} = \langle \mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{R} \rangle$ 表示一个任意的规划领域, $\pi = \langle \mathcal{C}, c_0, act, ctxt \rangle$ 和 I 为 \mathcal{D} 任意的一个规划和初始状态集, q_0 为任意的一个初始状态——上下文序对 (即 $q_0 \in I \times \{c_0\}$), $K = \langle Q, T, L \rangle$ 为 π 从 I 所导出的执行结构, 而 h 表示一个任意的 $\mathcal{F}(\mathcal{P})$ 内的命题公式.

定义 9 (CTL 和 $E_A G_{LE}$ 目标公式的语义等价). 设 $g_1, g_2 \in \mathcal{F}_C(\mathcal{P}) \cup \mathcal{F}_E(\mathcal{P})$. 若 π 在 I 上满足 g_1 当且仅当 π 在 I 上满足 g_2 , 则称 g_1 和 g_2 所表示的扩展目标是语义等价的, 并记 $g_1 \equiv_{(C,E)} g_2$.

两个目标公式语义等价的直观含义是满足它们的规划集合总是相同.

从定义 8 和定义 9 不难发现, $g_1 \equiv_{(C,E)} g_2$ 当且仅当

- (1) $(K, q_0 \models g_1) \Leftrightarrow (F_{g_2}(q_0) = \emptyset)$, 其中 $g_1 \in \mathcal{F}_C(\mathcal{P})$ 且 $g_2 \in \mathcal{F}_E(\mathcal{P})$; 或者
- (2) $(K, q_0 \models g_1) \Leftrightarrow (K, q_0 \models g_2)$, 其中 $g_1, g_2 \in \mathcal{F}_C(\mathcal{P})$; 或者
- (3) $(F_{g_1}(q_0) = \emptyset) \Leftrightarrow (F_{g_2}(q_0) = \emptyset)$, 其中 $g_1, g_2 \in \mathcal{F}_E(\mathcal{P})$.

本节余下的内容会根据 CTL 和 $E_A G_{LE}$ 目标公式的语义, 证明对于许多不含 And 和 Repeat 算子的 $E_A G_{LE}$ 目标公式而言, 都存在着一个与之语义等价的 CTL 目标公式; 然后进一步分析指出 $E_A G_{LE}$ 与 CTL 相比在表达扩展目标方面的一些优缺点. 为了方便叙述, 在后面的内容里, 如无特别说明则所讨论的任意 $E_A G_{LE}$ 目标公式 $g \in \mathcal{F}_E(\mathcal{P})$ 都是不含 And 和 Repeat 算子的.

4.1 用 CTL 目标公式语义等价表示的 $E_A G_{LE}$ 目标公式

根据 CTL 和 $E_A G_{LE}$ 公式的语法 (定义 5), 不难发现: 若 g 既属于 $\mathcal{F}_C(\mathcal{P})$ 也属于 $\mathcal{F}_E(\mathcal{P})$, 那么 g 必然是一个命题公式 (即 $g \in \mathcal{F}(\mathcal{P})$).

定理 1. 若 $g \in \mathcal{F}_C(\mathcal{P}) \cap \mathcal{F}_E(\mathcal{P})$, 则 $g \equiv_{(C,E)} g$.

证明. 因为根据定义 7 有 $F_g(q_0) = \emptyset \Leftrightarrow q_0 \models g$, 所以由定义 8 和定义 9, 得 $g \equiv_{(C,E)} g$. 证毕.

定理 1 说明无论将命题公式 g 看作是 CTL 公式还是 $E_A G_{LE}$ 公式, 其语义皆不变.

在定义 7 中, 求极小路径集的操作 \min (其含义

见第 2 节末) 经常被用到. 有关它的辅助定理 1 将在后面的证明中常被用到.

辅助定理 1. 设 $FPaths \subseteq PATHSOF(K)$. $\min(FPaths) = \emptyset$ 当且仅当 $FPaths = \emptyset$.

证明.

(1) 必要性. $FPaths = \emptyset$

$$\Rightarrow \{\sigma \in FPaths \mid (\forall \sigma' \in PATHSOF(K)) ((\sigma' < \sigma) \rightarrow (\sigma' \notin FPaths))\} = \emptyset$$

\Rightarrow (min 的含义, 见第 2 节末) $\min(FPaths) = \emptyset$.

(2) 充分性 (用反证法, 假设 $FPaths \neq \emptyset$). 令 $\min(FPaths) = \emptyset$,

$$\begin{aligned} & FPaths \subseteq PATHSOF(K) \text{ 且 } FPaths \neq \emptyset \\ & \Rightarrow (\exists \sigma \in FPaths) (\forall \sigma' \in FPaths) (|\sigma| \leq |\sigma'|) \\ & \Rightarrow (\text{min 的含义, 见第 2 节末}) (\exists \sigma \in FPaths) (\sigma \in \min(FPaths)). \end{aligned}$$

这与 $\min(FPaths) = \emptyset$ 矛盾.

所以原假设不成立, 即 $FPaths = \emptyset$.

由 (1) 和 (2), 辅助定理 1 得证. 证毕.

下面的定理 2 说明了 $\mathcal{F}_E(\mathcal{P})$ 中所有只含有 DoMaint 和 TryMaint 算子 (即持续性目标算子) 的公式都可以被语义等价地表示成 CTL 目标公式.

定理 2. 设 $g_1 = \text{DoMaint } h, g_2 = \text{TryMaint } h$. $g_1 \equiv_{(C,E)} AG h \equiv_{(C,E)} g_2$.

证明.

(1) 先证 $g_1 \equiv_{(C,E)} AG h$.

因为 $F_{g_1}(q_0) = \emptyset$

$$\begin{aligned} & \Leftrightarrow (\text{定义 7}) (\forall q \in Q) ((q \text{ 是 } q_0 \text{ 可达的}) \rightarrow (q \models h)) \\ & \Leftrightarrow (\forall \sigma \in HISTORIESOF(K)) ((FIRST(\sigma) = q_0) \rightarrow ((\forall q \in \sigma) (q \models h))) \end{aligned}$$

$$\Leftrightarrow (\text{定义 6}) K, q_0 \models AG h$$

所以 $g_1 \equiv_{(C,E)} AG h$.

(2) 再证 $g_2 \equiv_{(C,E)} AG h$.

因为 $F_{g_2}(q_0) = \emptyset$

$$\begin{aligned} & \Leftrightarrow (\text{定义 7}) \min(\{\rho \in PATHSOF(K) \mid \\ & \quad (FIRST(\rho) = q_0) \wedge (LAST(\rho) \not\models h)\}) = \emptyset \\ & \Leftrightarrow (\text{辅助定理 1}) \{\rho \in PATHSOF(K) \mid \\ & \quad (FIRST(\rho) = q_0) \wedge (LAST(\rho) \not\models h)\} = \emptyset \\ & \Leftrightarrow (\forall \sigma \in HISTORIESOF(K)) ((FIRST(\sigma) = q_0) \rightarrow ((\forall q \in \sigma) (q \models h))) \end{aligned}$$

$$\Leftrightarrow (\text{定义 6}) K, q_0 \models AG h.$$

所以 $g_2 \equiv_{(C,E)} AG h$.

由 (1) 和 (2), 定理 2 得证. 证毕.

虽然定理 2 指出了单独的 DoMaint h 和

TryMaint h 实际上是语义等价的,但这并不表示当它们与其它 $E_A G_{LE}$ 算子结合使用时其语义仍会等价(例如见后面的定理 5).

下面的定理 3 则说明了 $\mathcal{F}_E(\mathcal{P})$ 中所有只含有 DoReach 和 TryReach 算子(即可达性目标算子)的公式也都可以被语义等价地表示成 CTL 目标公式.

定理 3. 设 $g1 = \text{DoReach } h$, $g2 = \text{TryReach } h$.
 $g1 \equiv_{(C,E)} \text{AF } h$, $g2 \equiv_{(C,E)} A((\text{EF } h) \text{ W } h)$.

证明.

(1) 先证 $g1 \equiv_{(C,E)} \text{AF } h$.

因为 $F_{g1}(q_0) = \emptyset$

$\Leftrightarrow (\text{定义 } 7) \rightarrow (\exists \sigma \in \text{HISTORIESOF}(K))$

$((\text{FIRST}(\sigma) = q_0) \wedge (\forall q \in \sigma)(q \models h))$

$\Leftrightarrow (\forall \sigma \in \text{HISTORIESOF}(K))((\text{FIRST}(\sigma) = q_0) \rightarrow (\exists q \in \sigma)(q \models h))$

$\Leftrightarrow (\text{定义 } 6) K, q_0 \models \text{AF } h$

所以 $g1 \equiv_{(C,E)} \text{AF } h$.

(2) 再证 $g2 \equiv_{(C,E)} A((\text{EF } h) \text{ W } h)$.

因为 $F_{g2}(q_0) = \emptyset$

$\Leftrightarrow (\text{定义 } 7) \min(\{\rho \in \text{PATHSOF}(K) \mid$
 $(\text{FIRST}(\rho) = q_0) \wedge (\forall q' \in \rho)(q' \models h) \wedge$
 $(\forall \rho' \in \text{PATHSOF}(K))((\rho' \geq \rho) \rightarrow$
 $(\text{LAST}(\rho') \models h))\}) = \emptyset$

$\Leftrightarrow (\text{辅助定理 } 1) \{\rho \in \text{PATHSOF}(K) \mid$
 $(\text{FIRST}(\rho) = q_0) \wedge (\forall q' \in \rho)(q' \models h) \wedge$
 $(\forall \rho' \in \text{PATHSOF}(K))((\rho' \geq \rho) \rightarrow$
 $(\text{LAST}(\rho') \models h))\} = \emptyset$

$\Leftrightarrow (\forall \rho \in \text{PATHSOF}(K))(((\text{FIRST}(\rho) = q_0) \wedge$
 $(\forall q' \in \rho)(q' \models h)) \rightarrow (\exists \rho' \in \text{PATHSOF}(K))$
 $((\rho' \geq \rho) \wedge (\text{LAST}(\rho') \models h)))$

$\Leftrightarrow (\forall \rho \in \text{HISTORIESOF}(K))((\text{FIRST}(\rho) = q_0) \rightarrow$
 $((\exists q \in \rho)(q \models h)) \vee (\forall q' \in \rho)(\exists \rho' \in$
 $\text{PATHSOF}(K))((\text{FIRST}(\rho') = q') \wedge$
 $(\text{LAST}(\rho') \models h)))$

$\Leftrightarrow (\text{定义 } 6) K, q_0 \models A((\text{EF } h) \text{ W } h)$.

所以 $g2 \equiv_{(C,E)} A((\text{EF } h) \text{ W } h)$.

由(1)和(2),定理 3 得证.

证毕.

实际上,定理 3 表明了 DoReach h 和 TryReach h 所表达的是强规划和强循环规划目标^[9].虽然在文献[4]和文献[7]等文献中, TryReach h 所表示的扩展目标一度被认为无法用 CTL 公式表示,但定理 3 却指出事实并非如此——它可以被语义等价地表示成 $A((\text{EF } h) \text{ W } h)$.

分析 $E_A G_{LE}$ 的语法(定义 5)可以发现: Fail 和 Then 与 DoMaint、TryMaint、DoReach 和 TryReach 等算子不同,它们可以嵌套别的 $E_A G_{LE}$ 算子使用.下面的定理 4 和定理 5 表明某些含有 Fail 和 Then 算子的 $E_A G_{LE}$ 目标也可以被语义等价地表示成 CTL 目标.

定理 4. 设 $g1, g2 \in \mathcal{F}_E(\mathcal{P})$, $g' \in \mathcal{F}_C(\mathcal{P})$, 且 $g2 \equiv_{(C,E)} g'$. 那么对于 $g = g1 \text{ Then } g2$ 而言:

(1) 若 $g1 = h$, 则 $g \equiv_{(C,E)} h \wedge g'$;

(2) 若 $g1 = \text{DoMaint } h$ 或 $\text{TryReach } h$, 则 $g \equiv_{(C,E)} \text{AG } h$;

(3) 若 $g1 = \text{DoReach } h$, 则 $g \equiv_{(C,E)} A(\neg h \text{ U } (h \wedge g'))$;

(4) 若 $g1 = \text{TryReach } h$, 则 $g \equiv_{(C,E)} A((\neg h \wedge \text{EF } h) \text{ W } (h \wedge g'))$.

证明.

因为 $F_g(q_0) = \emptyset$

$\Leftrightarrow (\text{定义 } 7) F_{g1}(q_0) \cup \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in S_{g1}(q_0)) \wedge$
 $(\sigma_2 \in F_{g2}(\text{LAST}(\sigma_1)))\} = \emptyset$

$\Leftrightarrow F_{g1}(q_0) = \emptyset$ 且 $\{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in S_{g1}(q_0)) \wedge$
 $(\sigma_2 \in F_{g2}(\text{LAST}(\sigma_1)))\} = \emptyset$

$\Leftrightarrow F_{g1}(q_0) = \emptyset$ 且 $(\forall \sigma \in S_{g1}(q_0))(F_{g2}(\text{LAST}(\sigma)) = \emptyset)$

$\Leftrightarrow (\text{定理条件}) F_{g1}(q_0) = \emptyset$ 且 $(\forall \sigma \in S_{g1}(q_0))$
 $(K, \text{LAST}(\sigma) \models g')$. (*)

(1) 若 $g1 = h$, 那么(*)

$\Leftrightarrow (\text{定理 } 1) q_0 \models h$ 且 $(\forall \sigma \in S_{g1}(q_0))(K, \text{LAST}(\sigma) \models g')$

$\Leftrightarrow (\text{定义 } 7) q_0 \models h$ 且 $(\forall \sigma \in \{\{q_0\}\})(K, \text{LAST}(\sigma) \models g')$

$\Leftrightarrow K, q_0 \models h$ 且 $K, q_0 \models g'$

$\Leftrightarrow (\text{定义 } 6) K, q_0 \models h \wedge g'$.

(2) 若 $g1 = \text{DoMaint } h$ 或 $\text{TryMaint } h$, 那么(*)

$\Leftrightarrow (\text{定理 } 2) K, q_0 \models \text{AG } h$ 且 $(\forall \sigma \in S_{g1}(q_0))(K,$
 $\text{LAST}(\sigma) \models g')$

$\Leftrightarrow (\text{定义 } 7) K, q_0 \models \text{AG } h$ 且 $(\forall \sigma \in \emptyset)(K,$
 $\text{LAST}(\sigma) \models g')$

$\Leftrightarrow K, q_0 \models \text{AG } h$.

(3) 若 $g1 = \text{DoReach } h$, 那么(*)

$\Leftrightarrow (\text{定理 } 3) K, q_0 \models \text{AF } h$ 且 $(\forall \sigma \in S_{g1}(q_0))(K,$
 $\text{LAST}(\sigma) \models g')$

$\Leftrightarrow (\text{定义 } 6) (\forall \sigma = \{q_0, q_1, \dots\} \in$

$\text{HISTORIESOF}(K))(\exists i \geq 0)(K, q_i \models h)$ 且

$(\forall \sigma \in S_{g^1}(q_0))(K, LAST(\sigma) \models g')$
 $\Leftrightarrow (\forall \sigma = \{q_0, q_1, \dots\} \in HISTORIESOF(K))$
 $(\exists i \geq 0)(i > \forall j \geq 0)((K, q_j \not\models h) \text{ 且 } (K, q_i \models h))$
 $\text{ 且 } (\forall \sigma \in S_{g^1}(q_0))(K, LAST(\sigma) \models g')$
 $\Leftrightarrow (\text{定义 } 7)(\forall \sigma = \{q_0, q_1, \dots\} \in$
 $HISTORIESOF(K))(\exists i \geq 0)(i > \forall j \geq 0)$
 $((K, q_j \not\models h) \text{ 且 } (K, q_i \models h) \text{ 且 } (K, q_i \models g'))$
 $\Leftrightarrow (\text{定义 } 6) K, q_0 \models A(\neg h \cup (h \wedge g'))$.
 (4) 若 $g1 = \text{TryReach } h$, 那么 (*)
 $\Leftrightarrow (\text{定理 } 3) K, q_0 \models A((EF \ h) \ W \ h) \text{ 且 } (\forall \sigma \in$
 $S_{g^1}(q_0))(K, LAST(\sigma) \models g')$
 $\Leftrightarrow (\text{定义 } 6)(\forall \sigma = \{q_0, q_1, \dots\} \in$
 $HISTORIESOF(K))((\exists i \geq 0)(K, q_i \models h)$
 $\text{ 或者 } (\forall k \geq 0)(K, q_k \models EF \ h)), \text{ 且 } (\forall \sigma \in$
 $S_{g^1}(q_0))(K, LAST(\sigma) \models g')$
 $\Leftrightarrow (\forall \sigma = \{q_0, q_1, \dots\} \in HISTORIESOF(K))$
 $((\exists i \geq 0)(K, q_i \models h) \text{ 或者 } (\forall k \geq 0)((K, q_k \models$
 $EF \ h) \text{ 且 } (K, q_k \not\models h))), \text{ 且 } (\forall \sigma \in S_{g^1}(q_0))(K,$
 $LAST(\sigma) \models g')$
 $\Leftrightarrow (\forall \sigma = \{q_0, q_1, \dots\} \in HISTORIESOF(K))$
 $((\exists i \geq 0)(i > \forall j \geq 0)((K, q_j \not\models h) \text{ 且 } (K, q_i \models$
 $h)) \text{ 或 } (\forall k \geq 0)((K, q_k \models EF \ h) \text{ 且 } (K, q_k \not\models h))),$
 $\text{ 且 } (\forall \sigma \in S_{g^1}(q_0))(K, LAST(\sigma) \models g')$
 $\Leftrightarrow (\text{定义 } 7)(\forall \sigma = \{q_0, q_1, \dots\} \in$
 $HISTORIESOF(K))((\exists i \geq 0)(i > \forall j \geq 0)$
 $((K, q_j \not\models h) \text{ 且 } (K, q_i \models h) \text{ 且 } (K, q_i \models g')) \text{ 或 }$
 $(\forall k \geq 0)((K, q_k \models EF \ h) \text{ 且 } (K, q_k \not\models h)))$
 $\Leftrightarrow (\text{定义 } 6) K, q_0 \models A((\neg h \wedge EF \ h) \ W \ (h \wedge g'))$.
 所以由以上(1)、(2)、(3)和(4), 定理 4 得证.

证毕.

虽然文献[10]提出分别用 $AF(h \wedge g')$ 和 $A((EF(h \wedge g')) \ W \ (h \wedge g'))$ 来表示 $(\text{DoReach } h)$ Then g 和 $(\text{TryReach } h)$ Then g (其中 $g \equiv_{(C,E)} g'$), 但是当中没有考虑到相应的 $E_A G_{LE}$ 公式语义对极小路径集的要求, 所以有关的 CTL 公式所表示的规划目标较弱.

Then 算子的直观含义是“顺序实现”. “ $g1$ Then $g2$ ”形式的 $E_A G_{LE}$ 目标公式所表示的规划意图是“先实现子目标 $g1$ 然后再实现子目标 $g2$ ”. 其实根据定义 7, Then 算子是满足结合律的.

辅助定理 2. 设 $g1, g2, g3 \in \mathcal{F}_E(\mathcal{P})$. 令 $g = (g1 \text{ Then } g2) \text{ Then } g3, g' = g1 \text{ Then } (g2 \text{ Then } g3)$, 那么对于 $\forall q \in Q$ 有 $F_g(q) = F_{g'}(q)$ 并且

$$S_g(q) = S_{g'}(q).$$

证明. 记 $g4 = g1$ Then $g2, g5 = g2$ Then $g3$.

(1) 先证 $F_g(q) = F_{g'}(q)$.

$$\begin{aligned}
 & F_g(q) \\
 &= (\text{定义 } 7) F_{g^4}(q) \cup \{\sigma'_1 \cdot \sigma_3 \mid (\sigma'_1 \in S_{g^4}(q)) \wedge \\
 & \quad (\sigma_3 \in F_{g^3}(LAST(\sigma'_1)))\} \\
 &= (\text{定义 } 7) F_{g^4}(q) \cup \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in S_{g^1}(q)) \wedge \\
 & \quad (\sigma_2 \in F_{g^2}(LAST(\sigma_1)))\} \cup \{\sigma'_1 \cdot \sigma_3 \mid (\sigma'_1 \in \\
 & \quad S_{g^4}(q)) \wedge (\sigma_3 \in F_{g^3}(LAST(\sigma'_1)))\} \\
 &= (\text{定义 } 7) F_{g^1}(q) \cup \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in S_{g^1}(q)) \wedge \\
 & \quad (\sigma_2 \in F_{g^2}(LAST(\sigma_1)))\} \cup \{\sigma_1 \cdot \sigma_2 \cdot \sigma_3 \mid (\sigma_1 \in \\
 & \quad S_{g^1}(q)) \wedge (\sigma_2 \in S_{g^2}(LAST(\sigma_1))) \wedge (\sigma_3 \in \\
 & \quad F_{g^3}(LAST(\sigma_2)))\} \\
 &= F_{g^1}(q) \cup \{\sigma_1 \cdot \sigma'_2 \mid (\sigma_1 \in S_{g^1}(q)) \wedge (\sigma'_2 \in \\
 & \quad F_{g^2}(LAST(\sigma_1))) \cup \{\sigma'_2 \cdot \sigma'_3 \mid (\sigma'_2 \in \\
 & \quad S_{g^2}(LAST(\sigma_1))) \wedge (\sigma'_3 \in F_{g^3}(LAST(\sigma'_2)))\} \\
 &= (\text{定义 } 7) F_{g^1}(q) \cup \{\sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in S_{g^1}(q)) \wedge \\
 & \quad (\sigma_2 \in F_{g^5}(LAST(\sigma_1)))\} \\
 &= (\text{定义 } 7) F_{g'}(q).
 \end{aligned}$$

(2) 再证 $S_g(q) = S_{g'}(q)$.

$$\begin{aligned}
 & S_g(q) \\
 &= (\text{定义 } 7) \{\sigma'_1 \cdot \sigma_3 \mid (\sigma'_1 \in S_{g^4}(q)) \wedge (\sigma_3 \in \\
 & \quad S_{g^3}(LAST(\sigma'_1)))\} \\
 &= (\text{定义 } 7) \{\sigma_1 \cdot \sigma_2 \cdot \sigma_3 \mid (\sigma_1 \in S_{g^1}(q)) \wedge (\sigma_2 \in \\
 & \quad S_{g^2}(LAST(\sigma_1))) \wedge (\sigma_3 \in S_{g^3}(LAST(\sigma_2)))\} \\
 &= (\text{定义 } 7) \{\sigma_1 \cdot \sigma'_2 \mid (\sigma_1 \in S_{g^1}(q)) \wedge (\sigma'_2 \in \\
 & \quad S_{g^5}(LAST(\sigma_1)))\} \\
 &= (\text{定义 } 7) S_{g'}(q).
 \end{aligned}$$

所以由以上(1)和(2), 辅助定理 2 得证. 证毕.

根据辅助定理 2, 任何不含 Fail 算子的 $E_A G_{LE}$ 目标公式都可以语义等价地转化为定理 4 所描述的形式. 所以我们可以直接得到推论 1.

推论 1. 设 $g \in \mathcal{F}_E(\mathcal{P})$ 且 g 中不包含 Fail 算子, 则必然存在 $g' \in \mathcal{F}_C(\mathcal{P})$ 使得 $g \equiv_{(C,E)} g'$.

而 Fail 算子的直观含义是“若失败则”, 它是 $E_A G_{LE}$ 表示优先处理机制的体现. “ $g1 \text{ Fail } g2$ ”形式的目标公式所表示的规划意图是“优先实现子目标 $g1$, 若失败则实现子目标 $g2$ ”.

定理 5. 设 $g1, g2 \in \mathcal{F}_E(\mathcal{P}), g' \in \mathcal{F}_C(\mathcal{P})$, 且 $g2 \equiv_{(C,E)} g'$. 那么对于 $g = g1 \text{ Fail } g2$ 而言:

- (1) 若 $g1 = h$, 则 $g \equiv_{(C,E)} h \vee g'$;
- (2) 若 $g1 = \text{DoMaint } h$, 则 $g \equiv_{(C,E)} (AG \ h) \vee g'$;
- (3) 若 $g1 = \text{TryMaint } h$, 则 $g \equiv_{(C,E)} A(h \ W$

$(\neg h \wedge g')$);

(4) 若 $g1 = \text{DoReach } h$, 则 $g \equiv_{(C,E)} (\text{AF } h) \vee g'$;

(5) 若 $g1 = \text{TryReach } h$, 则 $g \equiv_{(C,E)} A ((\text{EF } h)$

$W (h \vee (\text{AG } \neg h) \wedge g'))$.

证明. 因为 $F_g(q_0) = \emptyset$

$\Leftrightarrow (\text{定义 } 7) \{ \sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in F_{g1}(q_0)) \wedge (\sigma_2 \in F_{g2}(\text{LAST}(\sigma_1))) \} = \emptyset$

$\Leftrightarrow (\forall \sigma \in F_{g1}(q_0)) (F_{g2}(\text{LAST}(\sigma)) = \emptyset)$

$\Leftrightarrow (\text{定理条件}) (\forall \sigma \in F_{g1}(q_0)) (K, \text{LAST}(\sigma) \models g')$
(**)

(1) 若 $g1 = h$, 那么(**)

$\Leftrightarrow (\text{定义 } 7) F_{g1}(q_0) = \emptyset$ 或者 $(\forall \sigma \in \{ \{q_0\} \}) (K, \text{LAST}(\sigma) \models g')$

$\Leftrightarrow (\text{定理 } 1) q_0 \models h$ 或者 $K, q_0 \models g'$

$\Leftrightarrow (\text{定义 } 6) K, q_0 \models h \vee g'$.

(2) 若 $g1 = \text{DoMaint } h$, 那么(**)

$\Leftrightarrow (\text{定义 } 7) F_{g1}(q_0) = \emptyset$ 或者 $(\forall \sigma \in \{ \{q_0\} \}) (K, \text{LAST}(\sigma) \models g')$

$\Leftrightarrow (\text{定理 } 2) K, q_0 \models \text{AG } h$ 或者 $K, q_0 \models g'$

$\Leftrightarrow (\text{定义 } 6) K, q_0 \models (\text{AG } h) \vee g'$.

(3) 若 $g1 = \text{TryMaint } h$, 那么(**)

$\Leftrightarrow (\text{可以将所有 } \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K) \text{ 分成两个互不相交类别:}$

(a) $(\forall k \geq 0) (K, q_k \models h)$;

(b) $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \neg h) \wedge (K, q_j \models h))$)

$(\forall \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K))$

$((\forall k \geq 0) (K, q_k \models h)$, 或者 $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \neg h) \wedge (K, q_j \models h))$), 并且

$(\forall \sigma \in F_{g1}(q_0)) (K, \text{LAST}(\sigma) \models g')$

$\Leftrightarrow (\text{定义 } 7) (\forall \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K)) ((\forall k \geq 0) (K, q_k \models h)$, 或者 $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \neg h) \wedge (K, q_j \models h) \wedge (K, q_i \models g'))$)

$\Leftrightarrow (\text{定义 } 6) K, q_0 \models A (h \text{ W } (\neg h \wedge g'))$

(4) 若 $g1 = \text{DoReach } h$, 那么(**)

$\Leftrightarrow (\text{定义 } 7) F_{g1}(q_0) = \emptyset$ 或者 $(\forall \sigma \in \{ \{q_0\} \}) (K, \text{LAST}(\sigma) \models g')$

$\Leftrightarrow (\text{定理 } 3) K, q_0 \models \text{AF } h$ 或者 $K, q_0 \models g'$

$\Leftrightarrow (\text{定义 } 6) K, q_0 \models (\text{AF } h) \vee g'$

(5) 若 $g1 = \text{TryReach } h$, 那么(**)

$\Leftrightarrow (\text{可以将所有 } \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K) \text{ 分成 3 个互不相交的类别:}$

(a) $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models h) \wedge (K, q_j \models \neg h))$;

(b) $(\forall k \geq 0) (K, q_k \models \neg h \wedge (\text{EF } h))$;

(c) $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \text{AG } \neg h) \wedge (K, q_j \models \neg h \wedge (\text{EF } h)))$)

$(\forall \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K)) ((\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models h) \wedge (K, q_j \models \neg h))$, 或者 $(\forall k \geq 0) (K, q_k \models \neg h \wedge (\text{EF } h))$, 或者 $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \text{AG } \neg h) \wedge (K, q_j \models \neg h \wedge (\text{EF } h)))$), 并且 $(\forall \sigma_1 \in F_{g1}(q_0)) (K, \text{LAST}(\sigma_1) \models g')$

$\Leftrightarrow (\text{定义 } 7) (\forall \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K)) ((\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models h) \wedge (K, q_j \models \neg h))$, 或者 $(\forall k \geq 0) (K, q_k \models \neg h \wedge (\text{EF } h))$, 或者 $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \text{AG } \neg h) \wedge (K, q_j \models \neg h \wedge (\text{EF } h)))$), 并且 $(\forall \sigma_1 \in F_{g1}(q_0)) (K, \text{LAST}(\sigma_1) \models g')$

$\Leftrightarrow (\text{定义 } 7) (\forall \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K)) ((\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models h) \wedge (K, q_j \models \neg h))$, 或者 $(\forall k \geq 0) (K, q_k \models \neg h \wedge (\text{EF } h))$, 或者 $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \text{AG } \neg h) \wedge (K, q_j \models \neg h \wedge (\text{EF } h)) \wedge (K, q_i \models g'))$)

$\Leftrightarrow (\forall \sigma = \{q_0, q_1, \dots\} \in \text{HISTORIESOF}(K)) ((\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models h) \wedge (K, q_j \models \text{EF } h))$, 或者 $(\forall k \geq 0) (K, q_k \models \text{EF } h)$, 或者 $(\exists i \geq 0) (i > \forall j \geq 0) ((K, q_i \models \text{AG } \neg h) \wedge (K, q_j \models \text{EF } h) \wedge (K, q_i \models g'))$)

$\Leftrightarrow (\text{定义 } 6) A ((\text{EF } h) \text{ W } (h \vee (\text{AG } \neg h) \wedge g'))$
所以由以上(1)、(2)、(3)、(4)和(5), 定理 5 得证.

证毕.

虽然定理 2 指出单独的 $\text{DoMaint } h$ 和 $\text{TryMaint } h$ 公式所表示的目标含义相同. 但在定理 5 中, 我们可以发现: 若 DoMaint 和 TryMaint 算子分别与 Fail 算子结合使用, 那么其目标公式的语义将会不同.

可以用 Fail 算子表示子目标间的优先处理机制是 $E_A G_{LE}$ 较 CTL 在表示规划目标方面的一大特点, 并且 $(\text{TryReach } h_1) \text{ Fail } (\text{DoReach } h_2)$ 一类的 $E_A G_{LE}$ 目标公式也曾被认为是 CTL 所不能表达的^[4,7]. 但是根据定理 3 和定理 5, 我们可以用 CTL 目标公式 $A ((\text{EF } h_1) \text{ W } (h_1 \vee (\text{AG } \neg h_1) \wedge (\text{AF } h_2)))$ 语义等价地表示这类目标.

其实根据定义 7, Fail 算子也是满足结合律的.

辅助定理 3. 设 $g1, g2, g3 \in \mathcal{F}_E(\mathcal{P})$. 令 $g = (g1 \text{ Fail } g2) \text{ Fail } g3, g' = g1 \text{ Fail } (g2 \text{ Fail } g3)$, 那么对于 $\forall q \in Q$ 有 $F_g(q) = F_{g'}(q)$ 并且 $S_g(q) = S_{g'}(q)$.

证明. 记 $g4 = g1 \text{ Fail } g2, g5 = g2 \text{ Fail } g3$.

(1) 先证 $F_g(q) = F_{g'}(q)$.

$$\begin{aligned}
 & F_g(q) \\
 = & (\text{定义 7}) \{ \sigma'_1 \cdot \sigma_3 \mid (\sigma'_1 \in F_{g^4}(q)) \wedge (\sigma_3 \in F_{g^3}(\text{LAST}(\sigma'_1))) \} \\
 = & (\text{定义 7}) \{ \sigma_1 \cdot \sigma_2 \cdot \sigma_3 \mid (\sigma_1 \in F_{g^1}(q)) \wedge (\sigma_2 \in F_{g^2}(\text{LAST}(\sigma_1))) \wedge (\sigma_3 \in F_{g^3}(\text{LAST}(\sigma_2))) \} \\
 = & (\text{定义 7}) \{ \sigma_1 \cdot \sigma'_2 \mid (\sigma_1 \in F_{g^1}(q)) \wedge (\sigma'_2 \in F_{g^5}(\text{LAST}(\sigma_1))) \} \\
 = & (\text{定义 7}) F_{g'}(q).
 \end{aligned}$$

(2) 再证 $S_g(q) = S_{g'}(q)$.

$$\begin{aligned}
 & S_g(q) \\
 = & (\text{定义 7}) S_{g^4}(q) \cup \{ \sigma'_1 \cdot \sigma_3 \mid (\sigma'_1 \in F_{g^4}(q)) \wedge (\sigma_3 \in S_{g^3}(\text{LAST}(\sigma'_1))) \} \\
 = & (\text{定义 7}) S_{g^1}(q) \cup \{ \sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in F_{g^1}(q)) \wedge (\sigma_2 \in S_{g^2}(\text{LAST}(\sigma_1))) \} \cup \{ \sigma'_1 \cdot \sigma_3 \mid (\sigma'_1 \in F_{g^4}(q)) \wedge (\sigma_3 \in S_{g^3}(\text{LAST}(\sigma'_1))) \} \\
 = & (\text{定义 7}) S_{g^1}(q) \cup \{ \sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in F_{g^1}(q)) \wedge (\sigma_2 \in S_{g^2}(\text{LAST}(\sigma_1))) \} \cup \{ \sigma_1 \cdot \sigma_2 \cdot \sigma_3 \mid (\sigma_1 \in F_{g^1}(q)) \wedge (\sigma_2 \in F_{g^2}(\text{LAST}(\sigma_1))) \wedge (\sigma_3 \in S_{g^3}(\text{LAST}(\sigma_2))) \} \\
 = & S_{g^1}(q) \cup \{ \sigma_1 \cdot \sigma'_2 \mid (\sigma_1 \in F_{g^1}(q)) \wedge (\sigma'_2 \in S_{g^2}(\text{LAST}(\sigma_1))) \} \cup \{ \sigma'_2 \cdot \sigma'_3 \mid (\sigma'_2 \in F_{g^2}(\text{LAST}(\sigma_1))) \wedge (\sigma'_3 \in S_{g^3}(\text{LAST}(\sigma'_2))) \} \\
 = & (\text{定义 7}) S_{g^1}(q) \cup \{ \sigma_1 \cdot \sigma_2 \mid (\sigma_1 \in F_{g^1}(q)) \wedge (\sigma_2 \in S_{g^5}(\text{LAST}(\sigma_1))) \} \\
 = & (\text{定义 7}) S_{g'}(q).
 \end{aligned}$$

所以由以上(1)和(2), 辅助定理 3 得证. 证毕.

根据辅助定理 3, 任何不含 Then 算子的 $E_A G_{LE}$ 目标公式都可以语义等价地转化为定理 5 所描述的形式. 所以我们可以直接得到推论 2.

推论 2. 设 $g \in \mathcal{F}_E(\mathcal{P})$ 且 g 中不包含 Then 算子, 则必然存在 $g' \in \mathcal{F}_C(\mathcal{P})$ 使得 $g \equiv_{\{C, E\}} g'$.

结合辅助定理 2 和辅助定理 3, 我们可以将所有包含 Then 或 Fail 算子的 $E_A G_{LE}$ 目标公式等价地改写成 Then 和 Fail 相间嵌套的标准形式.

推论 3. 设 $g \in \mathcal{F}_E(\mathcal{P})$ 且 g 中包含 Then 或 Fail 算子. 记 $OP_0 = \text{Then}$, $OP_1 = \text{Fail}$ 且 $x \in \{0, 1\}$, 则 g 可被等价地改写为 $(\dots(\dots(g_0 OP_{(x+1) \bmod 2} g_1) \dots OP_{(x+i) \bmod 2} g_i) \dots) OP_{(x+n) \bmod 2} g_n)$, 其中 g_0 不含 Then 或 Fail 算子.

根据推论 3, 在定理 4 和定理 5 中所考虑的 $E_A G_{LE}$ 目标公式 g 都是 $n=1$ 时的特殊情况.

4.2 $E_A G_{LE}$ 较之于 CTL 的优缺点

通过 $E_A G_{LE}$ 的语法(即定义 5), 我们可以发现

$E_A G_{LE}$ 注重于表示规划的“意图”和失败处理机制(如其中的 DoReach、TryReach 和 Fail 算子), 而这些在 CTL 的语法中则没有被突出. 所以在描述扩展的规划目标时, 用 $E_A G_{LE}$ 比用 CTL 更自然且更易被理解. 另外, $E_A G_{LE}$ 的语义系统(见定义 7)利用了极小路径集 \min 这一概念, 这使得 $E_A G_{LE}$ 目标公式不仅包含了对规划执行效果的时态要求, 还包含了对规划解搜索过程的控制信息^[4]. 这导致了:

(1) 在目标表示的层次上, 为了弥补搜索控制信息的缺失, CTL 公式要比与之语义等价的 $E_A G_{LE}$ 公式显得繁琐得多. 例如, 根据定理 3 和定理 4, 对于 $E_A G_{LE}$ 目标公式 $\text{TryReach } h_0 \text{ Then}(\text{TryReach } h_1 \text{ Then}(\dots(\text{TryReach } h_n \text{ Then } g) \dots))$, 与之语义等价的 CTL 目标公式必须包含 3 个 $h_i (n \geq i \geq 0)$.

(2) 在搜索规划解的层次上, 无论是针对 CTL 目标公式还是 $E_A G_{LE}$ 目标公式, 规划算法都要建立用于指导搜索过程的控制自动机^[4-5]. 但是在对 CTL 目标公式构造自动机时, 算法需要不断地将对当前状态的时态要求映射到对其直接后继状态的时态要求; 而对于 $E_A G_{LE}$ 目标公式, 相应的构造过程则简单得多, 只需要将与各子目标公式对应的局部自动机作拼接即可.

以上是 $E_A G_{LE}$ 较之于 CTL 在表示规划目标方面的优点.

但是根据 $E_A G_{LE}$ 目标公式语义的严格定义(即定义 7), 对于 $\forall g \in \mathcal{F}_E(\mathcal{P})$ 和 $\forall q \in Q$, $S_g(q)$ 和 $F_g(q)$ 都无法描述任何固定长度的路径 $path$ (除非 $path = \{q\}$). 然而, 在 CTL 目标公式中, 我们可以用“下一状态”时态算子 X 来描述任意固定长度的路径(见定义 6).

另外, 对于任何满足 $E_A G_{LE}$ 目标公式 g 的规划 π 而言, $F_g(q_0)$ 必须为 \emptyset (见定义 8) 即 $(\forall \rho \in \text{PATHSOF}(K)) ((\text{FIRST}(\rho) = q_0) \rightarrow (\rho \notin F_g(q_0)))$. 所以, $E_A G_{LE}$ 目标公式的语义都是建立在所有可能的执行路径的共性上的. 由于在 $E_A G_{LE}$ 目标公式中不存在类似于求非(即 \rightarrow)的操作(见定义 7), 所以我们无法用 $E_A G_{LE}$ 目标公式表示那些仅仅在某些可能的执行路径上才成立的时态性质. 然而, 在 CTL 目标公式中, 我们可以用存在路径量词 E 来描述这类时态性质(见定义 6).

5 结论和未来工作

本文在规划的执行结构这一语义层次上对

CTL 和 E_{AGLE} 这两种扩展目标表示语言做了严格的比较,证明了对于许多 E_{AGLE} 目标公式(包括了文献[4]和文献[7]所列举的所有不可被 CTL 表示的 E_{AGLE} 目标公式)而言,都存在着一个与之语义等价 CTL 目标公式,并进一步分析指出了 E_{AGLE} 与 CTL 相比在表示规划目标时更简洁、相应的控制自动机构造过程也更简单,但是它无法表示一些基本的 CTL 语义特征.

将那些在本文中还未被考虑到的 E_{AGLE} 目标公式(例如,在推论 3 中 $n>1$ 时的公式,以及那些包含 And 或 Repeat 算子的 E_{AGLE} 公式)与 CTL 目标公式作语义比较,看看它们可否被语义等价地表示成 CTL 目标公式,并将有关语义比较的结果用于改进相应的规划算法是我们下一步的工作.而在部分可观察的规划领域^[11]中,规划不仅要考虑上下文信息还要考虑观测信息^[12],如何在这种框架^[13]下表示扩展目标与搜索规划解也是我们未来的研究目标.另外,在 MDP 规划问题中,规划领域带有概率信息而规划目标则体现为效能函数^[14],如何给出一个既可以考虑规划领域里的概率信息也可以考虑复杂的时态性质要求的扩展目标表示框架则是我们未来的另一个研究方向.

参 考 文 献

[1] Bacchus F, Kabanza F. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 2000, 116: 123-191

[2] Kabanza F, Barbeau M, St-Denis R. Planning control rules for reactive agents. *Artificial Intelligence*, 1997, 95(1): 67-113

[3] Emerson E A. Temporal and modal logic//van Leeuwen J. *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*. Amsterdam: Elsevier, 1990: 995-1072

[4] Dal Lago U, Pistore M, Traverso P. Planning with a language for extended goals//*Proceedings of the 18th National*

Conference on Artificial Intelligence(AAAI'02). Edmonton, Alberta, Canada, 2002: 447-454

[5] Pistore M, Traverso P. Planning as model checking for extended goals in non-deterministic domains//*Proceedings of the 17th International Joint Conference on Artificial Intelligence(IJCAI'01)*. Seattle, WA, USA, 2001: 479-484

[6] Jensen R, Veloso M, Bryant R. Guided symbolic universal planning//*Proceedings of the 13th International Conference on Automated Planning and Scheduling(ICAPS'03)*. Trento, Italy, 2003: 123-132

[7] Ghallab M, Nau D, Traverso P. *Automated Planning: Theory and Practice*. San Francisco, CA, USA: Morgan Kaufmann, Elsevier, 2004

[8] Pistore M, Barbon F, Bertoli P, Shaparau D, Traverso P. Planning and monitoring web service composition//*Proceedings of the Artificial Intelligence: Methodology, Systems, and Applications, 11th International Conference(AIMSA'04)*. Varna, Bulgaria, 2004: 106-115

[9] Cimatti A, Pistore M, Roveri M, Traverso P. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 2003, 147: 35-84

[10] Huang Wei, Wen Zhong-Hua, Jiang Yun-Fei, Chen Ai-Xiang. Comparison between two languages used to express planning goals: CTL and E_{AGLE} //*Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence(PRICA'06)*. Guilin, China, 2006: 180-189

[11] Bertoli P, Cimatti A, Roveri M, Traverso P. Strong planning under partial observability. *Artificial Intelligence*, 2006, 170: 337-384

[12] Huang Wei, Wen Zhong-Hua, Jiang Yun-Fei, Wu Li-Hua. Observation reduction for strong plans//*Proceedings of the 20th International Joint Conference on Artificial Intelligence(IJCAI'07)*. Hyderabad, India, 2007: 1930-1935

[13] Bertoli P, Cimatti A, Pistore M, Traverso P. A framework for planning with extended goals under partial observability//*Proceedings of the 13th International Conference on Automated Planning and Scheduling(ICAPS'03)*. Trento, Italy, 2003: 215-225

[14] Thiebaux S, Gretton C, Slaney J, Price D, Kabanza F. Decision-theoretic planning with non-Markovian rewards. *Journal of Artificial Intelligence Research*, 2006, 25: 17-74



HUANG Wei, born in 1979, Ph.D.. His research interests include automated planning, model-based diagnosis, and automated reasoning.

supervisor. His research interests include automated planning, model-based diagnosis, and automated reasoning.

WEN Zhong-Hua, born in 1966, Ph. D., associate professor. His research interests include automated planning and graph theory.

PENG Hong, born in 1956, Ph. D., professor, Ph. D. supervisor. His research interests include data mining and technology of intelligent network.

Background

This work is an important component of the research on domain knowledge extraction, which is one of the main tasks of “the Research on Domain Knowledge Extraction and Reasoning in Automated Planning” project. The project aims to propose an efficient planning algorithm that can make use of domain knowledge, and it includes three main tasks: (1) the research on domain knowledge extraction, (2) the research on domain knowledge reasoning and, (3) the study of strategies for domain knowledge utilization in automated planners. The project is supported by the National Natural Science Foundation of China under grant No. 60773201, the Guangdong Natural Science Foundation under grant No. 07006474,

and the Guangdong Scientific and Technological Project Foundation under grant No. 2007B01020044.

Some achievements of the planning group are reached by now: A formalism of planning with domain constraint based on model checking, a new SAT planner using genetic algorithm for solving SAT problems based on learning clause weights, a research on relations of effect of action for STRIPS domain, a strategy of extracting domain knowledge, and the first algorithm called STRONG-FO-PO for observation reduction of strong plans under full observability, which are presented by Wu Kang-Hen, Ling Ying-Biao, Wu Xiang-Jun, Jiang Yun-Fei and Huang Wei et al.