一种求解认知难题的模型检测方法

骆翔宇[1),2)] 苏开乐[3)] 顾 明[1)]

[1)](清华大学软件学院信息系统安全教育部重点实验室 北京 100084)

[2)](桂林电子科技大学计算机与控制学院 广西 桂林 541004)

[3)](北京大学信息科学技术学院 北京 100871)

摘 要 用公告逻辑建模并求解和与积认知难题.提出一种动态认知模型,将环境认知模型与公告导致的认知模型线性组合,从而在时态认知逻辑模型检测技术中扩展支持公告逻辑的建模与验证.该模型检测方法不仅可以用于搜索认知难题的所有解,而且可以验证相关的时态认知性质,这一特性是当前认知逻辑模型检测工具 MCK、MCMAS 和 DEMO 不能完全支持的.作者采用 OBDD 开发了相关的符号化模型检测工具 MCTK 并对和与积难题进行建模和验证,实验结果说明了文中方法的正确性和高效性.

关键词 模型检测;OBDD;公告逻辑;时态认知逻辑;和与积难题

中图法分类号 TP301 **DOI 号**: 10.3724/SP.J.1016.2009.00406

A Model Checking Approach for Solving Epistemic Riddles

LUO Xiang-Yu[1),2)] SU Kai-Le[3)] GU Ming[1)]

[1)](*Key Laboratory of Security for Information System of Ministry of Education, School of Software, Tsinghua University, Beijing* 100084)

[2)](*School of Computer & Control, Guilin University of Electronic Technology, Guilin, Guangxi* 541004)

[3)](*School of Electronics Engineering and Computer Science, Peking University, Beijing* 100871)

Abstract This paper aims to model and solve the Sum and Product Riddle in public announcement logic. A dynamic epistemic model is proposed, that is the linear temporal combination of the epistemic model of environment and the epistemic models after each announcement, such that the authors' model checking technique for temporal epistemic logic can be extended to support the modeling and verification of public announcement logic. This model checking method not only can help to find all solutions, but also verify related temporal epistemic properties. Such characteristic is not fully supported by the current version of MCK, MCMAS and DEMO. The authors implemented the proposed method in the symbolic model checker MCTK via OBDD and verified the sum and product riddle. The experimental results show that the proposed method is correct and efficient.

Keywords model checking; ordered binary decision diagram; public announcement logic; temporal epistemic logics; sum and product riddle

收稿日期:2009-04-20;最终修改稿收到日期:2009-07-22.本课题得到国家自然科学基金重点项目(90718039)、国家"九七三"重点基础研究发展规划项目基金(2010CB328103)、国家杰出青年科学基金(60725207)、国家自然科学基金(60763004)、广西青年科学基金(桂科青0728090)和中国博士后科学基金(20090450389)资助.**骆翔宇**,男,1974 年生,博士,副教授,主要研究方向包括模型检测、时态逻辑、认知逻辑、知识推理、多主体系统、安全协议验证等. E-mail: shiangyuluo@gmail.com. **苏开乐**,男,1964 年生,博士,教授,博士生导师,主要研究领域包括模型检测、知识推理、非单调推理、多主体系统、模态逻辑、时态逻辑、概率推理、安全协议验证、逻辑程序设计等. **顾 明**,女,1962 年生,教授,博士生导师,主要研究领域包括形式化方法、信息安全、软件中间件技术等.

1 引　言

认知难题(epistemic riddles)的一个共性是在一个多主体系统(multi-agent system)中经过一系列关于主体的知识或无知的公告后,主体将获知环境中的事实.在任一系统全局状态中,每一主体只能访问自身的局部状态(即全局状态的一部分).如果系统状态域是所有主体局部状态集合的全笛卡尔积,则所有主体的一个公共知识是任一主体均不能获知其他主体局部状态的相关信息,这时主体公告他的无知对于其他主体来说是没有价值的;否则主体的无知公告对其他主体来说是有价值的.认知难题的例子有和与积问题[1]、泥巴小孩问题[2]、谁是和问题[3]以及俄罗斯牌[4]信息安全协议等.本文以经典的和与积问题为案例研究认知难题的自动求解算法.1969 年,Freudenthal 首次在文献[5]中用 Dutch 语言解释了和与积问题,接着在 1970 年求得该问题的解[6].我们引用文献[1]中对该问题的描述如下:

主体 J 向 S 和 P 说道:我有两个不同的整数 x 和 y,它们满足 $1<x<y$ 和 $x+y\leq 100$.我将秘密地把和 $s=x+y$ 告诉 S,而把积 $p=xy$ 告诉 P.请你们确定这两个数(x,y)是什么.J 将和与积分别秘密告知 S 和 P 后,发生如下对话:

(1)P 说:"我不知道这两个数";

(2)S 说:"我早已知道你不知道这两个数";

(3)P 说:"我现在知道这两个数了";

(4)S 说:"我现在也知道这两个数了".

请问这两个数(x,y)是什么?

由于主体只是公告了他们关于这两个数的知识或无知,并没有这两个数的具体描述,因此初看起来这些公告对于主体推测这两个具体的数是没有价值的.然而事实并非如此,主体可通过其他主体的公告逐步推断出这两个数的实际值.例如,这两个数不是 2 和 4,因为只有 2 和 4 的积是 8,这样 P 可推断出这两个数是 2 和 4.这两个数也不是素数,因为两个素数的积是唯一的,因此 P 也可根据积推断出这两个素数.此外,这两个数也不是 14 和 16,因为素数 7 和 23 的和同样是 30,而素数 7 和 23 可被 P 推断出,因此如果和是 30,那么 S 应该认为 P 可能知道这两个数.然而 S 却说他知道 P 不知道这两个数.因此这两个数不可能是 14 和 16.通过此类推断,S 和 P 可逐步排除那些不可能的数对,最终得到该问题的唯一解是(4,13).

和与积问题可建模为一个多主体系统(MAS).多主体系统领域关注于主体的知识、信念、愿望和意图等精神状态的形式表示与推理.而动态认知逻辑(Dynamic Epistemic Logic)则用于研究主体个体知识或群体知识通信所导致的主体知识的变化.公告逻辑(Public Announcement Logic)也是一种动态认知逻辑,由于其可直接描述主体或主体群体的公告,因此非常适用于和与积这类问题的建模.van Ditmarsch 等在文献[7]中宣称他们首次用动态认知模型检测工具 DEMO[4] 解决了该问题,但是他们也说明了时态知识逻辑[2,8] 模型检测工具 MCK 0.2.0[9] 和 MCMAS 0.7[10] 不能解决该问题,因为他们的规格描述语言的条件语句中不支持知识逻辑公式的描述.本文提出一种动态认知模型,使得我们可以在我们的时态认知逻辑模型检测工具 MCTK[11-12] 基础上扩展支持公告逻辑的建模和验证.我们在 MCTK 中实现了该模型检测方法并解决了和与积问题,实验结果表明了我们的方法不仅正确,而且与 van Ditmarsch 等在文献[7]中基于 DEMO 的方法相比,我们的方法在效率上占有绝对优势.从另一个角度来说,我们的方法结合了传统时态认知逻辑模型检测技术与公告逻辑建模与验证方法,这一特性是 MCK、MCMAS 和 DEMO 等工具不能完全支持的.

本文第 2 节介绍公告逻辑;第 3 节介绍基于公告逻辑的和与积问题建模方法;第 4 节介绍基于 MCTK 的认知难题建模与验证方法,包括动态认知模型及其符号化描述语言;多主体有限状态程序;第 5 节介绍基于 MCTK 的和与积问题建模方法;第 6 节给出实验结果;最后在第 7 节总结并讨论下一步研究工作.

2 公告逻辑

给定一个主体集合 $Ag=\{1,\cdots,n\}$ 和原子命题集合 P,公告逻辑的语法归纳定义如下:

$$\varphi ::= p\mid\neg\varphi\mid\varphi\wedge\psi\mid K_a\varphi\mid C_\Gamma\varphi\mid[\varphi]\psi,$$

其中 $p\in P$,$a\in Ag$ 并且 $\Gamma\subseteq Ag$.$K_a\varphi$ 读作"主体 a 知道公式 φ 成立".$C_\Gamma\varphi$ 读作"公式 φ 成立是主体群体 Γ 的公共知识".而$[\varphi]\psi$ 读作"公式 ψ 将在公告 φ 之后成立".

现在,我们可以在认知模型

$$M=\langle S,V,\sim_1,\cdots,\sim_n\rangle$$

上定义公告逻辑的语义.令 L_e 是环境中的状态集

合，L_i 是每一主体 $i\in Ag$ 的局部状态集合，我们定义 $S=L_e\times L_1\times\cdots\times L_n$ 为(全局)状态集合. $V:P\to\mathcal{P}(S)$是一个评价函数. 对于任意 $p\in P$，V 给出使得 p 为真的所有状态的集合. 我们为每一主体 $i\in Ag$ 赋予一个状态集合上的认知等价关系 $\sim_i$. 定义函数 $l_i:S\to L_i$，用于从一个全局状态 $s\in S$ 中获取主体 i 的局部状态. 这样，对于任意 $s,s'\in S$，$s\sim_i s'$ 当且仅当 $l_i(s)=l_i(s')$. 这意味着对于主体 i 来说，s 和 s' 是不可区分的. 公告逻辑的语义归纳定义如下：

(1) $M,s\models p$ iff $s\in V(p)$，其中 $p\in P$；

(2) $M,s\models\neg\varphi$ iff $M,s\not\models\varphi$；

(3) $M,s\models\varphi\wedge\psi$ iff $M,s\models\varphi$ 并且 $M,s\models\psi$；

(4) $M,s\models K_i\varphi$ iff 对于任一满足 $s\sim_i s'$ 的状态 $s'\in S$，$M,s'\models\varphi$；

(5) $M,s\models C_\Gamma\varphi$ iff 对于任一满足 $s\sim_\Gamma^C s'$ 的状态 $s'\in S$，$M,s'\models\varphi$，其中 $\sim_\Gamma^C$ 是 $\bigcup_{i\in\Gamma}\sim_i$ 的传递闭包；

(6) $M,s\models[\varphi]\psi$ iff $M,s\models\varphi$ 蕴含 $M|_\varphi,s\models\psi$. $M|_\varphi=\langle S',V',\sim'_1,\cdots,\sim'_n\rangle$ 也是一个认知模型，其中 $S'=\{s\in S\mid M,s\models\varphi\}$；对任一 $p\in P$，$V'(p)=V(p)\cap S'$；对任一 $i\in Ag$，$\sim'_i=\sim_i\cap(S'\times S')$.

给定一个认知模型 M 和一个公式 φ，φ 在 M 有效(记为 $M\models\varphi$)当且仅当对于任一 M 中的状态 s，$M,s\models\varphi$.

3 基于公告逻辑的和与积问题建模

本节采用公告逻辑对和与积问题进行建模. 首先确定该问题涉及的原子命题集合和主体集合. 从引言可知，整数对 (x,y) 包含于 $I=\{(x,y)\in N^2\mid 1<x<y$ 并且 $x+y\leq 100\}$. 因此，需要用 14 个命题变量 $\{x_6,x_5,\cdots,x_0\}$ 和 $\{y_6,y_5,\cdots,y_0\}$ 对 x 和 y 进行二进制编码. 我们分别用符号 E_x^i 和 E_y^j 表示命题“$x=i$”和“$y=j$”. 例如，E_x^{13} 表示命题逻辑公式 $\neg x_6\wedge\neg x_5\wedge\neg x_4\wedge x_3\wedge x_2\wedge\neg x_1\wedge x_0$. 由于文献[1]中至少需要 194 个命题符号表示 x 和 y 的每一取值，因此与文献[1]中的编码方法相比，我们的方法更加紧凑.

现在考虑主体集合. 由于主体 J 用于确保其他主体获得该问题相关的背景知识，其扮演的角色类似于多主体系统中的环境，因此在建模中不需考虑主体 J 的知识. 这样，主体集合就是 $Ag=\{\mathrm{S},\mathrm{P}\}$.

命题“S 知道 (x,y) 是 $(4,13)$”可表示为公告逻辑公式 $K_\mathrm{S}(E_x^4\wedge E_y^{13})$. 现在考虑如何用公告逻辑公式表示命题“S 知道 (x,y)”. 首先，对于主体 S，我们可分别用 $K_\mathrm{S}x=\bigwedge_{0\leq i\leq 6}(K_\mathrm{S}x_i\vee K_\mathrm{S}\neg x_i)$ 和 $K_\mathrm{S}y=\bigwedge_{0\leq i\leq 6}(K_\mathrm{S}y_i\vee K_\mathrm{S}\neg y_i)$ 表示命题“S 知道 x”和“S 知道 y”. 例如，公式 $K_\mathrm{S}x$ 意味着 S 知道 x 的所有二进制编码变量的取值. 因此，命题“S 知道 (x,y)”可表示为 $K_\mathrm{S}(x,y)=K_\mathrm{S}x\wedge K_\mathrm{S}y$. 类似地，命题“P 知道 (x,y)”可表示为 $K_\mathrm{P}(x,y)=K_\mathrm{P}x\wedge K_\mathrm{P}y$，其中 $K_\mathrm{P}x$，$K_\mathrm{P}y$ 的结构类似于 $K_\mathrm{S}x$，$K_\mathrm{S}y$. 这样，该问题的 4 个公告可依次表示为 $\neg K_\mathrm{P}(x,y)$，$K_\mathrm{S}\neg K_\mathrm{P}(x,y)$，$K_\mathrm{P}(x,y)$ 和 $K_\mathrm{S}(x,y)$.

现在，我们可以在该问题的认知模型

$$\Theta=\langle I,V,\sim_\mathrm{S},\sim_\mathrm{P}\rangle$$

上解释这些公告了，其中 $I=\{(x,y)\in N^2\mid 1<x<y$ 并且 $x+y\leq 100\}$. 对于任意的整数 $x,y,x',y'\in I$，$(x,y)\sim_\mathrm{S}(x',y')$ 当且仅当 $x+y=x'+y'$；$(x,y)\sim_\mathrm{P}x',y'$ 当且仅当 $xy=x'y'$. 评价函数 V 定义为：对任一 $i(=0,\cdots,6)$，$V(x_i)=\{(x,y)\in I\mid((x/2^i)\bmod 2)=1\}$；$V(y_i)=\{(x,y)\in I\mid((y/2^i)\bmod 2)=1\}$，其中“/”是整数求余运算符. 评价函数 V 根据 x 和 y 的二进制编码规则为每一状态 (x,y) 赋予在其上为真的命题变量集合.

最后，我们可定义模型有效公式 $\Theta\models[K_\mathrm{S}\neg K_\mathrm{P}(x,y)][K_\mathrm{P}(x,y)][K_\mathrm{S}(x,y)](E_x^4\wedge E_y^{13})$ 来表示数对(4,13)是和与积问题的唯一解. 根据文献[1,7]的分析方法，我们认为公告(1)是多余的，因为公告(2)中的“S 早已知道”表明了“P 不知道这两个数”在系统的初始环境中就已经成立了，并不是公告(1)导致的结果. 因此下文中忽略公告(1).

4 基于 MCTK 的认知难题模型检测方法

本节将扩展我们的基于 OBDD(Ordered Binary Decision Diagram)的时态认知逻辑模型检测工具 MCTK[11-12]，从而支持认知难题中的知识计算和公告逻辑的建模与验证.

4.1 动态认知模型 $\mathcal{M}$

我们提出一种新的动态认知模型(Dynamic Epistemic Model)

$$\mathcal{M}=\langle S,V,\sim_1,\cdots,\sim_n,(\varphi_1,\cdots,\varphi_m)\rangle,$$

$\mathcal{M}$ 是认知模型的一个扩展，其中

(1) $(\varphi_1,\cdots,\varphi_m)$ 是 m 个公告的有序序列，每一公告由命题认知公式(命题逻辑公式与知识模态词

的组合)表示. φ_1 表示第一个公告. 对任一 $i\in\{1,\cdots,m-1\}$, φ_{i+1}将会在 φ_i 公告之后公告.

(2) $S=W\times A$ 是(全局)状态集合,其中 W 是可能世界集合,由原子命题集合 P 编码. A 是从 0 到 m 的整数域,用于对 φ_i 公告后的每一可能世界赋予标识码 $i\in A$.

(3) $V=V_W\cup V_A$ 是评价函数 V_W 和 V_A 的组合,其中 $V_W:P\to\mathcal{P}(W)$为每一原子命题 $p\in P$ 赋予一个使得 p 为真的所有可能世界集合. 对于每一 $i\in A$, $V_A:A\to\mathcal{P}(W)$为 φ_i 公告后的每一可能世界赋予一个标识码 i. 注意, $V_A(0)$是任何公告之前的初始环境中的可能世界集合.

(4) $\sim_i=S\times S$. 我们为每一主体 $i\in Ag$ 赋予一个状态集合 S 上的认知等价关系 $\sim_i$. 简便起见,我们将前面提到的函数 $l_i:S\to L_i$ 适当扩展,用于从状态 S 中获取主体 i 的局部状态. 我们假设对于任一 $(w,a)\in S$, $l_i((w,a))$均包含 a. 这样,对于任意状态 $(w,a),(w',a')\in S$, $(w,a)\sim_i(w',a')$当且仅当 $l_i((w,a))=l_i((w',a'))$. 这意味着如果 $(w,a)\sim_i(w',a')$则 $a=a'$. 因此 w 和 w'对于主体 i 来说是不可区分的,并且主体 i 知道所有公告发生的次序.

我们的目标是将动态认知模型 $\mathcal{M}$ 构造为 $m+1$ 个认知模型 $M_0,\cdots,M_m$ 的线性时态组合,其中

$$M_0=\langle\theta\times 0,V_0,\sim_1^0,\cdots,\sim_n^0\rangle$$

是公告前的初始环境认知模型, $\theta\subseteq W$ 是整个动态认知模型 $\mathcal{M}$ 的初始可能世界集合. $V_0=V_W^0\cup V_A$,其中对每一 $p\in P$ 有 $V_W^0=V_W(p)\cap\theta$. 对任一 $i\in Ag$,我们定义 $V_A(0)=\theta$ 和 $\sim_i^0=\sim_i\cap((\theta\times 0)\times(\theta\times 0))$. 对每一 $i(=1,\cdots,m)$,认知模型 M_{i+1} 与 M_i 的时态迁移关系如图 1 所示,表示公式 φ_{i+1}将在 M_i 模型上评估,并在下一时态步被公告,公告所得的认知模型就是 M_{i+1}.

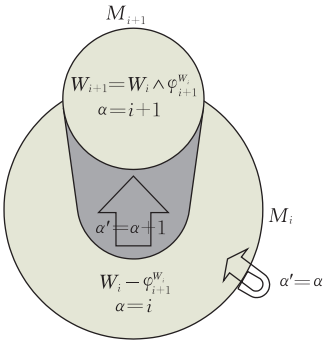


图 1　动态认知模型

也就是说 M_i 是公告 φ_i 后得到的认知模型,其定义为

$$M_i=\langle W_i\times i,V_i,\sim_1^i,\cdots,\sim_n^i\rangle,$$

其中 $W_i\subseteq W_{i-1}$是 M_i 的可能世界集合. $V_i=V_W^i\cup V_A$,其中对每一 $p\in P$ 有 $V_W^i=V_W(p)\cap W_i$. 对每一 $j\in Ag$,我们定义 $V_A(i)=W_i$ 和 $\sim_j^i=\sim_j\cap((W_i\times i)\times(W_i\times i))$. 在图 1 的 M_{i+1}中,我们定义 $W_{i+1}=W_i\cap\varphi_{i+1}^{W_i}$,其中 $\varphi_{i+1}^{W_i}$是在当前模型 M_i 的可能世界集合 W_i 上满足 φ_{i+1}的可能世界集合. 因此 $W_{i+1}=W_i\cap\varphi_{i+1}^{W_i}$意味着 W_{i+1}是可能世界集合 W_i 中满足公告 φ_{i+1}的子集. 也就是说 $W_{i+1}=\{w\in W_i\mid M_i,(w,i)\models\varphi_{i+1}\}$. 忽略整数域 A,由公告逻辑语义容易证明 $M_{i+1}\Leftrightarrow M_i|_{\varphi_{i+1}}$. 这样, m 个公告发布后,认知难题的所有解就是最后一个认知模型 M_m 的可能世界集合 W_m.

4.2　多主体有限状态程序 $\mathcal{P}_{\mathcal{M}}$

现在,我们可以将动态认知模型

$$\mathcal{M}=\langle S,V,\sim_1,\cdots,\sim_n,(\varphi_1,\cdots,\varphi_m)\rangle$$

符号化地表示为一个多主体有限状态程序

$$\mathcal{P}_{\mathcal{M}}=\langle x\cup\alpha,\theta(x)\wedge\alpha=0,\tau(x\cup\alpha,x'\cup\alpha'),O_1,\cdots,O_n,(\varphi_1,\cdots,\varphi_m)\rangle,$$

其中,

(1) $x=\{x_1,\cdots,x_k\}$是可能世界集合的编码变量. $\alpha=\{a_0,\cdots,a_{b-1}\}$是用于表示整数域 $A=[0,m]$ 的 b 个二进制编码变量,其中 $b=\lceil\log_2(m+1)\rceil$. 对每一 $i(=1,\cdots,m)$,我们用命题"$\alpha=i$"表示公式 φ_i 已被公告. 这个命题可通过二进制编码规则表示为 α 上命题公式 E_α^i. 因此 $E_\alpha^i=$TRUE 当且仅当相应的公式 φ_i 已被公告. 注意 $E_\alpha^m=$TRUE 也表示主体对话的结束. 为便于阅读,我们在下文中交替使用 $\alpha=i$ 或 E_α^i. 由于任一 S 中的全局状态可表示为 $x\cup\alpha$ 上的一个赋值,则 $\mathcal{P}(S)$中的任一状态集合也可表示为 $x\cup\alpha$ 或其真子集上的公式.

(2) θ 是 x 上的公式,表示初始环境中的可能世界集合. 命题"$\alpha=0$"表示为公式 $E_\alpha^0=\bigwedge\limits_{0\le i<b}\neg a_i$. $E_\alpha^0=$TRUE 表示所有公告还未发布. 我们称 $\theta(x)\wedge\alpha=0$ 为初始条件.

(3)状态迁移关系 $\tau(x\cup\alpha\cup x'\cup\alpha')$定义为下列公式的合取

$$\bigwedge_{1\le i\le k}(x_i'\leftrightarrow x_i)\tag{1}$$

$$(\alpha=m\to(\alpha'\leftrightarrow\alpha))\tag{2}$$

$$EpistemicConstraint(\mathcal{P}_{\mathcal{M}})\tag{3}$$

其中 $x'=\{x_1',\cdots,x_k'\}$是 x 的一个复制版本,用于编

码迁移关系中的下一状态. $\alpha'=\{a'_0,\cdots,a'_{b-1}\}$ 是 α 的一个复制版本,用于编码 α 的下一值.

(4) 对每一 $i\in\{1,\cdots,n\}$,$O_i\subseteq x\cup\alpha$ 是主体 i 的可观察变量集合. 我们假设 $\alpha\subseteq O_i$,使得所有主体均能观察(知道)公告的发布次序.

(5) 对任一命题变量 $p\in x\cup\alpha$,如果状态 $(w,a)\in V(p)$,则 $(w,a)(p)=\mathrm{TRUE}$,这里在 $\mathcal{P}_{\mathcal{M}}$ 中忽略 V.

如图 1 所示,在迁移关系 τ 中,式(1)约束所有认知模型 $M_i(i=1,\cdots,m)$ 中对 x 的解释都与初始环境认知模型 M_0 的相同,使得所有认知模型 M_i 的可能世界集合都约束为初始环境认知模型可能世界集合的一个子集. 式(2)在最后一个认知模型 M_m 中实现"自环"迁移关系,表示问题的解维持不变.

我们在图 2 中定义认知约束构造函数 $EpistemicConstraint(\mathcal{P}_{\mathcal{M}})$,用于计算 $\mathcal{M}$ 中的认知模型 $M_i(i=1,\cdots,m)$ 的可能世界集合 W_i,并为 W_i 的每一可能世界赋予一个标识码 i. W_{curr} 和 W_{next} 分别是当前和下一认知模型的可能世界集合. 注意,初始环境的可能世界集合 $\theta(x)$ 并不能从 MCTK 的输入语言中直接提取. 但是初始环境的状态集合 $Init(x\cup\alpha)$ 可从输入语言中以关键词"INIT"为前导的语句中获得,因此我们可在图 2 第 2 行通过 OBDD 计算 $\exists\alpha\ Init(x\cup\alpha)$ 来获得 $\theta(x)$. 第 4 行的公式 $\varphi_{i+1}^{W_{\mathrm{curr}}}$ 在上一个认知模型 M_i 的可能世界集合 W_{curr} 上计算满足公告 φ_{i+1} 的可能世界集合. 公式 $\varphi_{i+1}^{W_{\mathrm{curr}}}$ 的构造方法如等式(4)所示.

$$\varphi^W=\begin{cases}\text{The OBDD of }\varphi, & \varphi\in x;\\ \neg f^W, & \varphi=\neg f;\\ f^W\wedge g^W, & \varphi=f\wedge g;\\ \forall(x-O_i)(W\rightarrow f^W), & \varphi=K_i f\end{cases}\tag{4}$$

```
OBDD  function EpistemicConstraint(P_M)
1     OBDD R := TRUE;
2     OBDD W_next, W_curr := ∃α Init(x∪α); /*W_curr := θ(x)*/
3     for i := 0 to m−1 do
4       W_next := W_curr ∧ φ_{i+1}^{W_curr};
5       R := R ∧ ((α = i ∧ W_next) → α' = α+1);
6       R := R ∧ ((α = i ∧ W_curr ∧ ¬W_next) → α' = α);
7       W_curr := W_next;
8     return R;
```

图 2 认知约束构造函数

我们稍后解释等式(4). 从图 2 第 4 行中可看到 W_{next} 是 W_{curr} 中满足公告 φ_{i+1} 的可能世界集合. 为了构造 M_i 和 M_{i+1} 的线性时态关系,我们通过第 5、6 行在迁移关系中增加两个约束. 第 5 行约束只有在满足 W_{next} 的当前认知模型 M_i 的可能世界子集中,α 才会在下一步中加 1. 在与式(1)的共同约束下,第 5 行可保证下一认知模型的可能世界集合为 W_{next},同时赋予每一可能世界一个标识码 $i+1$. 第 6 行约束在不满足 W_{next} 的当前认知模型 M_i 的可能世界子集中,α 的值将在下一步中保持不变. 同样在与式(1)的共同约束下,第 6 行可保证当前认知模型中不满足下一认知模型的可能世界仍然保留在当前认知模型中. 另一个增加此约束的重要原因是可保证构造出来的迁移关系是完全的(total)[13],这意味着对动态认知模型中的任一状态,至少存在一个迁移使得当前状态变迁到另一状态. 这种迁移关系的完全性是多数主流模型检测技术所要求的.

由上述关于多主体有限状态程序 $\mathcal{P}_{\mathcal{M}}$ 的定义,我们可通过图 3 中的一个简单函数计算出认知模型 $M_i(i=0,\cdots,m)$ 的可能世界集合(表示为 $G^i_{\mathcal{P}_{\mathcal{M}}}$).

```
OBDD  function G(P_M, int i)
1     OBDD R := θ(x);
2     for j := 0 to i−1 do
3       R := R ∧ φ^R_{j+1};
4     return R;
```

图 3 计算 M_i 的可能世界集合 $G^i_{\mathcal{P}_{\mathcal{M}}}$ 的函数

4.3 MCTK 中知识与时间模型检测算法的实现

本节考虑如何在 MCTK 中实现基于 OBDD 的知识和时间模型检测算法. 该算法涉及到两种关于知识模态词的模型检测算法,一种是基于某一认知模型的,而另一种是基于整个动态认知模型的. 第 2 种知识检测算法同时支持时态算子.

我们首先考虑基于认知模型的知识检测算法. 事实上,该算法就是等式(4)的最后一行公式. 等式(4)的前 3 行布尔构造是平凡的,我们重点关注 $\varphi=K_i f$ 的情况,这一情况由命题 2 定义.

在证明命题 2 之前,我们首先介绍局部(local)命题[14]的概念,然后给出命题 1. 一个 i-local 命题在认知关系 $\sim_i$ 导出的状态等价类的所有状态中均有相同的解释. 形式地,给定一个认知模型 M 和一个主体 i,一个公式 φ 是 i-local 当且仅当对 M 的任一状态 s,如果 $M,s\models\varphi$,则对于任一满足 $s\sim_i s'$ 的状态 s',$M,s'\models\varphi$. 这样,i-local 公式是否成立仅取决于主体 i 的局部状态.

命题 1. 给定一个由多主体有限状态程序 $\mathcal{P}_{\mathcal{M}}$ 生成的认知模型 $M_j(j=0,\cdots,m)$,则一个公式 φ 在 M_j 中是 i-local 当且仅当存在一个 O_i 上的命题公式 ψ 使得 $M_j\models(\varphi\leftrightarrow\psi)$,即对任一可能世界 $w\in W_j$ 有 $M_j,(w,j)\models(\varphi\leftrightarrow\psi)$.

由于本文考虑的认知模型可视为一个“平坦的”模型,即该模型包含一个初始状态集合,其中任一状态均不会迁移到新的状态.然后在此平坦的认知模型上计算主体知识.然而我们在文献[11-12]中提出的知识模型检测算法是基于整个系统的有限状态空间的,即认知模型的状态集合是从初始状态出发,不断展开迁移关系得到的系统全局可达状态集合.由于关于知识的模型检测算法思想都是基于局部命题语义的,因此命题 1 的证明过程类似于文献[11]中的命题 1,只是各自考虑的状态集合有所不同.这里忽略证明过程.

命题 2.　给定一个由多主体有限状态程序 $\mathcal{P}_{\mathcal{M}}$ 生成的认知模型 $M_j(j=0,\cdots,m)$ 和一个命题认知公式 φ,则 $M_j\models K_i\varphi\Leftrightarrow\forall(x\cup\alpha-O_i)(G^j_{\mathcal{P}_{\mathcal{M}}}\rightarrow\varphi^{G^j_{\mathcal{P}_{\mathcal{M}}}})$.

与上述关于命题 1 的解释相同,命题 2 的证明过程类似于文献[11]的命题 3.这里忽略证明过程.

现在,我们可以通过下式用 OBDD 计算 M 中的全局可达状态集合 $G(\mathcal{P}_{\mathcal{M}})$:

$$\mathrm{lfp}Z\Big[(\theta(x)\wedge E^0_\alpha)\vee(\exists(x\cup\alpha)(Z\wedge\tau(x\cup\alpha,x'\cup\alpha')))\Big(\frac{x'\cup\alpha'}{x\cup\alpha}\Big)\Big],$$

其中 $\mathrm{lfp}Z\xi(Z)$ 是从 $x\cup\alpha$ 上的布尔公式到 $x\cup\alpha$ 上的布尔公式的操作 ξ 的一个最小不动点.$\psi\Big(\frac{x'\cup\alpha'}{x\cup\alpha}\Big)$ 是将 ψ 中的 $x'\cup\alpha'$ 变量替换为 $x\cup\alpha$ 中的相应变量之后得到的公式.$G(\mathcal{P}_{\mathcal{M}})$ 可由图 4 所示函数计算得到.注意图 4 并不是上述最小不动点的直接实现,而是做了一些优化.在图 4 中,第 4 行用 *upper* 保存当前的可达状态集合.第 5 行从上次新产生的状态集合 *lower* 开始计算所有可能的直接后继状态的集合 *image*,这时得到的 *image* 是建立在 $x'\cup\alpha'$ 之上的.第 6 行把 *image* 变换为 $x\cup\alpha$ 之上的公式后将其合并到可达状态集合 R 中.第 7 行将产生的新状态集合保存到 *lower* 后重新循环搜索可达状态,直至没有任何新的状态产生为止,即 $lower(x\cup\alpha)==$ FALSE.

OBDD	function $G(\mathcal{P}_{\mathcal{M}})$
1	OBDD $lower(x\cup\alpha)=init(\mathcal{P}_{\mathcal{M}})$;
2	OBDD $R(x\cup\alpha)=init(\mathcal{P}_{\mathcal{M}})$;
3	while ($lower(x\cup\alpha)$!=FALSE) do
4	OBDD $upper(x\cup\alpha)=R(x\cup\alpha)$;
5	OBDD $image(x'\cup\alpha')=\exists x\cup\alpha.(lower(x\cup\alpha)\wedge\tau(x\cup\alpha,x'\cup\alpha'))$;
6	$R(x\cup\alpha)=R(x\cup\alpha)\vee image\Big(\frac{x'\cup\alpha'}{x\cup\alpha}\Big)$;
7	$lower(x\cup\alpha)=R(x\cup\alpha)\wedge\neg upper(x\cup\alpha)$;
8	return $R(x\cup\alpha)$;

图 4　可达状态集合计算函数 $G(\mathcal{P}_{\mathcal{M}})$

从 $G(\mathcal{P}_{\mathcal{M}})$ 的计算过程和 $\mathcal{P}_{\mathcal{M}}$ 的结构可知,$G(\mathcal{P}_{\mathcal{M}})$ 实际是所有 $M_i(i=0,\cdots,m)$ 的状态集合 S_i 的并集.由于每一 S_i 是附带标识码 $\alpha=i$ 的可能世界集合 W_i,因此可能世界集合 W_i 可由公式 $\exists\alpha(G(\mathcal{P}_{\mathcal{M}})\wedge\alpha=i)$ 计算得到,其中通过存在量化将附加的公告变量 α 消除了.这样,最后一个公告 φ_m 发布之后,认知难题的所有解就是由公式 $\exists\alpha(G(\mathcal{P}_{\mathcal{M}})\wedge\alpha=m)$ 表示的可能世界集合.

实际上,状态集合 S_i 也可由图 3 的函数计算得到,因此可能世界集合 W_i 也可由公式 $\exists\alpha G(\mathcal{P}_{\mathcal{M}},i)$ 表示.公式 $\exists\alpha G(\mathcal{P}_{\mathcal{M}},i)$ 的计算只需用到初始可能世界集合 $\theta(x)$ 以及公告序列,而 $\exists\alpha(G(\mathcal{P}_{\mathcal{M}})\wedge\alpha=i)$ 需要构造迁移关系 $\tau(x\cup\alpha,x'\cup\alpha')$ 并计算全局可达状态集合 $G(\mathcal{P}_{\mathcal{M}})$,因此我们相信公式 $\exists\alpha G(\mathcal{P}_{\mathcal{M}},i)$ 的计算效率高于 $\exists\alpha(G(\mathcal{P}_{\mathcal{M}})\wedge\alpha=i)$.这样,如果我们只需得到该问题的解,则执行公式 $\theta(x)$ 和 $\exists\alpha G(\mathcal{P}_{\mathcal{M}},m)$ 的计算即可.如果我们还需对主体知识在时态上的变化进行检测,则两种可能世界集合的计算方法都是必需的.

如图 1 所示,通过将动态认知模型 $\mathcal{M}$ 构造为 $m+1$ 个认知模型 $M_0,\cdots,M_m$ 的线性时态组合,我们可以符号化地表示 $\mathcal{M}$ 为多主体有限状态程序 $\mathcal{P}_{\mathcal{M}}$,这与我们在文献[11-12]中所用的带 n 个主体的有限状态程序兼容.这样,我们不仅可以对认知难题进行公告逻辑建模并搜索所有解,而且可以沿用我们在文献[11-12]中开发的模型检测算法验证问题相关的一些时态认知性质.这一特性不能被 DEMO、MCK 和 MCMAS 等认知逻辑模型检测工具完全支持.关于我们的时态认知逻辑模型检测方法并不是本文的主要内容,详细内容请查阅文献[11-12].我们将在下文通过实例说明这一特性.

5　和与积问题的 MCTK 建模

我们的模型检测工具 MCTK[11] 是在时态逻辑模型检测工具 NuSMV 2.1.2 基础上扩展实现的,在 NuSMV 输入语言基础上扩展支持对主体可观察变量的定义,使得我们可以在 MCTK 中描述和与积问题的多主体有限状态程序(图 5 所示):

$$\mathcal{P}_\Theta=\langle x\cup\alpha,\theta(x)\wedge\alpha=0,\tau(x\cup\alpha,x'\cup\alpha'),O_S,O_P,(K_S\neg K_P(x,y),K_P(x,y),K_S(x,y))\rangle.$$

```
1  MODULE main()
2  VAR
3      a:      0..3;       --announcement variable, m=3
4      x:      2..100;     y : 2..100;
5      sum:    4..200;     product : 4..10000;
6      S:      SumAgent(sum, a);
7      P:      ProductAgent(product, a);
8  INIT        x>1 & y>x & x+y<=100 &
9              sum=x+y & product=x*y &
10             a=0;
11 TRANS       next(x)=x & next(y)=y &
12             next(sum)=sum & next(product)=product &
13             (a=3 -> next(a)=a);
14 MODULE SumAgent(Observable sum, Observable av)
15 MODULE ProductAgent(Observable product,Observable av)
```

图 5 和与积问题的多主体有限状态程序

在图 5 中,第 3 行定义一个整型变量 a,这一变量名专为表示公告动作 α 而保留.根据第 3 行的分析结果,我们只对公告(2)、(3)和(4)建模,因此变量 a 的取值范围是[0,3].整数变量 x 和 y 表示数对.为了使主体 S 和 P 分别只能观察到和与积,我们建立两个附加的整型变量 *sum* 和 *product*,分别表示 x 和 y 的和与积.因此,$\mathcal{P}_\Theta$的变量集合 $x \cup \alpha$ 是 a,x,y,*sum* 和 *product* 的二进制编码变量的集合.

第 8~9 行根据题意约束这 4 个整型变量的初始值,构造出$\mathcal{P}_\Theta$的初始条件 $\theta(x)$.第 10 行意味着在初始环境中没有发生任何公告.第 11~12 行表示式(1),使得 4 个整型变量的初始值一旦在 $\theta(x)$的约束下被随机选定,则始终保持不变.注意到在 MCTK 中,用"next(v)"表示变量 v 的下一版本 v'.第 13 行表示式(2).在此我们并没有显式地构造式(3),即 $EpistemicConstraint(\mathcal{P}_\Theta)$,这是因为当前实现的 MCTK 输入语言不支持可能世界集合的表示和 φ^W 的计算.我们在 MCTK 内部实现函数 $EpistemicConstraint(\mathcal{P}_\Theta)$并在 MCTK 构建了基于 OBDD 的基本迁移关系之后调用.这样,我们就可以通过 OBDD 合取操作在 OBDD 表示的基本迁移关系中加入认知约束 $EpistemicConstraint(\mathcal{P}_\Theta)$.

第 14~15 行是主体 S 和 P 的模块定义,其中的形式参数 *sum* 和 av 对于 S 是可观察的,而 *product* 和 av 对于 P 是可观察的.第 6~7 行声明主体实例 S 和 P,S 和 P 的实际参数分别是{*sum*,*a*}和{*product*,*a*},这意味着 S 的可观察变量集合 O_S是整型变量{*sum*,*a*}的二进制编码变量集合,而 P 的可观察变量集合 O_P是整型变量{*product*,*a*}的二进制编码变量集合.将公告变量 a 加入主体 S 和 P 的可观察变量集合的目的是使得他们知道公告的发布次序.

6 实验结果

我们已成功实现本文提出的认知难题模型检测工具 MCTK,其中的算法是基于科罗拉多大学 Fabio Somenzi 开发的 CUDD OBDD 软件包实现的.我们在一台配置为 P8600 2.4GHz Core 2 Duo 处理器和 4GB 内存,操作系统为 Ubuntu 8.10 Linux 系统的笔记本电脑上用 MCTK 验证了和与积问题.验证所需时间约 48s,包括符号化模型构建时间和公式 $G(\mathcal{P}_\Theta,3)$的计算时间.消耗 12425.712KB 内存.模型检测结果是 $G(\mathcal{P}_\Theta,3)\Leftrightarrow(E_x^4 \wedge E_y^{13})$.这意味着当且仅当初始世界是(4,13)时,才可能依次发布公告(2)、(3)和(4).换句话说,和与积问题的唯一解是(4,13).这一结果与文献[1,7]得到的一致.

为了与文献[1,7]的试验结果进行比较,我们在同一实验平台上,基于动态认知模型检测工具 DEMO 验证了文献[1]中图 1 描述的和与积问题,得到的结果也是只有唯一解(4,13).其中采用 Haskell 编译器 GHC 6.8.2 对该问题模型描述程序进行编译.执行该编译程序得到结果所需时间大约是 12min20s.消耗 9319.284KB 内存.显然,除了消耗的内存多一些外,我们的工具 MCTK 的效率与 DEMO 相比占有绝对优势.初略分析的结果是 DEMO 可以看作是基于显式状态建模的模型检测方法,其内部可能没有采用优化技术,因此基本上只能对单一状态进行操作.而我们的模型构建和公式检测都采用基于 OBDD 的符号化方法,可以用公式紧凑地表示状态集合并在其上执行各种高效的逻辑运算.因此与 DEMO 相比,我们的符号化方法可处理更大规模的问题,验证效率更高.

在 MCTK 的具体实现中,我们不仅支持时态认知逻辑 $ECKL_n$(CTL*[15]与认知模态词的组合)的模型检测,而且还在 NuSMV 的 CTL 模型检测算法上扩展支持认知模态词的检测检测算法,因此支持计算树认知逻辑 CTLK 的模型检测,对于时态部分只包含 CTL 时态算子的公式,CTLK 的模型检测效率更高.我们也对和与积问题验证了一些时态认知性质.我们首先验证了 CTL 性质

$$AG(a=3 \rightarrow (x=4 \,\&\, y=13)),$$

其中时态算子"AG"表示"任一路径的任一状态"或"始终(always)".验证所需时间约 49s.消耗 12425.712KB 内存.这一性质表示一旦第 3 个(最后一个)公告发布了,则认知模型中只有一个可能世

界(4,13),这就是该问题的解.这一性质说明了我们构建的动态认知模型是正确的.

第 2 个被验证性质是一个 CTLK 公式:

$$AG(((x=4\ \&\ y=13\ \&\ a=2)->((sum\ K\ x)\ \&\ (sum\ K\ y)))\ \&\ ((!(x=4\ \&\ y=13)\ \&\ a=2)->!((sum\ K\ x)\ \&\ (sum\ K\ y))))),$$

其中"$sum\ K\ x$"表示公式 $K_S x$,"$sum\ K\ y$"表示公式 $K_S y$.验证所需时间约 60s.消耗 12425.712KB 内存.这一性质表示如果数对(x,y)是(4,13),则 P 宣告他不知道这两个数之后,S 将会知道(x,y)就是(4,13);但是如果数对(x,y)不是(4,13),则 P 宣告他不知道这两个数之后,S 仍然不知道这两个数.我们还验证了第 3 个 CTL 性质

$$(x=4\ \&\ y=13)->AG(a=2->(x=4\ \&\ y=13)).$$

这一性质表示如果(x,y)的初始值为(4,13),则(4,13)必然保留在 P 宣告他不知道这两个数之后得到的认知模型中.结合第 2 个性质,可推断在 P 发布公告后 S 将宣告他知道这两个数.这就是该问题的实际情况.

7　结论及未来工作

本文首先用公告逻辑对和与积问题进行建模,然后用 MCTK 的输入语言即多主体有限状态程序描述该问题.该问题的动态认知模型可通过 MCTK 从多主体有限状态程序自动编译生成.结合我们在文献[11-12]中开发的时态认知逻辑符号化模型检测方法,我们可将问题求解和时态认知模型检测的算法转化为一系列基于 OBDD 的运算.实验结果表明了我们的方法的正确性和高效性.将来我们考虑在此基础上扩展更常见的通信方式(如点对点通信、组群通信和广播等)的建模与验证方法,还可考虑支持多协议会话的建模等,从而扩展动态认知的建模能力并提高验证算法效率,有望解决更多的认知难题.本文的工作也可进一步扩展到更复杂的安全协议形式化验证领域.

参　考　文　献

[1] van Ditmarsch H P, Ruan J, Verbrugge L C. Model checking sum and product//Proceedings of the 18th Australian Joint Conference on Artificial Intelligence (AI 2005). Sydney, Australia, 2005: 790-795

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

[3] van Ditmarsch H P, van der Hoek W, Kooi B P. Dynamic epistemic logic and knowledge puzzles//Proceedings of the 15th International Conference on Conceptual Structures (ICCS 2007). Sheffield, UK, 2007: 45-58

[4] van Eijck J. Dynamic epistemic modelling. Centrum voor Wiskunde en Informatica, Amsterdam: Technical Report CWI Report SEN-E0424, 2004

[5] Freudenthal H. Formulation of the sum-and-product problem. Nieuw Archief voor Wiskunde, 1969, 3(17): 152

[6] Freudenthal H. Solution of the sum-and-product problem. Nieuw Archief voor Wiskunde, 1970, 3(18): 102-106

[7] van Ditmarsch H P, Ruan J, Verbrugge R. Sum and product in dynamic epistemic logic. Journal of Logic and Computation, 2008, 18(4): 563-588

[8] Luo X, Su K, Sattar A, Reynolds M. Verification of multi-agent systems via bounded model checking//Proceedings of the 19th Australian Joint Conference on Artificial Intelligence (AI 2006). Hobart, Australia, 2006: 69-78

[9] Gammie P, van der Meyden R. MCK: Model checking the logic of knowledge//Proceedings of the 16th International Conference on Computer Aided Verification (CAV 2004). LNCS 3114. Boston, MA, USA, 2004: 479-483

[10] Lomuscio A, Raimondi F. MCMAS: A model checker for multi-agent systems//Proceedings of the TACAS-2006. LNCS 3920. Vienna, Austria, 2006: 450-454

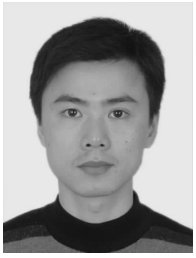
[11] Su K, Sattar A, Luo X. Model checking temporal logics of knowledge via OBDDs. The Computer Journal, 2007, 50(4): 403-420

[12] Su K. Model checking temporal logics of knowledge in distributed systems//Proceedings of the 19th National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence (AAAI 2006). San Jose, California, USA, 2004: 98-103

[13] Clarke E M, Grumberg O, Peled D A. Model Checking. Cambridge, Massachusetts: The MIT Press, 2000

[14] Engelhardt K, van der Meyden R, Moses Y. Knowledge and the logic of local propositions//Gilboa I ed. Proceedings of the 7th conference on Theoretical Aspects of Rationality and Knowledge (TARK 1998). San Fransisco: Morgan Kaufmann Publishers Inc., 1998: 29-41

[15] Su Kai-Le, Luo Xiang-Yu, Lv Guan-Feng. Symbolic model checking for CTL*. Chinese Journal of Computers, 2005, 28(11): 1798-1806(in Chinese)
(苏开乐,骆翔宇,吕关锋. 符号化模型检测 CTL*. 计算机学报, 2005, 28(11): 1798-1806)



LUO Xiang-Yu, born in 1974, Ph.D., associate professor. His research interests include model checking, temporal logics, epistemic logics, reasoning about knowledge, multi-agent systems, verification of security protocols.

SU Kai-Le, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include model checking, reasoning about knowledge, non-monotonic reasoning, multi-agent systems, modal logics, temporal logics, probability reasoning, verification of security protocols, logic programming.

GU Ming, born in 1962, professor, Ph. D. supervisor. Her research interests include formal methods, information security and software middleware.

Background

Epistemic riddles, such as muddy children puzzles, sum and product problem, and "what sum" problem, are usually considered as multi-agent systems (MAS), an ordered sequence of announcements about the knowledge or ignorance of agents involved will be issued successively, which result in some new knowledge about the fact in the environment. In a multi-agent system, each agent has its local state in each global state of the system. It is commonly known that each agent can only knows (observe) its local state. In MAS community a particular emphasis is given to the formal representation of the mental attitudes of agents, such as agents' knowledge, beliefs, desires, intentions and so on. Dynamic epistemic logic was developed to study the changes of agents' individual knowledge or group knowledge caused by communication about agents' or group knowledge. Public Announcement Logic is a kind of dynamic epistemic logic. It is very suitable for modeling the sum and product problem because it is able to describe the actions of public announcement about agents' or group knowledge. van Ditmarsch et al. claimed that they was the first to use an automated model checker DEMO to tackle the problem. They stated that the problem cannot be tackled in two state-of-the-art symbolic model checkers for temporal logics of knowledge, MCK 0.2.0 and MCMAS 0.7, because they does not support checking epistemic formulas as preconditions in their specification languages.

In the earlier version of this paper published in WI-IAT 2008, the authors developed a preliminary model checking algorithm to reduce the model checking problem in public announcement logic to a series of symbolic (Boolean) manipulations of the set of states that satisfies a given epistemic formula, with Ordered Binary Decision Diagrams (OBDD). However, in that paper the authors just focus on presenting the basic algorithm for finding solutions, the complete formal model couldn't be shown clearly due to the limited space. In this paper, the authors propose a new dynamic epistemic model for public announcement logic, which is the linear temporal combination of the epistemic model of initial environment and the epistemic model resulted from each announcement. The dynamic epistemic model is compatible with the model used in MCTK, a BDD-based model checker for temporal epistemic logics developed by the authors. In this modeling way the model checking algorithm of MCTK can be applied to the verification of temporal epistemic properties, such characteristic is not supported by DEMO. On the other hand, the modeling way can be thought of as an extension of traditional model checkers for temporal epistemic logics, to support the modeling and verification of public announcement logic. Therefore, this could be a solution to the limitation of the traditional model checkers for temporal epistemic logics for public announcement logic. The proposed method is implemented in MCTK and the sum and product problem is verified. The experimental results show that the proposed method is correct and efficient.