ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.12, December 2006, pp.2485−2498
DOI: 10.1360/jos172485

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

有界模型检测同步多智体系统的时态认知逻辑*

骆翔宇[1,2+], 苏开乐[2,3], 杨晋吉[2]

[1](桂林电子科技大学 计算机系,广西 桂林 541004)
[2](中山大学 计算机科学系,广东 广州 510275)
[3](河南科技大学 电子信息工程学院,河南 洛阳 471003)

Bounded Model Checking for Temporal Epistemic Logic in Synchronous Multi-Agent Systems

LUO Xiang-Yu[1,2+], SU Kai-Le[2,3], YANG Jin-Ji[2]

[1](Department of Computer Science, Guilin University of Electronic Technology, Guilin 541004, China)
[2](Department of Computer Science, Sun Yat-Sen University, Guangzhou 510275, China)
[3](Institute of Electronics and Information Engineering, He'nan University of Science and Technology, Luoyang 471003, China)
+ Corresponding author: Phn: +86-773-5604481, E-mail: shiangyuluo@gmail.com, http://www.guet.edu.cn

Luo XY, Su KL, Yang JJ. Bounded model checking for temporal epistemic logic in synchronous multi-Agent systems. *Journal of Software*, 2006,17(12):2485−2498. http://www.jos.org.cn/1000-9825/17/2485.htm

Abstract: This paper presents an approach to the verification of temporal epistemic logic in synchronous multi-Agent systems via bounded model checking (BMC). By incorporating epistemic modalities into temporal logic CTL*, a temporal epistemic logic $ECKL_n$ is introduced, and it is interpreted under the semantics of synchronous interpreted system. The temporal epistemic expressive power of $ECKL_n$ is greater than that of Penczek and Lomuscio's logic CTLK. Agents' knowledge interpreted under synchronous semantics can be skillfully attained by the state position function, which avoids extending the encoding of the states and the transition relation of plain temporal epistemic model for time domain. The technical details and the correctness of the BMC method for logic $AECKL_n$/$EECKL_n$ (the universal or existential fragment of $ECKL_n$) are given. A case study of train controller system is presented to illustrate the processing of the BMC method.

Key words: model checking; bounded model checking; multi-Agent system; synchronous temporal epistemic model; temporal epistemic logic

摘 要: 提出在同步的多智体系统中验证时态认知逻辑的有界模型检测(bounded model checking,简称 BMC)算法.基于同步解释系统语义,在时态逻辑 CTL*的语言中引入认知模态词,从而得到一个新的时态认知逻辑 $ECKL_n$.通过引入状态位置函数的方法获得同步系统的智能体知识,避免了为时间域而扩展通常的时态认知模型的状态及迁移关系编码.$ECKL_n$的时态认知表达能力强于另一个逻辑CTLK.给出该算法的技术细节及正确性证明,并用火车控制系统实例解释算法的执行过程.

* Supported by the National Natural Science Foundation of China under Grant Nos.60496327, 10410638, 60473004 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2005CB321900 (国家重点基础研究发展规划(973)); the Natural Science Foundation of Guangdong Province of China under Grant Nos.04205407, 06023195 (广东省自然科学基金)
Received 2005-02-22; Accepted 2006-01-09

关键词: 模型检测;有界模型检测;多智体系统;同步时态认知模型;时态认知逻辑
中图法分类号: TP301 文献标识码: A

模型检测作为一种自动形式化验证有限状态系统的技术已得到了广泛的研究与应用[1].在一些芯片设计公司,该技术已集成到他们的标准质量保证过程中.最近几年,该技术在多智体系统(multi-Agent system,简称MAS)和规划等 AI 领域中的应用已成为一个研究热点.由于大多数模型检测工具采用的规范是由时态逻辑表示的公式(如 SMV 的规范是 CTL 公式,SPIN 的规范是 LTL 公式),而多智体系统更强调对智能体的精神状态(如知识、信念、愿望和意图等)进行形式化表示和推理[2],因此,多智体系统验证的一个主流方向是在传统模型检测技术采用的时态逻辑语言中加入认知模态词来描述智能体拥有的信息及其心智状态,从而扩展现有的模型检测技术[3,4].

上述研究方向中已经取得了一些研究成果.知识逻辑的模型检测首先由 Halpern 和 Vardi 提出,然后, van der Hoek 和 Wooldridge 把时态认知逻辑 CKL_n 的模型检测问题转化为 LTL 模型检测问题[3].另外,他们在文献[4]中给出了将知识公式转化为 ATL 公式后用模型检测器 MOCHA 来验证的方法.然而,这些方法并没有直接实现验证知识与时间的模型检测工具.2003 年,van der Meyden 等人采用 OBDD 等技术开发的时态认知逻辑模型检测工具 MCK 在 CTL 或 LTL 逻辑的语言中加入认知算子,其中知道算子可以根据可观察语义、时钟语义和完美记忆语义进行解释.我们在文献[5]中讨论了根据知识语义和集合理论检测知识与公共知识的模型检测方法.另外,我们在文献[6]中提出了一种根据带局部命题的解释系统语义进行 CKL_n 符号化模型检测的算法,并采用 OBDD 技术,在模型检测工具 NuSMV2.1.2 的基础上开发了一个时态认知逻辑模型检测工具 MCTK.

然而,采用 OBDD 的模型检测技术可能会随着模型规模的扩大而产生状态的指数爆炸问题,因此,Biere 等人首先在 1999 年提出 LTL 的符号化有界模型检测(bounded model checking,简称 BMC)方法[7],其基本思想是只采用足以用来验证某一规范的部分状态空间,将 LTL 的存在模型检测问题转化为一个命题公式的可满足性问题.其优点是不会遇到状态爆炸问题,并且能够快速获取最小长度的反例,内存消耗远小于基于 OBDD 的方法,而且无须对变量进行静态或动态排序.因此,BMC 是基于 OBDD 模型检测技术的一个重要补充.随后,Penczek 等人在 2002 年提出了 ACTL(CTL 的全称片断)的 BMC 方法,并实现了一个 BMC 工具(BBMC).2004 年,Woźna 在文献[8]中论述了 ACTL*(CTL*的全称片断)的 BMC 方法,但没有实现相应的工具.2003 年,Penczek 等人提出在多智体系统中验证时态认知逻辑 ACTLK(CTLK 的全称片断)的有界模型检测方法[9],并开发了相应的 BMC 工具(BMCIS).ACTLK 逻辑是在 ACTL 逻辑中加入知道、全都知道、分布知识和公共知识 4 个认知算子后得到的.

根据以上研究进展,关于如何有界模型检测基于 CTL*的时态认知逻辑自然地成为一个有待解决的问题.我们已在文献[10]中初步提出相应的解决方案.本文将具体提出一种基于 CTL*的时态认知逻辑的有界型检测算法并给出完整的正确性证明.首先,基于同步解释系统语义,在时态逻辑 CTL*的语言中扩展认知算子(知道、全都知道、分布知识和公共知识算子),从而得到一个新的时态认知逻辑 $ECKL_n$,同时可以得到 $EECKL_n$ 逻辑($ECKL_n$ 的存在片断)和 $AECKL_n$ 逻辑($ECKL_n$ 的全称片断),$AECKL_n$ 与 $EECKL_n$ 的表达能力相同;然后,扩展文献[8]的 ACTL*有界模型检测方法,使之能够处理 $AECKL_n$ 和 $EECKL_n$ 逻辑的认知算子.另外,同步解释系统的语义需要引入时间域,我们引入一个状态位置函数来获取同步语义下的智能体知识,从而巧妙地避免了因为引入时间域而扩展普通解释系统的状态及迁移关系编码.与文献[9]的方法相比,我们的 $AECKL_n$/$EECKL_n$ 逻辑的有界模型检测算法的优势是:(1) 在时态方面,由于 $AECKL_n$ 采用的时态逻辑是 ACTL*,它同时包含 ACTL 和 LTL 逻辑.显然,$AECKL_n$ 的时态表达能力要强于基于 ACTL 的 ACTLK 逻辑;(2) 在认知方面,由于在同步解释系统中智能体的知识不仅包含自身的局部状态,而且也包含系统的共享时钟,因此在同步解释系统中,智能体可以获得比普通解释系统中的智能体更多的知识.另一方面,我们允许认知公式(主算子为认知算子的公式)约束的子公式为状态公式或路径公式;而 ACTLK 逻辑只允许包含状态公式.因此,$AECKL_n$ 的认知表达能力也强于 ACTLK 逻辑.

本文第 1 节提出同步解释系统语义.第 2 节定义 $ECKL_n$ 逻辑的语法和同步语义.第 3 节提出 $EECKL_n$ 逻辑的有界语义.第 4 节给出 $EECKL_n$ 逻辑的同步语义和有界语义的等价性证明.第 5 节给出从迁移关系和 $EECKL_n$ 公式到命题公式的转换方法及 BMC 算法框架.第 6 节证明公式转换的正确性.在第 7 节中,用火车控制系统实例解释算法的执行过程.第 8 节总结并展望未来的工作.

1 同步解释系统

为了表示多智体系统计算状态间的认知关系,本文采用文献[2]的同步解释系统语义.假设一个多智体系统由一个环境和 n 个智能体 $A=\{1,\ldots,n\}$ 组成.环境有一个局部状态集合 L_e 和一个动作集合 Act_e.每个智能体 $i\in A$ 也有一个局部状态集合 L_i 和一个动作集合 Act_i.系统的全局状态定义为 $G=L_e\times L_1\times\ldots\times L_n$.每个智能体 i 根据协议 $P_i:L_i\to 2^{Act_i}$ 执行动作,环境也对应一个协议 $P_e:L_e\to 2^{Act_e}$.我们通过一个迁移函数 $t:G\times Act\to G$ 来刻画系统的行为,其中,$Act\subseteq Act_e\times Act_1\times\ldots\times Act_n$ 是联合动作集合.*run* r 是一个从时间(自然数)到全局状态 G 的函数 $r:N\to G$,可视为由 G 中的全局状态组成的无穷序列.假设 r 是一个 *run*,m 是时间(自然数),则(r,m)称为一个 *point*.我们定义 $r(m)=(s_e,s_1,\ldots,s_n)$为在 $point(r,m)$的一个全局状态(简称状态),其中 $s_e\in L_e,s_i\in L_i(i\in A)$.令 $r_e(m)=s_e,r_i(m)=s_i(i\in A)$,则 $r_i(m)$就是智能体 i 在 $point(r,m)$的局部状态.

为了在解释系统中引入知识,我们对每个智能体 $i\in A$ 定义一个等价关系$\sim_i$.令 s 和 s' 为 G 中的两个全局状态,则 $s\sim_i s'$ 当且仅当 $s_i=s'_i$,即全局状态 s 和 s' 对于智能体 i 来说是不可辨别的(indistinguishable).令 $r(n)=s,r'(n')=s'$,则$(r,n)\sim_i(r',n')$当且仅当 $s_i=s'_i$.直觉上看,如果状态 s 和 s' 对于智能体 i 是不可辨别的,那么,它就认为状态 s 中也可能包含状态 s'.因此,智能体的知识完全由它的局部状态决定.然后,我们引入 K_i(知道)、D_Γ(分布知识)、E_Γ(全都知道)和 C_Γ(公共知识)4 个认知算子,其中 $i\in A,\Gamma\subseteq A.K_i,D_\Gamma$和 E_Γ用到的认知关系分别定义为 $\sim_i,\sim_\Gamma^D=\bigcap_{i\in\Gamma}\sim_i$ 和 $\sim_\Gamma^E=\bigcup_{i\in\Gamma}\sim_i$,而 C_Γ的认知关系是 $\sim_\Gamma^E$ 的传递闭包,用 $\sim_\Gamma^C$ 表示[2].

本文讨论的是同步的多智体系统.由于许多系统通常都会假设所有智能体均可访问一个共享的时钟,或者假设它们的动作在 *round* 中发生,并且它们在系统的整个运行过程中始终知道当前的 *round*(在一个系统的一个 *run* 中,*round* m 在时间点 $m-1$ 和 m 之间发生).这些假设意味着时间是系统中所有智能体的公共知识,它们同步地执行动作.形式地,已知一个系统 K,对于所有的智能体 $i\in A$ 和 K 中任意的 $point(r,n)$和(r',n'),如果$(r,n)\sim_i(r',n')$,则 $n=n'$,那么,K 就是一个同步系统.这样,我们可将文献[9]定义 1 中的认知可达关系修改为同步认知可达关系,从而得到定义 1 的同步时态认知模型.

定义 1(同步时态认知模型). 已知一个智能体集合 $A=\{1,\ldots,n\}$,则同步时态认知模型是一个二元组 $M=(K,V)$,其中,$K=(S,s_0,T,\overset{T}{\sim_1},\ldots,\overset{T}{\sim_n})$,并且:(1) S 是系统 K 的 *point* 集合;(2) $s_0\in S$ 是系统 K 的初始状态对应的 *point*;(3) $T\subseteq S\times S$ 是一个完全(total)二元后继关系(状态迁移关系);(4) $\overset{T}{\sim_i}\subseteq S\times S\ (i\in A)$ 是智能体 i 的同步认知可达关系,令$(r,m),(r',m')\in S$,则 $(r,m)\overset{T}{\sim_i}(r',m')$ 当且仅当 $r_i(m)=r'_i(m')$并且 $m=m'$;(5) $V:S'\times PV\to\{\text{true,false}\}$是一个命题变量集合 PV 的评价函数(S'是 S 对应的状态集合),使得对于所有$(r,m)\in S$ 和 $p\in PV$,有 $V(r(m))(p)\in\{\text{true,false}\}$,即 V 是 PV 中每一命题在每一状态下的真值赋值函数.

因此,对于一个同步时态认知模型,令 $i\in A$ 和$\Gamma\subseteq A$,则认知算子 K_i,D_Γ和 E_Γ用到的同步认知关系分别定义为 $\overset{T}{\sim_i},\overset{T}{\sim_\Gamma^D}=\bigcap_{i\in}\overset{T}{\sim_i}\Gamma$ 和 $\overset{T}{\sim_\Gamma^E}=\bigcup_{i\in\Gamma}\overset{T}{\sim_i}$,而 C_Γ的同步认知关系是 $\overset{T}{\sim_\Gamma^E}$ 的传递闭包,用 $\overset{T}{\sim_\Gamma^C}$ 表示.容易证明:对于同步模型中任意的两个 $point(r,n)$和(r',n'),$(r,n)\overset{T}{\sim_\Gamma^Y}(r',n')$ 当且仅当 $(r,n)\sim_\Gamma^E(r',n')$ 并且 $n=n'$,其中,$Y\in\{D,E,C\}$.如无特别说明,下文简单地用“模型”来表示同步时态认知模型.

2 $ECKL_n$ 逻辑及其子集

本节将在 Emerson 和 Clarke 的时态逻辑 CTL*[1]中扩展上一节引入的 K_i(知道)、D_Γ(分布知识)、E_Γ(全都知道)和 C_Γ(公共知识)4 个认知算子.为了便于处理存在模型检测问题,我们又引入了与这些认知算子对应的 4

个对偶算子.

令 $Y\in\{K_i,D_\Gamma,E_\Gamma,C_\Gamma\}$,$\varphi$是一个公式,则 $\overline{Y}$ 是 Y 的对偶认知算子,并且 $Y\varphi\equiv\neg\overline{Y}\neg\varphi$.通过以上扩展,我们得到一个新的时态认知逻辑 ECKL_n.

ECKL_n 逻辑的时态部分包含路径量词和线性时间模态词两种算子.路径量词有 A(在所有计算路径)和 E(在某条计算路径).线性时间算子包含 X(下一步)、F(最终)、G(总是)和 U(直到).显然,ECKL_n 逻辑的时态部分包含线性时态逻辑 LTL 和分支时态逻辑 CTL.设 PV 是包含 true 的命题变量集合,ECKL_n 公式由状态公式(在一个特定的状态中为 true)和路径公式(在一条特定的路径中为 true)组成.其语法由下列规则给出:

- 如果 $p\in PV$,则 p 是一个状态公式;
- 如果α和β是状态公式,则$\neg\alpha,\alpha\wedge\beta$和$\alpha\vee\beta$也是状态公式;
- 如果α是一个状态公式,则α也是一个路径公式;
- 如果α和β是路径公式,则$\neg\alpha,\alpha\wedge\beta$和$\alpha\vee\beta,X\alpha,F\alpha,G\alpha$和$\alpha U\beta$是路径公式;
- 如果α是一个路径公式,则 $E\alpha,\overline{K}_i\alpha,\overline{D}_\Gamma\alpha,\overline{E}_\Gamma\alpha$ 和 $\overline{C}_\Gamma\alpha$ 是一个状态公式,其中,$i\in A,\Gamma\subseteq A$.

ECKL_n 逻辑是由上述规则生成的所有状态公式的集合.如果φ和 ψ是 ECKL_n 公式,则 $\varphi\to\psi\equiv\neg\varphi\vee\psi$, $A\varphi\equiv\neg E\neg\varphi$, $Y\varphi\equiv\neg\overline{Y}\neg\varphi$,其中,$Y\in\{K_i,D_\Gamma,E_\Gamma,C_\Gamma\}$.如果约束 ECKL_n 逻辑的否定词只能作用在命题变量上,则称这样的逻辑为 EECKL_n 逻辑.在上述语法中,将$\neg\alpha$改为$\neg p$ 即可得到 EECKL_n 逻辑的语法.而 AECKL_n 逻辑的语言定义为$\{\neg\varphi|\varphi\in\mathrm{EECKL}_n\}$.

定义 2(ECKL_n 逻辑的同步语义). 令 M 是一个模型,(r,n)是 M 的一个 *point*,α,β是 ECKL_n 公式.$(M,r,n)|=\alpha$表示α在 M 的 *point*(r,n)为 true.这里省略 M,如无特别说明,均隐含 M.关系$|=$归纳定义如下:

$(r,n)|=p$ iff $V(r(n))(p)=\text{true}$;

$(r,n)|=\neg p$ iff $(r,n)|\neq p$;

$(r,n)|=X\alpha$ iff $(r,n+1)|=\alpha$;

$(r,n)|=G\alpha$ iff $\forall_{n'\geq n}(r,n')|=\alpha$;

$(r,n)|=E\alpha$ iff 存在一个 *run* r'和时间 n',使得 $r(n)=r'(n')$并且$(r',n')|=\alpha$;

$(r,n)|=\overline{K}_i\alpha$ iff 存在一个 *run* r'和时间 n',使得$(r,n)\sim_i(r',n')$并且 $n=n'$并且$(r',n')|=\alpha$;

$(r,n)|=\overline{Y}_\Gamma\alpha$ iff 存在一个 *run* r'和时间 n',使得 $(r,n)\sim_\Gamma^Y(r',n')$ 并且 $n=n'$并且$(r',n')|=\alpha$,其中,$Y\in\{D,E,C\}$.

$(r,n)|=\alpha\vee\beta$ iff $(r,n)|=\alpha$或者$(r,n)|=\beta$;

$(r,n)|=\alpha\wedge\beta$ iff $(r,n)|=\alpha$并且$(r,n)|=\beta$;

$(r,n)|=F\alpha$ iff $\exists_{n'\geq n}(r,n')|=\alpha$;

$(r,n)|=\alpha U\beta$ iff $\exists_{n'\geq n}((r,n')|=\beta$并且$\forall_{n\leq l<n'}(r,n')|=\alpha)$;

定义 3(有效性). 一个 ECKL_n 公式φ在模型 $M=((S,s_0,T,\overset{T}{\sim}_1,\ldots,\overset{T}{\sim}_n),V)$ 中是有效的(记为 $M|=\varphi$)当且仅当$(M,r,0)|=\varphi$,其中$(r,0)=s_0$,即φ在模型 M 的初始状态为 true.

3 EECKL_n 逻辑的有界语义

根据 ECKL_n 逻辑同步语义,本节将在 ECTL*逻辑的有界语义[8]中加入时间域,并扩展认知算子的语义解释,从而得到 EECKL_n 逻辑的有界语义.

令 M 为一个模型,一条 *path* 是一个可能的无限状态序列$\pi=\{s_0,s_1,\ldots\}$,使得对于任意的 $i\geq0$,有 $s_i\in S$ 对应的状态集合并且$(s_i,s_{i+1})\in T$.注意:迁移关系 T 是由 *point* 定义的,而这里的 s_i 和 s_{i+1} 是状态而不是 *point*.为方便起见,我们在下文中出现 *point* 的地方也允许用状态表示 *point*,不再加以说明.令 k 是一个正自然数,则一条 *k-path* 是一个长度为 k 的 *path*,$\pi_k=\{s_0,\ldots,s_k\}$.另外,用$\pi_k(i)$表示π_k 中的状态 $s_i(0\leq i\leq k)$.对于一条 k-path π_k,如果存在某个 $l\in\{0,\ldots,k\}$使得$(\pi_k(k),\pi_k(l))\in T$,则称π_k 是一条(k,l)-loop.如果存在 $0\leq l\leq k$ 使得π_k 是一条(k,l)-loop,则简单地称π_k 为 *k-loop*.显然,一条(k,l)-loop 可以表示具有无穷多状态的路径.

为了将一个 EECKL_n 公式的存在模型检测问题转化为有界模型检测问题和 SAT 问题,我们只考虑由模型 M 中所有 k-path 组成的部分状态空间,这些 k-path 可以从 M 的任意状态开始展开.

定义 4(*k*-model). 令 $M=((S,s_0,T,\overset{T}{\sim}_1,\ldots,\overset{T}{\sim}_n),V)$ 是一个模型,k 是一个正自然数,则 M 的一个 k-model 是一个结构 $M_k=((S,s_0,P_k,\overset{T}{\sim}_1,\ldots,\overset{T}{\sim}_n),V)$,其中,P_k 是 M 中所有 *k-path* 的集合.

已知 M 的一个 k-model M_k,由于时态算子的有界语义定义依赖于 M_k 中当前考虑的一条 k-path π_k 是否为一个 k-loop,我们预先定义一个函数 $loop(\pi_k)=\{l|0\le l\le k$ 并且$(\pi_k(k),\pi_k(l))\in T\}$.

由于 k-model M_k 的一条 k-path π_k 可以看作是 M 的一个 run r 的一部分,因此,我们可将 r 的一部分映射到π_k 上.假设存在 $n\ge 0$ 和 $0\le m\le k$,使得 $r(n)=\pi_k(m)$.对于某个时间 $c\ge n$,我们计算π_k 中的状态$\pi_k(i)$所处的位置 i,使得 $\pi_k(i)=r(c)$.也就是说,M_k 的状态$\pi_k(i)$表示 M 的状态 $r(c)$.$r(c)$映射到π_k 上的状态位置 i 可用如下定义计算:

定义 5(状态位置函数). 已知一个模型 M 和 M 的一个 k-model M_k.令 r 为 M 的一个 run,与 r 相对应的π_k 是 M_k 的一条 k-path,并且存在 $n\ge 0$ 和 $0\le m\le k$,使得 $r(n)=\pi_k(m)$.令时间 $c\ge n$ 和 $l\le k$,函数

$$pos(n,m,k,l,c):=\begin{cases} m+c-n, & \text{if} \quad c\le n+k-m \\ l+(c-n-l+m)\%(k-l+1), & \text{else if} \ \ l\in loop(\pi_k) \end{cases}.$$

返回π_k 中满足$\pi_k(pos(n,m,k,l,c))=r(c)$的状态的位置,其中%是取余运算.这样,$M_k$ 的状态$\pi_k(pos(n,m,k,l,c))$就表示 M 的状态 $r(c)$.进一步地,我们为认知算子定义状态位置函数 $f(k,l,c):=pos(0,0,k,l,c)$.

定义 5 的 pos 函数可将 run r 中时间大于 $n+k-m$ 的任何状态映射到 M_k 的一条只有有限个状态的(k,l)-loop π_k 上.换句话说,可用 M_k 的一条(k,l)-loop 表示 M 的一条完整的(或部分的)run.然后,我们适当修改文献[8]的 ECTL*有界语义,并加入时间域及认知算子的有界同步语义,得到 EECKL$_n$ 逻辑有界同步语义,其形式定义如下:

定义 6(**EECKL$_n$** 逻辑的有界同步语义). 令 M_k 是一个 k-model,α,β是 EECKL$_n$ 公式.如果α是一个状态公式,则 $M_k,[(\pi_k,l),m,c]|=\alpha$表示$\alpha$在 M_k 的状态π_k和时间 c 为 true;如果α是一个路径公式,则 $M_k,[(\pi_k,l),m,c]|=\alpha$表示$\alpha$在 M_k 的 k-path π_k 的后缀中为 true,而这个π_k 的后缀起始于状态$\pi_k(m)$和时间 c.这里省略 M_k,如无特别说明,均隐含 M_k.关系|=归纳定义如下:

$[(\pi_k,l),m,c]|=p$ iff $V(\pi_k(m))(p)$=true, $[(\pi_k,l),m,c]|=\alpha\vee\beta$ iff $[(\pi_k,l),m,c]|=\alpha$ or $[(\pi_k,l),m,c]|=\beta$,

$[(\pi_k,l),m,c]|\ne p$ iff $V(\pi_k(m))(p)$=false, $[(\pi_k,l),m,c]|=\alpha\wedge\beta$ iff $[(\pi_k,l),m,c]|=\alpha$ and $[(\pi_k,l),m,c]|=\beta$,

$[(\pi_k,l),m,c]|=E\alpha$ iff $\exists_{\pi'_k\in P_k}(\pi'_k(0)=\pi_k(m)$ and $\exists_{0\le l'\le k}[(\pi'_k,l'),0,c]|=\alpha)$,

$$[(\pi_k,l),m,c]|=X\alpha \text{ iff } \begin{cases} l\notin loop(\pi_k): \text{ if } \ m\ge k \ \ \text{then false else } [(\pi_k,l),m+1,c+1]|=\alpha, \\ l\in loop(\pi_k): \text{ if } \ m\ge k \ \ \text{then } [(\pi_k,l),l,c+1]|=\alpha \ \text{ else } [(\pi_k,l),m+1,c+1]|=\alpha. \end{cases}$$

$$[(\pi_k,l),m,c]|=F\alpha \text{ iff } \begin{cases} l\notin loop(\pi_k): \exists_{m\le i\le k}\,[(\pi_k,l),i,c+i-m]|=\alpha, \\ l\in loop(\pi_k): \exists_{m\le i\le k}\,[(\pi_k,l),i,c+i-m]|=\alpha \ \text{ or } \ \exists_{l\le i<m}\,[(\pi_k,l),i,c+k-m+1+i-l]|=\alpha. \end{cases}$$

$$[(\pi_k,l),m,c]|=G\alpha \text{ iff } \begin{cases} l\notin loop(\pi_k): \text{false}, \\ l\in loop(\pi_k) \text{ and } l\ge m: \forall_{m\le i\le k}\,[(\pi_k,l),i,c+i-m]|=\alpha, \\ l\in loop(\pi_k) \text{ and } l<m: \forall_{m\le i\le k}\,[(\pi_k,l),i,c+i-m]|=\alpha \ \text{ and} \\ \qquad\qquad \forall_{l\le i<m}\,[(\pi_k,l),i,c+k-m+1+i-l]|=\alpha. \end{cases}$$

$$[(\pi_k,l),m,c]|=\alpha\mathrm{U}\beta \text{ iff } \begin{cases} l\notin loop(\pi_k): \exists_{m\le i\le k}([(\pi_k,l),i,c+i-m]|=\beta \text{ and } \forall_{m\le j<i}\,[(\pi_k,l),j,c+j-m]|=\alpha), \\ l\in loop(\pi_k): \exists_{m\le i\le k}([(\pi_k,l),i,c+i-m]|=\beta \text{ and } \forall_{m\le j<i}\,[(\pi_k,l),j,c+j-m]|=\alpha) \text{ or} \\ \qquad \exists_{l\le i<m}([(\pi_k,l),i,c+k-m+1+i-l]|=\beta \text{ and } \forall_{m\le j\le k}\,[(\pi_k,l),j,c+j-m]|=\alpha \\ \qquad \text{and } \forall_{l\le j<i}\,[(\pi_k,l),j,c+k-m+1+j-l]|=\alpha). \end{cases}$$

$[(\pi_k,l),m,c]|=Y\alpha$ iff $\exists\pi'_k\in P_k$ such that $\pi'_k(0)=s_0$ and

$$\begin{cases} \text{if } c\le k \text{ then } \pi_k(m)\overset{Y}{\sim}\pi'_k(c) \text{ and } \exists_{0\le l'\le k}\,[(\pi'_k,l'),c,c]|=\alpha \\ \qquad \text{else } \exists_{0\le l'\le k}\,(l'\in loop(\pi'_k) \text{ and } \pi_k(m)\overset{Y}{\sim}\pi'_k(f(k,l',c)) \text{ and } [(\pi'_k,l'),f(k,l',c),c]|=\alpha), \end{cases}$$

$$\text{where } (Y,\overset{Y}{\sim})\in\{(\overline{K}_i,\sim_i),(\overline{D}_\Gamma,\sim^D_\Gamma),(\overline{E}_\Gamma,\sim^E_\Gamma)\}.$$

$[(\pi_k,l),m,c]|=\overline{C}_\Gamma\alpha$ iff $[(\pi_k,l),m,c]|=\vee_{i=1}^{k}(\overline{E}_\Gamma)^i\alpha$.

需要指出的是:当检测一个时态公式在状态$\pi_k(m)$和时间 c 是否为 true 时,如果 $loop(\pi_k)\neq\varnothing$,那么,即使当前考虑的一个状态在$\pi_k$中的位置小于 m,时间 c 也总是递增的.另外,当检测一个认知公式 $\overline{K}_i\alpha$ 在状态$\pi_k(m)$和时间 c 是否为 true 时,我们考虑是否存在一条从系统的初始状态开始展开的 k-path π'_k,使得 π'_k 中存在一个状态 s',导致公式α在状态 s'为 true,而$\pi_k(m)$与 s'对于智能体 i 来说是不可辨别的,并且状态 s'当前对应的时钟与时间 c 相等.状态 s'的获取需要根据当前考虑的 k-path 和时间用上述计算状态位置的方法获得.这样的定义保证了认知算子的有界同步语义与定义 2 的同步语义的一致性.相关正确性证明在第 4 节给出.其余认知公式的语义解释与 $\overline{K}_i\alpha$ 类似,这里不再赘述.如无特别说明,下文简单地用"有界语义"来表示有界同步语义.

定义 **7**(**EECKL$_n$** 逻辑有界语义的有效性). 一个 EECKL$_n$ 公式φ在一个 k-model M_k 中是有效的(记为 $M\models_k\varphi$)当且仅当存在某个 $0\le l\le k$ 使得 $M_k,[(\pi_k,l),0,0]\models\varphi$,其中,$\pi_k(0)$等于 s_0 对应的状态,即φ在 M_k 的初始状态为 true.

4 EECKL$_n$ 逻辑有界语义的正确性

本节将要证明 EECKL$_n$ 模型检测问题($M\models\varphi$)可以转化为 EECKL$_n$ 有界模型检测问题($M\models_k\varphi$).这里省略一些与文献[8,9]类似的证明.值得注意的是:由于 EECKL$_n$ 逻辑有界语义的时态部分在文献[8]的 ECTL*逻辑有界语义中扩展了时间域,而认知部分也考虑了时间域,因此,在这些省略的证明中需要考虑时间域.本节主要针对认知部分的正确性进行证明.首先定义 EECKL$_n$ 公式长度.

定义 **8**(**EECKL$_n$** 公式长度). 令φ是一个 EECKL$_n$ 公式,φ的长度(记为 $len(\varphi)$)归纳定义如下:

- $len(\varphi)=len(\neg\varphi)=0$, 如果$\varphi\in PV$;
- $len(\varphi)=len(\alpha)+len(\beta)+1$, 如果$\varphi=\alpha Y\beta$,其中,$Y\in\{\vee,\wedge,U\}$;
- $len(\varphi)=len(\alpha)+1$, 如果$\varphi=Z\alpha$,其中, $Z\in\{E,X,F,G,\overline{K}_i,\overline{D}_\Gamma,\overline{E}_\Gamma,\overline{C}_\Gamma\}$.

通过以下证明可以看到:定义 6 的有界同步语义与定义 2 的同步语义是等价的.

引理 **1**. 已知一个模型 M 和一个 LTL 公式α,令 $Y\in\{E,\overline{K}_i\}$ $(i\in A)$.如果$(M,r,0)\models Y\alpha$,则存在 $k\le|M|\cdot|\alpha|\cdot2^{|\alpha|}$和 $0\le l\le k$,使得 $r(0)=\pi_k(0)=s_0$ 对应的状态,并且 $M_k,[(\pi_k,l),0,0]\models Y\alpha$.其中,$\pi_k$是 k-model M_k 的一条 k-path;$|M|=|S|+|T|$是 M 的可达状态及状态迁移关系个数之和.

证明:1) 令 $Y=E$,则此引理可直接由文献[8]的引理 3 得出.下面考虑 $Y=\overline{K}_i$ 的情况.

2) 令 $Y=\overline{K}_i$ 并且$(r,0)$是 M 的一个初始 *point*.由定义 2 的同步语义,有 $(M,r,0)\models\overline{K}_i\alpha$ 当且仅当存在一个 *run* r' 和时间 c',使得$(r,0)\sim_i(r',c')$并且 $c'=0$ 并且$(M,r',c')\models\alpha$.我们可以选择一条与 *run* r'相对应的路径π,并且满足 $\pi(0)=r'(c')$.根据$(M,r',c')\models\alpha$的语义,我们有 LTL 公式α从状态$\pi(0)$开始,在路径π上为 true.又因为 $c'=0$,我们有 $\pi(0)=s_0$,所以,$(M,r',c')\models\alpha$的验证也就是确定α在 M 中是否是存在有效的,进而, $(M,r,0)\models\overline{K}_i\alpha$ 的验证可以看作是 LTL 公式α在 M 中的一个存在模型检测问题[7].

对于 LTL 公式的存在模型检测问题,可以将其转化为自动机验证问题.首先,构造系统模型 M 的自动机;然后,根据文献[11]构造 LTL 公式α的 Büchi 自动机,这个 Büchi 自动机至多有$|\alpha|\cdot2^{|\alpha|}$个状态;最后,构造这两个自动机的积自动机 P,并检测 P 所接受的语言是否为空.因此,如果α在 M 中是存在有效的,那么,在自动机 P 中存在一条起始于初始状态的路径 r',并且 r'的尾部是一个包含在接受状态强连通组件中的环路.选择路径 r'作为一条 k-loop,其中 k 不超过$|M|\cdot|\alpha|\cdot2^{|\alpha|}$($P$ 包含的最大状态数);然后,将路径 r'映射到模型 M 并获得一个(k,l')-loop π'_k .由上述关于同步语义的结果,我们有 π'_k 满足 $s_0\sim_i\pi'_k(0)$,并且存在某个 $0\le l'\le k$,使得 $M_k,[(\pi'_k,l'),0,0]\models\alpha$.因此,由定义 6 的有界语义,我们有 $M_k,[(\pi_k,l),0,0]\models\overline{K}_i\alpha$. □

引理 **2**. 令 M 是一个模型,φ是一个 EECKL$_n$ 公式.如果$(M,r,c)\models\varphi$,则存在 $k\le|M|\cdot|\varphi|\cdot2^{|\varphi|}$和 $l,m\in\{0,\ldots,k\}$,使得 $r(c)=\pi_k(m)$并且 $M_k,[(\pi_k,l),m,c]\models\varphi$.其中,$\pi_k$是 k-model M_k 的一条 k-path.

证明:如果φ是一个路径公式,那么,此引理可由文献[8]的引理 1 中加入时间域后进行类似的证明而得到,这里不再赘述.下面主要考虑φ是认知公式的情况.

如果φ是一个状态公式,则施归纳于公式φ的长度来证明.如果φ是一个命题变量或其否定,此引理直接成立.

令$(M,r,c)|=\varphi$,并假设对于φ的所有真子公式,此引理均成立.容易证明当$\varphi=\alpha\vee\beta$或$\varphi=\alpha\wedge\beta$时此引理成立.

1) 令α是一个 LTL 公式,考虑下列情况:

- 令$\varphi=\overline{E}_\Gamma\alpha$.由于$\overline{E}_\Gamma\alpha=\vee_{i\in A}\overline{K}_i\alpha$,由归纳假设及引理 1 中 $Y=\overline{K}_i$ 的情况可知,对于某个智能体 $i\in A$,存在某个$k\le|M|\cdot|\alpha|\cdot 2^{|\alpha|}$和$0\le l\le k$,使得$M_k,[(\pi_k,l),m,c]|=\overline{K}_i\alpha$,其中,$r(c)=\pi_k(m)$.再由布尔连接词的情况可知:当$\varphi=\overline{E}_\Gamma\alpha$时,此引理成立.
- 令$\varphi=\overline{D}_\Gamma\alpha$.证明类似于引理 1 中 $Y=\overline{K}_i$ 的情况.
- 令$\varphi=\overline{C}_\Gamma\alpha$.由于$(M,r,c)|=\overline{C}_\Gamma\alpha$当且仅当$(M,r,c)|=\vee_{i\in|S|}\{\overline{E}_\Gamma\}^i\alpha$,其中$|S|$是 M 的可达状态数,因此,根据归纳假设及上述情况可知,此引理成立.

2) 令α不是一个"纯的"LTL 公式,即α包含以路径量词或认知模态词为主算子的状态子公式.为了处理任意类型的状态子公式,我们扩展文献[1]中的状态标记(labelling)技术如下:首先,令 $Y,Y_1,\ldots,Y_n$, $Z\in\{E,\overline{K}_a,\overline{D}_\Gamma,\overline{E}_\Gamma,\overline{C}_\Gamma\}$,其中 $a\in A,\Gamma\subseteq A$,并且令$\varphi=Y\alpha$.假设α所有的最大子公式是 $Y_1\alpha_1,\ldots,Y_n\alpha_n$(也就是说,每个 $Y_i\alpha_i$ $(i=1,\ldots,n)$是 $Y\alpha$的一个状态子公式,但不是 $Y\alpha$的任一子公式 $Z\beta$的子公式,其中,$Z\beta$不包括 $Y\alpha$和 $Y_1\alpha_1,\ldots,Y_n\alpha_n$).

然后,对于每个 $Y_i\alpha_i$ $(i=1,\ldots,n)$,引入一个新的原子命题 p_i,并且对于 M_k 中所有的状态 s,只要 $Y_i\alpha_i$ 在 s 成立,那么就在 s 中加入标记 p_i,并用原子命题 p_i 替换 $Y_i\alpha_i$.这样,α就替换成一个"纯的"LTL 公式 α'.另外,由同步语义的定义,$(M,r,c)|=\overline{E}_\Gamma\alpha'\mid\overline{D}_\Gamma\alpha'\mid\overline{C}_\Gamma\alpha'$的验证也可以看作是 LTL 公式$\alpha'$在 M 中的一个存在模型检测问题.证明类似于引理 1 中 $Y=\overline{K}_a$ 的情况,这里不再赘述.因此,由上述情况及引理 1,有$(M,r,c)|=Y\alpha$蕴涵 $M_k,[(\pi_k,l),m,c]|=Y\alpha$. □

引理 **3**. 令 M 是一个模型,φ是一个 EECKL_n 公式.以下两个条件均成立:

(1) $M_k,[(\pi_k,l),m,c]|=\varphi$ 蕴涵 对于任意的 $k'\ge k$,有 $M_{k'},[(\pi_k,l),m,c]|=\varphi$;

(2) $M_k,[(\pi_k,l),m,c]|=\varphi$ 蕴涵 $(M,r,c)|=\varphi$,其中,$r(c)=\pi_k(m)$.

证明:可直接施归纳于公式φ的长度来证明. □

定理 **1**. 令 M 是一个模型,φ是一个 EECKL_n 公式.存在 $k\le|M|\cdot|\varphi|\cdot 2^{|\varphi|}$,使得 $M|=\varphi$当且仅当 $M|=_k\varphi$.

证明:直接由引理 2 和引理 3 证明. □

5 EECKL_n 公式转换

本节给出在同步解释系统中有界模型检测 $\mathrm{AECKL}_n/\mathrm{EECKL}_n$ 逻辑的算法.EECKL_n 公式转换的目的是根据 EECKL_n 逻辑有界语义,将一个 EECKL_n 公式在同步多智体系统中的有界模型检测问题转换为一个命题公式的可满足性问题.该算法在文献[8,9]的基础上扩展时间域,并加入对认知公式的转换.首先引入下面两个定义.

定义 **9**(子模型). 令 $M_k=((S,s_0,P_k,\overset{T}{\sim}_1,\ldots,\overset{T}{\sim}_n),V)$ 是一个 k-model.对于一个结构 $M'_k=((S',s_0,P'_k,\overset{T}{\sim}'_1,\ldots,\overset{T}{\sim}'_n),V')$,如果 $P'_k\subseteq P_k$, $states(P'_k)\subseteq S'\subseteq S$ 对应的状态集合,$\overset{T}{\sim}'_i=\overset{T}{\sim}_i\cap(S'\times S')$ $(i\in A)$ 并且 $V'=V|_{S'}$,则称 M'_k 是 M_k 的一个子模型.其中,$states(P'_k)$ 表示 P'_k 的可达状态集合;$V|_{S'}$将 V 的定义域 S 约束为 S'.

由于定义 9 将文献[9]的子模型的认知可达关系更改为同步的认知可达关系,因此它们是不同的.EECKL_n 逻辑在 M_k 的子模型 M'_k 中的有界语义与 M_k 的相同.我们在文献[8]的定义 10 的函数 f_k 中加入对认知算子的计算,以确定足够用来在子模型 M'_k 中验证一个 EECKL_n 公式所需的 k-path 个数,使得 EECKL_n 公式φ在 M_k 中的有效性验证等价于φ在 M'_k 中的有效性验证.函数 f_k 的定义如下:

定义 **10**. f_k 是一个从 EECKL_n 公式集合到自然数域的函数,其定义如下:

$f_k(p)=f_k(\neg p)=0$,其中 $p\in PV$;$f_k(\alpha\vee\beta)=\max(f_k(\alpha),f_k(\beta))$;$f_k(\alpha\wedge\beta)=f_k(\alpha)+f_k(\beta)$;$f_k(X\alpha)=f_k(F\alpha)=f_k(\alpha)$;

$f_k(G\alpha)=(k+1)\cdot f_k(\alpha)$;$f_k(\alpha U\beta)=k\cdot f_k(\alpha)+f_k(\beta)$; $f_k(\overline{C}_\Gamma\alpha)=f_k(\alpha)+k$;$f_k(Y\alpha)=f_k(\alpha)+1$,其中,$Y\in\{E,\overline{K}_i,\overline{D}_\Gamma,\overline{E}_\Gamma\}$.

如前所述,本算法的基本思想是:公式φ的有效性可以通过验证一个命题公式$[M,\varphi]_k:=[M^{\varphi,s_0}]_k\wedge[\varphi]_{M_k}$的可

满足性的方法来确定.其中,$[M^{\varphi,s_0}]_k$ 表示当前考虑的子模型,而 $[\varphi]_{M_k}$ 是φ在 M_k 中成立的必要约束条件.$[M,\varphi]_k$ 的可满足性可用 SAT 解算器(如 ZChaff,PROVER 或 GRASP 等)来验证.

下面阐述公式转换的具体方法.由于在获取同步系统中的智能体知识时,在某一 k-path 中与某一时间对应的状态是通过第 3 节中计算状态位置的方法获得的,避免了为时间域而扩展通常的时态认知模型的状态及迁移关系编码,因此,对于某一指定的 EECKL_n 规范φ,我们可以沿用文献[9]的解释系统编码方法对同步时态认知模型进行编码.首先,在不考虑环境的情况下,系统的全局状态定义为 $G=\times_{i=1}^{n}L_i$.假设对于每一 $i\in\{1,\ldots,n\}$,有 $L_i\subseteq\{0,1\}^{n_i}$,其中 $n_i=\lceil\log|L_i|\rceil$.令 $m=\sum_{i=1}^{n}n_i$.这样,可以用全局状态变量 $w=(w[0],\ldots,w[m-1])$来表示某个全局状态 $s=(l_1,\ldots,l_n)=(s[0],\ldots,s[m-1])$,其中 $w[i](0\le i<m)$是命题变量.令 $I_i(1\le i\le n)$是智能体 i 的局部状态位的索引集合,即 $I_1=\{0,\ldots,n_1-1\},\ldots,I_n=\{m-n_n,\ldots,m-1\}$.另外,用全局状态变量的有限序列$(w_0,\ldots,w_k)$来表示一条 k-path,这个序列称为符号 k-path.为了构造命题公式 $[M^{\varphi,s_0}]_k$,需要构造 $f_k(\varphi)$条符号 k-path,每条符号 k-path $j(1\le j\le f_k(\varphi))$由$(w_{0,j},\ldots,w_{k,j})$表示.其中,$w_{i,j}(0\le i\le k)$是全局状态变量,表示第 j 条符号 k-path 的第 i 个状态.

令 w,v 是两个全局状态变量,PV 是命题变量集合,F 是 PV 上的公式集合.函数 $lit:\{0,1\}\times PV\to F$ 定义为 $lit(1,p)=p$ 和 $lit(0,p)=\neg p$.然后定义下列命题公式:

- $I_s(w):=\wedge_{i=0}^{m-1}lit(s[i],w[i])$.此公式用全局状态变量 w 表示状态 s.
- $T(w,v)$是 w,v 上的一个公式.令 s 和 s'分别是 w 和 v 的一个赋值,则 $T(w,v)$为 true 当且仅当$(s,s')\in T$.
- $L_{k,j}(l):=T(w_{k,j},w_{l,j})$.当第 j 条 k-path 是一条(k,l)-loop 时,$L_{k,j}(l)$为 true.
- $p(w)$是 w 上的一个公式.对于 w 的一个赋值 s,$p(w)$=true 当且仅当 $V(s)(p)$=true,其中,$p\in PV\cup$true.
- $H(w,v):=\wedge_{i=0}^{m-1}(w[i]\Leftrightarrow v[i])$.此公式表示 w 和 v 是逻辑等价的,即它们表示相同的状态.
- $H_i(w,v):=\wedge_{j\in I_i}(w[j]\Leftrightarrow v[j])$,其中 $i\in A$.此公式表示 w 和 v 中的 i-local 状态(智能体 i 的局部状态)是逻辑等价的,即 w 和 v 中的局部状态对于智能体 i 来说是相同的.

定义 11(状态迁移关系的公式转换). 令 $M_k=((S,s_0,P_k,\overset{T}{\sim_1},\ldots,\overset{T}{\sim_n}),V)$ 是模型 M 的一个 k-model,$s\in S$ 对应的状态集合,φ是一个 EECKL_n 公式,命题公式$[M^{\varphi,s}]_k$ 定义如下:

$$[M^{\varphi,s}]_k:=I_s(w_{0,0})\wedge\wedge_{1\le j\le f_k(\varphi)}\wedge_{0\le i<k}T(w_{i,j},w_{i+1,j}).$$

对于任意的 $0\le i\le k$ 和 $1\le j\le f_k(\varphi)$,$w_{i,j}$ 是全局状态变量,表示第 j 条符号 k-path 的第 i 个状态.

由定义 11 可知,命题公式$[M^{\varphi,s}]_k$ 从状态 s 开始,展开足以用来验证公式φ的 $f_k(\varphi)$条符号 k-path,并用全局状态变量 $w_{0,0}$ 表示状态 s,同时约束这 $f_k(\varphi)$条符号 k-path 均满足状态迁移关系.因此,k-model M_k 的状态迁移关系可由命题公式 $[M^{\varphi,s_0}]_k$ 来表示.通过定义 12,我们可以将一个 EECKL_n 公式φ转换为一个命题公式.

定义 12(**EECKL_n 公式转换**). 已知一个 k-model M_k,φ,α和β是 EECKL_n 公式.令 $L_{k,i}:=\vee_{l'=0}^{k}L_{k,i}(l')$ 并且 $x\in\{k,(k,l)\}$.我们用 $[\alpha]_k^{[m,n,c]}$ 表示在状态 $w_{m,n}$ 和时间 c 将α转换为一个命题公式.用 $[\alpha]_{k,l}^{[m,n,c]}$ 表示根据(k,l)-loop 的有界语义,在状态 $w_{m,n}$ 和时间 c 将α转换为一个命题公式.当 $x=k$ 时,$[\alpha]_x^{[m,n,c]}$ 表示 $[\alpha]_k^{[m,n,c]}$;否则,当 $x=(k,l)$时,$[\alpha]_x^{[m,n,c]}$ 表示 $[\alpha]_{k,l}^{[m,n,c]}$.EECKL_n 公式转换归纳定义如下:

$$[p]_x^{[m,n,c]}:=p(w_{m,n}),\ [\neg p]_x^{[m,n,c]}:=\neg p(w_{m,n}),\ [\alpha\vee\beta]_x^{[m,n,c]}:=[\alpha]_x^{[m,n,c]}\vee[\beta]_x^{[m,n,c]},\ [\alpha\wedge\beta]_x^{[m,n,c]}:=[\alpha]_x^{[m,n,c]}\wedge[\beta]_x^{[m,n,c]},$$

$$[X\alpha]_k^{[m,n,c]}:=\text{if } m<k \text{ then } [\alpha]_k^{[m+1,n,c+1]} \text{ else false},\ [X\alpha]_{k,l}^{[m,n,c]}:=\text{if } m<k \text{ then } [\alpha]_{k,l}^{[m+1,n,c+1]} \text{ else } [\alpha]_{k,l}^{[l,n,c+1]},$$

$$[F\alpha]_k^{[m,n,c]}:=\vee_{i=m}^{k}[\alpha]_k^{[i,n,c+i-m]},\ [F\alpha]_{k,l}^{[m,n,c]}:=\vee_{i=m}^{k}[\alpha]_{k,l}^{[i,n,c+i-m]}\vee\vee_{i=l}^{m-1}[\alpha]_{k,l}^{[i,n,c+k-m+1+i-l]},$$

$$[G\alpha]_k^{[m,n,c]}:=\text{false},\ [G\alpha]_{k,l}^{[m,n,c]}:=\text{if } l\ge m \text{ then } \wedge_{i=m}^{k}[\alpha]_{k,l}^{[i,n,c+i-m]} \text{ else } \wedge_{i=m}^{k}[\alpha]_{k,l}^{[i,n,c+i-m]}\wedge\wedge_{i=l}^{m-1}[\alpha]_{k,l}^{[i,n,c+k-m+1+i-l]},$$

$$[\alpha\mathrm{U}\beta]_k^{[m,n,c]}:=\vee_{i=m}^{k}([\beta]_k^{[i,n,c+i-m]}\wedge\wedge_{j=m}^{i-1}[\alpha]_k^{[j,n,c+j-m]}),$$

$$[\alpha\mathrm{U}\beta]_{k,l}^{[m,n,c]}:=\vee_{i=m}^{k}([\beta]_{k,l}^{[i,n,c+i-m]}\wedge\wedge_{j=m}^{i-1}[\alpha]_{k,l}^{[j,n,c+j-m]})\vee\vee_{i=l}^{m-1}([\beta]_{k,l}^{[i,n,c+k-m+1+i-l]}\wedge\wedge_{j=m}^{k}[\alpha]_{k,l}^{[j,n,c+j-m]}\wedge$$
$$\wedge_{j=l}^{i-1}[a]_{k,l}^{[j,n,c+k-m+1+j-l]}),$$

$$[E\alpha]_x^{[m,n,c]} := \vee_{i=1}^{f_k(\varphi)}(H(w_{m,n},w_{0,i}) \wedge ((\neg L_{k,i} \wedge [\alpha]_k^{[0,i,c]}) \vee \vee_{l'=0}^{k}(L_{k,i}(l') \wedge [\alpha]_{k,l'}^{0,i,c}))),$$

$$[Y\alpha]_x^{[m,n,c]} := \begin{cases} \text{if } c \le k \text{ then} \vee_{i=1}^{f_k(\varphi)}(I_{s_0}(w_{0,i}) \wedge Z(H_a(w_{m,n},w_{c,i})) \wedge ((\neg L_{k,i} \wedge [\alpha]_k^{[c,i,c]}) \vee \vee_{l'=0}^{k}(L_{k,i}(l') \wedge [\alpha]_{k,l'}^{[c,i,c]}))) \\ \quad \text{else} \vee_{i=1}^{f_k(\varphi)}(I_{s_0}(w_{0,i}) \wedge \vee_{l'=0}^{k}(L_{k,i}(l') \wedge Z(H_a(w_{m,n},w_{f(k,l',c),i})) \wedge [\alpha]_{k,l'}^{[f(k,l',c),i,c]})), \end{cases}$$

其中:$(Y,Z) \in \{(\overline{K}_a,\perp);(\overline{D}_\Gamma,\wedge_{a\in\Gamma});(\overline{E}_\Gamma,\vee_{a\in\Gamma})\}$;$\perp$表示$Z$为空.

$$[\overline{C}_\Gamma\alpha]_x^{[m,n,c]} := [\vee_{1\le i\le k}(\overline{E}_\Gamma)^i\alpha]_x^{[m,n,c]}.$$

根据定义 11 和定义 12 的公式转换构造命题公式$[M,\varphi]_k := [M^{\varphi,s_0}]_k \wedge [\varphi]_{M_k}$后,公式$\varphi$在$M_k$中的有效性可通过检测命题公式$[M,\varphi]_k$的可满足性来验证,其中,$[\varphi]_{M_k} := [\varphi]_k^{[0,0,0]}$.令$M$是一个模型,$\psi$是一个 AECKL$_n$或 EECKL$_n$公式,则 AECKL$_n$/EECKL$_n$逻辑的有界模型检测算法框架如下:

```
procedure BMC(M,ψ)
if (ψ是一个 AECKL_n 公式) φ=¬ψ else φ=ψ.          //φ是一个 EECKL_n 公式
for k=1 to |M|·|φ|·2^|φ|
    确定 k-model M_k 的命题变量,构造满足迁移关系的命题公式 T(w,v),其中,w,v 是全局状态变量.
    选择 M_k 的一个子模型 M'_k ,其中,|P'_k| ≤ f_k(φ) .
    根据定义 11,将 M'_k 的状态迁移关系转换为一个命题公式 [M^{φ,s0}]_k .
    根据定义 12,在 M'_k 中将公式φ转换为一个命题公式 [φ]_{M_k} .
    构造命题公式 [M,φ]_k := [M^{φ,s0}]_k ∧ [φ]_{M_k} ,并用 SAT 解算器验证[M,φ]_k 的可满足性.
    if ([M,φ]_k 是可满足的)
        if (ψ是一个 AECKL_n 公式) return "M|≠_k ψ" else return "M|=_k ψ".
if (ψ是一个 AECKL_n 公式) return "M|=_k ψ" else return "M|≠_k ψ".
```

6 公式转换的正确性

上一节给出了将EECKL$_n$逻辑的模型检测问题转换为命题可满足性问题的方法,本节将证明这种方法的正确性.这里省略一些与文献[8,9]类似的证明.值得注意的是,由于我们的 EECKL$_n$ 公式转换定义在文献[8]的ECTL*公式转换中扩展了时间域和认知算子,因此,在这些省略的证明中需要考虑时间域.其证明是类似而且冗长的.本节主要对 EECKL$_n$ 公式转换定义的认知部分的正确性进行证明.

引理 **4**. 令M_k是一个k-model,φ是一个 EECKL$_n$ 公式,并且$m,l\le k$和$c\ge 0$,则$M_k,[(\pi_k,l),m,c]|=\varphi$当且仅当存在一个$M_k$的子模型$M'_k(|P'_k| \le f_k(\varphi))$,使得$M'_k,[(\pi_k,l),m,c]|=\varphi$,其中,$P'_k$是$M'_k$的$k$-path 集合.

证明:($\Rightarrow$)施归纳于公式φ的长度来证明.当φ是命题变量或命题变量的否定时,此引理直接成立.假设对于φ的所有真子公式,此引理均成立.如果$\varphi=\alpha U\beta|X\alpha|F\alpha|G\alpha|E\alpha$,即$\varphi$是一个路径公式时,容易在文献[8]的引理 6 的相关证明中扩展时间域,因此证明过程与文献[8]的引理 6 是类似的,这里不再赘述.需要注意的是,当检测一个路径公式在状态$\pi_k(m)$和时间c是否为 true 时,如果$loop(\pi_k)\ne\varnothing$,那么,即使当前考虑的一个状态在$\pi_k$中的位置小于$m$,时间$c$也总是递增的.容易证明:当$\varphi=\alpha\vee\beta|\alpha\wedge\beta$时,此引理成立.下面讨论$\varphi$是认知公式的情况.

1) 令$\varphi=\overline{K}_i\alpha$,并且存在$0\le l\le k$使得$M_k,[(\pi_k,l),m,c]|=\overline{K}_i\alpha$.那么,根据有界同步语义定义 6,存在$\pi'_k\in P_k$使得$\pi'_k(0)=s_0$,并且下列两种情况成立:

- 如果$c\le k$,那么$\pi_k(m)\sim_i\pi'_k(c)$,并且存在$0\le l'\le k$使得$M_k,[(\pi'_k,l'),c,c]|=\alpha$;
- 如果$c>k$,那么存在$0\le l'\le k$,使得$l'\in loop(\pi'_k)$,$\pi_k(m)\sim_i\pi'_k(f(k,l',c))$和$M_k,[(\pi'_k,l'),f(k,l',c),c]|=\alpha$.

因此,根据归纳假设,则存在M_k的子模型$M'_k=((S',s_0,P'_k,\overset{T}{\sim}'_1,\ldots,\overset{T}{\sim}'_n),V')(|P'_k|\le f_k(\alpha))$满足下列条件:若$c\le k$,则存在$0\le l'\le k$使得$M_k,[(\pi'_k,l'),c,c]|=\alpha$;若$c>k$,则存在$l'\in loop(\pi'_k)$使得$M_k,[(\pi'_k,l'),f(k,l',c),c]|=\alpha$.

构造一个M_k的子模型$M''_k=((S'',s_0,P''_k,\overset{T}{\sim}''_1,\ldots,\overset{T}{\sim}''_n),V'')$,其中,$|P''_k|=P'_k\cup\{\pi'_k\}$.这样,根据$M''_k$的构造和有界同步

语义定义 6,存在 $0\le l\le k$ 使得 $M''_k,[(\pi_k,l),m,c]\models\overline{K}_i\alpha$,并且 $|P''_k|=|P'_k|+1\le f_k(\alpha)+1=f_k(\overline{K}_i\alpha)$.

2) 令 $\varphi=\overline{D}_\Gamma\alpha\,|\,\overline{E}_\Gamma\alpha$.证明过程类似于上述 $\varphi=\overline{K}_i\alpha$ 的情况.

3) 令 $\varphi=\overline{C}_\Gamma\alpha$,并且存在 $0\le l\le k$ 使得 $M_k,[(\pi_k,l),m,c]\models\overline{C}_\Gamma\alpha$.那么,根据有界同步语义定义 6,有 $M_k,[(\pi_k,l),m,c]\models\vee_{1\le i\le k}(\overline{E}_\Gamma)^i\alpha$.又根据归纳假设,存在一个 M_k 的子模型 $M^i_k=((S^i,s_0,P^i_k,\overset{T}{\sim}_1^i,\ldots,\overset{T}{\sim}_n^i),V^i)$ (其中,$1\le i\le k$ 并且 $|P^i_k|\le f_k((\overline{E}_\Gamma)^i\alpha)$)使得 $M^i_k,[(\pi_k,l),m,c]\models(\overline{E}_\Gamma)^i\alpha$.

构造一个 M_k 的子模型 $M'_k=((S',s_0,P'_k,\overset{T}{\sim}'_1,\ldots,\overset{T}{\sim}'_n),V')$,如果对于某个 $i\in\{1,\ldots,k\}$ 有 $M^i_k,[(\pi_k,l),m,c]\models(\overline{E}_\Gamma)^i\alpha$,则令 $P'_k=P^i_k$.显然有 $M'_k,[(\pi_k,l),m,c]\models\vee_{1\le i\le k}(\overline{E}_\Gamma)^i\alpha$,其中,$|P'_k|=\max_{1\le i\le k}|P^i_k|$.由定义 10,有 $f_k(\overline{E}_\Gamma\alpha)=f_k(\alpha)+1$,施归纳于公式 $(\overline{E}_\Gamma)^i\alpha$ 的长度可证明 $f_k((\overline{E}_\Gamma)^i\alpha)=f_k(\alpha)+i$.因此,$|P'_k|=\max_{1\le i\le k}|P^i_k|\le\max_{1\le i\le k}f_k((\overline{E}_\Gamma)^i\alpha)=f_k(\alpha)+k=f_k(\overline{C}_\Gamma\alpha)$.

($\Leftarrow$)可直接由子模型以及有界同步语义的定义证明此引理成立. □

引理 5. 令 M_k 是模型 M 的一个 k-model,φ是一个 EECKL$_n$ 公式.对于某个 $l\le k$ 及 M 中任意的 $point(r,c)$,存在一个 M_k 的子模型 $M'_k(|P'_k|\le f_k(\varphi))$,包含一条满足$\pi_k(m)=r(c)$的 k-path π_k(其中 $m\le k$),则 $M'_k,[(\pi_k,l),m,c]\models\varphi$ 当且仅当下列条件成立:

(1) 如果φ是一个状态公式,则 $[M^{\varphi,\pi_k(m)}]_k\wedge[\varphi]^{[0,0,c]}_k$ 是可满足的;

(2) 如果φ是一个路径公式,π_k 不是一条(k,l)-loop,则 $[M^{\varphi,\pi_k(m)}]_k\wedge[\varphi]^{[m,0,c]}_k$ 是可满足的;

(3) 如果φ是一个路径公式,π_k 是一条(k,l)-loop,则 $[M^{\varphi,\pi_k(m)}]_k\wedge[\varphi]^{[m,0,c]}_{k,l}$ 是可满足的.

证明:($\Rightarrow$)假设φ是一个路径公式.容易在文献[8]的引理 7 中扩展时间域后证明条件(1)和条件(2)成立,这里不再赘述.我们主要讨论φ是包含认知算子的状态公式的情况.假设φ是一个状态公式.令 $L_{k,i}:=\vee_{0\le l'\le k}L_{k,i}(l')$,其中,$0\le i\le k$.施归纳于公式$\varphi$的长度来证明条件(3)成立.当$\varphi$是一个命题变量或其否定时,条件(3)直接成立.假设对于φ的所有真子公式条件(3)均成立.当$\varphi=\alpha\vee\beta|\alpha\wedge\beta$时也容易证明条件(3)成立.考虑下列情况:

1) 令$\varphi=E\alpha$,存在一个 M_k 的子模型 $M'_k(|P'_k|\le f_k(E\alpha))$,并且存在 $m,l\le k$,使得 $M'_k,[(\pi_k,l),m,c]\models E\alpha$,其中,$\pi_k(m)=r(c)$.那么由定义 6,有 $\exists\pi'_k\in P_k(\pi'_k(0)=\pi_k(m)$,并且 $\exists_{0\le l'\le k}[(\pi'_k,l'),0,c]\models\alpha)$.由于α是一个路径公式,由归纳假设及条件(1)和条件(2),有公式 $E_1:=((\neg L_{k,0}\wedge[\alpha]^{[0,0,c]}_k)\vee\vee^k_{l'=0}(L_{k,0}(l')\wedge[\alpha]^{[0,0,c]}_{k,l'}))\wedge[M^{\alpha,\pi'_k(0)}]_k$ 是可满足的.令 i 是使 $H(w_{0,0},w_{0,i})$为 true 的新的符号 k-path 的编号.设 M'_k和M''_k 是 M_k 的两个子模型,其中 $P'_k\subseteq P''_k$,那么,如果 $M'\models_k\varphi$,则 $M''\models_k\varphi$.根据上述命题公式的构造,有公式 $E_2:=\vee^{f_k(\varphi)}_{i=1}(H(w_{0,0},w_{0,i})\wedge(\neg L_{k,i}\wedge[\alpha]^{[0,i,c]}_k)\vee\vee^k_{l'=0}(L_{k,i}(l')\wedge[\alpha]^{[0,i,c]}_{k,l'}))\wedge[M^{E\alpha,\pi'_k(0)}]_k$ 是可满足的.这样,由 EECKL$_n$ 公式转换定义 12 可知,$[E\alpha]^{[0,0,c]}_k\wedge[M^{E\alpha,\pi_k(m)}]_k$ 是可满足的,因为 E_2 与 $[E\alpha]^{[0,0,c]}_k\wedge[M^{E\alpha,\pi_k(m)}]_k$ 等价.

2) 令 $\varphi=\overline{K}_a\alpha$ $(a\in A)$,存在一个 M_k 的子模型 $M'_k(|P'_k|\le f_k(\overline{K}_a\alpha))$,并且存在 $m,l\le k$,使得 $M'_k,[(\pi_k,l),m,c]\models\overline{K}_a\alpha$,其中$\pi_k(m)=r(c)$.那么,由有界同步语义定义 6,有 $\exists\pi'_k\in P'_k$ 使得 $\pi'_k(0)=s_0$,并且若 $c\le k$,则 $\pi_k(m)\sim_a\pi'_k(c)$,并且 $\exists_{0\le l'\le k}M'_k,[(\pi'_k,l'),c,c]\models\alpha$;若 $c>k$,则 $\exists_{0\le l'\le k}(l'\in loop(\pi'_k)$,并且 $\pi_k(m)\sim_a\pi'_k(f(k,l',c))$, $M'_k,[(\pi'_k,l'),f(k,l',c),c]\models\alpha)$.令 i 是使 $I_{s_0}(w_{0,i})$ 为 true 的新的符号 k-path π'_k 的编号,考虑以下两种情况:

- 令 $c\le k$.因为 $M'_k,[(\pi'_k,l'),c,c]\models\alpha$ (α是一个路径公式),由归纳假设及条件(1)和条件(2),我们有:如果 π'_k 是一条(k,l')-loop,则公式 $[\alpha]^{[c,0,c]}_{k,l'}\wedge[M^{\alpha,\pi'_k(c)}]_k$ 是可满足的;否则,公式 $[\alpha]^{[c,0,c]}_k\wedge[M^{\alpha,\pi'_k(c)}]_k$ 是可满足的.因此,由上述公式的构造我们有:若 $loop(\pi'_k)=\varnothing$,则公式 $F_1:=\neg L_{k,i}\wedge H_a(w_{0,0},w_{c,i})\wedge[\alpha]^{[c,i,c]}_k\wedge[M^{\overline{K}_a\alpha,\pi_k(m)}]_k$ 可满足;若 $l'\in loop(\pi'_k)$,则公式 $F_2:=L_{k,i}(l')\wedge H_a(w_{0,0},w_{c,i})\wedge[\alpha]^{[c,i,c]}_{k,l'}\wedge[M^{\overline{K}_a\alpha,\pi_k(m)}]_k$ 可满足.这样,当 $c\le k$ 时,公式 $F_3:=I_{s_0}(w_{0,i})\wedge(F_1\vee\vee^k_{l'=0}F_2)$ 是可满足的.
- 令 $c>k$.如果 $loop(\pi'_k)=\varnothing$,显然,M的状态 $r(c)$不能映射到 k-path π'_k 的任何一个状态上,因此, π'_k 必须是一

条 k-loop. 假设 $l'\in loop(\pi'_k)$，由于 $M'_k,[(\pi'_k,l'),f(k,l',c),c]\models\alpha$，根据归纳假设及条件(2),有公式 $[\alpha]_{k,l'}^{[f(k,l',c),o,c]}\wedge[M^{\alpha,\pi'_k(f(k,l',c))}]_k$ 是可满足的.这样,根据上述公式的构造,有公式 $F_4:=L_{k,i}(l')\wedge H_a(w_{0,0},w_{f(k,l',c),i})\wedge[\alpha]_{k,l'}^{[f(k,l',c),i,c]}\wedge[M^{\overline{K}_a\alpha,\pi_k(m)}]_k$ 是可满足的.

因此,由上述结论我们有:如果 $c\le k$,则公式 $\vee_{i=1}^{f_k(\varphi)}F_3$ 是可满足的;否则,公式 $\vee_{i=1}^{f_k(\varphi)}(I_{s_0}(w_{0,i})\wedge\vee_{l'=0}^{k}F_4)$ 是可满足的.而且,由公式转换定义 12 可知,上述条件等价于公式 $[\overline{K}_a\alpha]_k^{[0,0,c]}\wedge[M^{\overline{K}_a\alpha,\pi_k(m)}]_k$ 也是可满足的.

3) 令 $\varphi=\overline{D}_\Gamma\alpha\,|\,\overline{E}_\Gamma\alpha$.证明过程类似于 $\varphi=\overline{K}_a\alpha$ 的情况.

4) 令 $\varphi=\overline{C}_\Gamma\alpha$.由于 $\overline{C}_\Gamma\alpha:=\vee_{1\le i\le k}(\overline{E}_\Gamma)^i\alpha$,可由 $\varphi=\alpha\vee\beta$ 和 $\varphi=\overline{E}_\Gamma\alpha$ 的情况归纳证明.

($\Leftarrow$)容易由状态迁移关系公式转换定义 11 和公式转换定义 12 证明条件(1)~条件(3)成立. □

定理 2. 令 M 是一个模型,φ 是一个 EECKL$_n$ 公式,那么 $M\models_k\varphi$ 当且仅当 $[M^{\varphi,s_0}]_k\wedge[\varphi]_{M_k}$ 是可满足的.

证明:直接由引理 4 和引理 5 证明此定理成立. □

7 实例:火车控制系统

本节通过文献[4]中的火车控制系统实例来解释算法的执行过程.该实例描述如下:

火车控制系统由两列火车和一个控制器组成.顺时针和逆时针行驶的火车各占一条环形轨道.两列火车必须通过一条只能容纳一列火车的隧道.隧道两头各有一盏交通灯(颜色非红即绿).两列火车各装有一个信号发生器,用来在火车接近隧道时向控制器发送信号.控制器可以收到两列火车的信号并控制隧道两头的交通灯.控制器的任务是确保在同一时刻两列火车永远不会同时进入隧道.这里假定这两列火车始终忠实地遵守遇到红灯立即停车的交通规则.图 1 和图 2 给出其系统示意图和各智能体的局部状态迁移图.

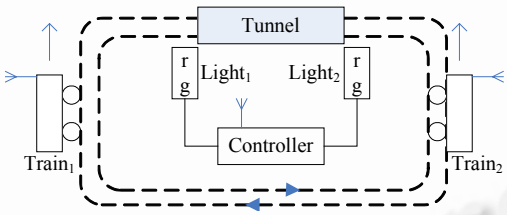

Fig.1 Train controller system (TCS)

图 1 火车控制系统

图 火车控制系统

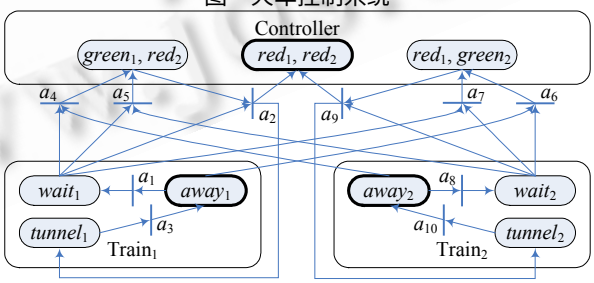

Fig.2 The local transition structure for TCS

图 2 局部状态迁移图

下面把这个实例转化为一个同步时态认知模型.令智能体集合 $A=\{1,2,3\}$,1,2,3 依次代表图 1 和图 2 中的 Train$_1$,Controller 和 Train$_2$.这里忽略环境.圆角矩形表示各智能体的局部状态,其中边框加粗的是初始局部状态.由于 Controller 有两盏信号灯,因此,它的局部状态是这两个信号灯颜色的组合.各智能体的局部状态集合是 $L_1=\{away_1,wait_1,tunnel_1\}$,$L_2=\{(red_1,red_2),(red_1,green_2),(green_1,red_2),(green_1,green_2)$,$L_3=\{away_2,wait_2,tunnel_2\}$.系统

的全局状态空间 $G=L_1\times L_2\times L_3$,初始状态 $s_0=(away_1,red_1,red_2,away_2)$.联合动作集合 $Act=\{a_1,\ldots,a_{10}\}$.对于一个动作 $a\in Act$,定义其前置条件 $pre(a)$和后置条件 $post(a)$均为一个局部状态集合,它们分别表示动作 a 执行前的先决条件和执行后的状态.另外,定义 $agent(a)$为执行动作 a 时可能会改变局部状态的智能体集合.例如,$pre(a_2)=\{wait_1,(green_1,red_2)\}$,$post(a_2)=\{tunnel_1,(red_1,red_2)\}$,$agent(a_2)=\{1,2\}$.

假设有一个 EECKL$_n$ 规范 $\varphi=EF\overline{K}_1FG\neg in_2$,命题变量 in_2 表示 Train$_2$ 正处于隧道中.令 $s\in S$ 对应的状态集合,则 $V(s)(in_2)$=true 当且仅当 $l_3(s)=tunnel_2$,其中,函数 $l_i:S'\to L_i$ 从一个全局状态 s 中获得智能体 i 的局部状态 s_i.然后,根据第 6 节的有界模型检测算法框架,令路径长度 k 从 1 至$|M|\cdot|\varphi|\cdot 2^{|\varphi|}$依次测试命题公式$[M,\varphi]_k$的可满足性.下面描述 k=2 时公式$[M,\varphi]_k$的构造过程.

7.1 命题公式$[M^{\varphi,s_0}]_2$的构造

首先对局部状态进行二进制编码.由公式转换的过程可知,L_1,L_3 分别需要 $n_1=n_3=2$ 个二进制位表示 3 个局部状态:$(0,0)=away_1=away_2$,$(0,1)=wait_1=wait_2$,$(1,0)=tunnel_1=tunnel_2$.L_2 需要 $n_2=2$ 个二进制位表示 4 个局部状态:$(0,0)=(red_1,red_2)$,$(0,1)=(red_1,green_2)$,$(1,0)=(green_1,red_2)$,$(1,1)=(green_1,green_2)$.因此,一个全局状态变量需要 $m=n_1+n_2+n_3=6$ 个二进制位来表示.智能体的局部命题变量索引集合分别是 $I_1=\{0,1\}$,$I_2=\{2,3\}$和 $I_3=\{4,5\}$.然后,令 $w=(w[0],\ldots,w[5])$和 $v=(v[0],\ldots,v[5])$是两个全局状态变量,构造下列命题公式:

- 对于每个智能体的每个局部状态,根据其二进制编码构造相应的公式,如 $p_{tunnel_2}(w):=w[4]\wedge\neg w[5]$.
- 系统的初始状态由命题公式 $I_{s_0}(w):=\wedge_{i\in\{0,\ldots,5\}}\neg w[i]$ 表示.
- $T(w,v):=\vee_{a\in Act}(\wedge_{i\in pre(a)}p_i(w)\wedge\wedge_{i\in post(a)}p_i(v)\wedge\wedge_{i\in B}(w[i]\Leftrightarrow v[i]))\vee(\wedge_{a\in Act}\vee_{i\in pre(a)}(\neg p_i(w))\wedge\wedge_{i\in\{0,\ldots,5\}}(w[i]\Leftrightarrow v[i]))$.其中,$B:=\cup_{i\in(A-agent(a))}I_i$是动作 a 不能改变的局部状态位的索引集合.令 s,s'分别是由全局状态变量 w 和 v 表示的状态,则 $T(w,v)$表示从 s 到 s'是可达的.$T(w,v)$由两个析取式组成:第 1 个表示在状态 s 存在一组可执行的联合动作,这些动作执行后状态变成 s',并且 s'中联合动作不能修改的局部状态位均保持不变;第 2 个表示如果在状态 s 不存在任何可执行的联合动作,则 $s=s'$,这保证了状态迁移关系 T 是完全的(total).这样,$T(w,v)$为 true 当且仅当$(s,s')\in T$.由定义 10 计算 $f_2(\varphi)=2$.根据定义 11 构造状态迁移关系命题公式:$[M^{\varphi,s_0}]_2:=I_{s_0}(w_{0,0})\wedge\wedge_{1\le j\le 2}\wedge_{0\le i<2}T(w_{i,j},w_{i+1,j})$.

7.2 命题公式$[\varphi]_{M_2}$的构造

令 $w=(w[0],\ldots,w[5])$和 $v=(v[0],\ldots,v[5])$是两个全局状态变量,定义下列公式:

- $in_2(w):=p_{tunnel_2}(w):=w[4]\wedge\neg w[5]$;
- $H(w,v):=\wedge_{i\in\{0,\ldots,5\}}(w[i]\Leftrightarrow v[i])$表示 w 和 v 是逻辑等价的;
- $H_i(w,v):=\wedge_{j\in I_i}(w[i]\Leftrightarrow v[i])$ 表示 w,v 中的 i-local 状态(智能体 i 的局部状态)是逻辑等价的,其中 $i\in A$;
- 令 $L_{k,i}:=\vee_{0\le l\le k}L_{k,i}(l)$,$L'_{k,i}:=\vee_{0\le l'\le k}L_{k,i}(l')$,其中 $0\le i\le k$.

因为$[\varphi]_{M_2}:=[\varphi]_2^{[0,0,0]}$,下面根据 EECKL$_n$ 公式转换定义 12 构造公式$[\varphi]_2^{[0,0,0]}$.为节省篇幅,我们省略非 k-loop 语义下的公式转换过程:

- $[\varphi]_2^{[0,0,0]}:=\vee_{i=1}^2(H(w_{0,0},w_{0,i})\wedge((\neg L'_{2,i}\wedge[F\overline{K}_1FG\neg in_2]_2^{[0,i,0]})\vee\vee_{l'=0}^2(L_{2,i}(l')\wedge[F\overline{K}_1FG\neg in_2]_{2,l'}^{[0,i,0]})))$;
- $[F\overline{K}_1FG\neg in_2]_{2,l'}^{[0,i,0]}:=\vee_{j=0}^2[\overline{K}_1FG\neg in_2]_{2,l'}^{[j,i,j]}$;
- $[\overline{K}_1FG\neg in_2]_{2,l'}^{[j,i,j]}:=\vee_{r=1}^2(I_{s_0}(w_{0,r})\wedge H_1(w_{j,i},w_{j,r})\wedge((\neg L_{2,r}\wedge[FG\neg in_2]_2^{[j,r,j]})\vee\vee_{l=0}^2(L_{2,r}(l)\wedge[FG\neg in_2]_{2,l}^{[j,r,j]})))$;
- $[FG\neg in_2]_{2,l}^{[j,r,j]}:=\vee_{i=j}^2[G\neg in_2]_{2,l}^{[i,r,i]}\vee\vee_{i=l}^{j-1}[G\neg in_2]_{2,l}^{[i,r,3+i-l]}$;
- 如果 $l\ge i$,则 $[G\neg in_2]_{2,l}^{[i,r,3+i-l]}$ 等于 $\wedge_{j=i}^2[\neg in_2]_{2,l}^{[j,r,3+j-l]}$;否则,等于 $\wedge_{j=i}^2[\neg in_2]_{2,l}^{[j,r,3+j-l]}\wedge\wedge_{j=l}^{i-1}[\neg in_2]_{2,l}^{[j,r,6+j-2l]}$;
- $[G\neg in_2]_{2,l}^{[i,r,i]}$的构造与$[G\neg in_2]_{2,l}^{[i,r,3+i-l]}$的类似;

  $[\neg in_2]_{2,l}^{[j,r,3+i-l]}:=[\neg in_2]_{2,l}^{[j,r,6+j-2l]}:=\neg in_2(w_{j,r}):=\neg w_{j,r}[4]\vee w_{j,r}[5]$.

最后,依据第 5 节的算法框架进行有界模型检测.由于可达全局状态总数$|S|$=9(controller 的局部状态不包括

($green_1$,$green_2$)),迁移关系个数$|T|=10$,因此,令 k 从 1 到$(|S|+|T|)\cdot|\varphi|\cdot 2^{|\varphi|}=19\times5\times2^5$依次构造命题公式$[M,\varphi]_k$.每次构造的过程类似于上述 $k=2$ 的情况.然后,用 ZChaff 验证$[M,\varphi]_k$ 的可满足性.结果是$[M,\varphi]_1$ 和$[M,\varphi]_2$ 不可满足而$[M,\varphi]_3$是可满足的,算法结束.因此,$M\models_k\varphi$成立.从直觉上看,系统存在一条计算路径使得动作 a_6 和 a_7 永远得不到执行.这样,Controller 的第 2 盏交通灯不可能变成 $green_2$,因此,Train_2 永远进不了隧道.

可以看出,规范φ中的认知算子约束的子公式是路径公式,因此不能用文献[9]中的 CTLK 逻辑加以描述.另外,我们也可以验证更复杂的 $\text{AECKL}_n/\text{EECKL}_n$ 规范.比如,认知算子约束的子公式同时包含路径公式和路径量词约束的状态子公式.显然,Halpern 的 CKL_n 逻辑也不能描述这类公式.

8 结束语

本文系统地论述了在同步的多智体系统中有界模型检测 $\text{AECKL}_n/\text{EECKL}_n$ 逻辑的方法.通过修改文献[9]的认知可达关系为同步认知可达关系,我们得到多智体同步解释系统.另一方面,为了提高时态表达能力,我们在文献[8]的 ECTL*逻辑中加入认知算子获得 EECKL_n 逻辑及其同步语义和有界同步语义,并给出 $\text{AECKL}_n/\text{EECKL}_n$ 逻辑的有界模型检测算法.该方法的优势在本文开始部分有叙述.另外,本文给出该方法的正确性证明,并通过火车控制系统实例解释算法的执行过程.

在工具开发方面,由于我们已根据文献[6]的理论在 NuSMV 2.1.2 开放源码的基础上采用 OBDD 技术实现了 ECKL_n 逻辑的模型检测工具 MCTK,而 MCTK 的输入语言经适当扩展即可适用于同步多智体系统及 $\text{AECKL}_n/\text{EECKL}_n$ 规范的描述,因此,拟在 MCTK 的有界模型检测模块中扩展本文的算法.另外,今后可在以下几方面完善和提高本算法的功能及效率:

1) 分析公式转换定义在逻辑上是否存在冗余,并在保持原有有界语义基础上化简、减小命题公式规模;
2) 将本文的有界模型检测问题转换为约束满足问题(constraint satisfaction problem)进行求解,使得转换更为自然、效率更高;
3) 智能体的知识扩展到具有完美记忆[2]的情况,使得智能体具有更强的信息获取能力;
4) 将该算法扩展到无界模型检测(unbounded model checking),使得我们可以验证完整的 ECKL_n 规范.

安全协议在不可靠不安全网络的通信中起着重要作用,它们主要负责建立起通信主体间的信任,并能够进行通信主体间的加密密钥的分配.一个安全协议的规范可能会考虑一个主体在协议运行期间或结束时拥有的知识和信念包含另一个主体的知识或信念,而协议中的信息交换在哲学领域被认为是动态认识论的问题.因此,我们的 $\text{AECKL}_n/\text{EECKL}_n$ 时态认知逻辑可以很自然地表示这些规范.另外,所有主体或入侵者显然知道他们在协议中执行到第几步,因此,我们的同步多智体系统有界模型检测算法也适合安全协议的建模,并对其安全规范进行自动的形式化验证.这将是本文算法的一个重要应用方向.

References:

[1] Clarke EM, Grumberg O, Peled DA. Model Checking. Cambridge: MIT Press, 2000.

[2] Fagin R, Halpern J, Moses Y, Vardi M. Reasoning About Knowledge. Cambridge: MIT Press, 1995.

[3] van der Hoek W, Wooldridge M. Model checking knowledge and time. In: Bosnacki D, Leue S, eds. Proc. of the 9th Int'l SPIN Workshop on Model Checking of Software. Berlin: Springer-Verlag, 2002. 95−111.

[4] van der Hoek W, Wooldridge M. Tractable multiagent planning for epistemic goals. In: Castelfranchi C, Johnson W, eds. Proc. of the 1st Int'l Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2002). New York: ACM Press, 2002. 1167−1174.

[5] Wu LJ, Su KL. A model checking algorithm for temporal logics of knowledge in multi-Agent systems. Journal of Software, 2004, 15(7):1012−1020 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/1012.htm

[6] Su KL. Model checking temporal logics of knowledge in distributed systems. In: McGuinness DL, Ferguson G, eds. Proc. of the 19th National Conf. on Artificial Intelligence, the 16th Conf. on Innovative Applications of Artificial Intelligence. Menlo Park: AAAI Press/MIT Press, 2004. 98−103.

[7] Biere A, Cimatti A, Clarke EM, Zhu Y. Symbolic model checking without BDDs. In: Cleaveland WR, ed. Proc. of the 5th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems. Berlin: Springer-Verlag, 1999. 193−207.

[8] Woźna B. Bounded model checking for the universal fragment of CTL*. Technical Report, TR-04-03, London: King's College London, 2004.

[9] Penczek W, Lomuscio A. Verifying epistemic properties of multi-Agent systems via bounded model checking. Fundamenta Informaticae, 2003,55(2):167−185.

[10] Luo XY, Su KL, Sattar A, Chen QL, Lv GF. Bounded model checking knowledge and branching time in synchronous multi-Agent systems. In: Dignum F, Dignum V, Koenig S, Kraus S, Singh MP, Wooldridge M, eds. Proc. of the 4th Int'l Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2005), New York: ACM Press, 2005. 1129−1130.

[11] Gastin P, Oddoux D. Fast LTL to Büchi automata translation. In: Berry G, Comon H, Finkel A, eds. Proc. of the 13th Conf. on Computer Aided Verification (CAV 2001). Berlin: Springer-Verlag, 2001. 53−65.

附中文参考文献:

[5] 吴立军,苏开乐.多智体系统时态认知规范的模型检测算法.软件学报,2004,15(7):1012−1020. http://www.jos.org.cn/1000-9825/15/1012.htm

骆翔宇(1974－),男,广西桂林人,博士,讲师,主要研究领域为模型检测,模态逻辑,时态逻辑,知识推理,多智体系统,安全协议验证.

苏开乐(1964－),男,教授,博士生导师,主要研究领域为模型检测,知识推理,非单调推理,多智体系统,模态逻辑,时态逻辑,概率推理,安全协议验证.

杨晋吉(1968－),男,博士生,副教授,主要研究领域为安全协议验证,人工智能逻辑,多媒体系统.