系 统 科 学 与 数 学
J. Sys. Sci. & Math. Scis.
28(8) (2008, 8), 1020–1029

基于吴方法的多值模型检验*

赵 林

(北京交通大学轨道交通控制与安全国家重点实验室, 北京 100044;
中国科学院成都计算机应用研究所, 成都 610041)

吴 尽 昭

(中国科学院成都计算机应用研究所, 成都 610041)

摘要 大型复杂系统的开发过程中不可避免的涉及到非确定或不一致信息的处理, 而多值模型检验作为经典模型检验的一种扩展, 是处理和分析包含此类信息模型的一种有效手段. 提出了一种系统化的多值逻辑 (涵盖经典逻辑) 的代数表示方法, 使用吴方法的基本思想和框架实现复杂系统形式验证中基于多值逻辑的模型检验的代数化, 建立了通过吴方法实现多值模型检验技术的整体框架. 这种代数化的多值模型检验方法可以作为现有方法的有力补充.

关键词 多值模型检验, 拟布尔逻辑, 多项式, 吴方法.

MR(2000) 主题分类号 68N30

1 引 言

模型检验 [1] 是一种用于验证有限状态并发系统的自动技术. 与传统的模拟、测试和推理验证相比, 它的主要优势在于能够保证验证的完备性并且无需专门人员的人工引导. 此外, 当检查到系统存在潜在缺陷时, 能给出反例来指示出错的原因. 在过去的十几年当中, 模型检验已被广泛地应用于并发系统如集成电路、通讯协议等的正确性验证.

现有的模型检验技术只能在经典逻辑之上进行自动推理. 然而, 在实际的软硬件开发过程中有很多问题是经典逻辑无法处理的. 一类典型的问题就是对非确定信息的处理. 系统的非确定性一般来源于必要信息的缺失, 这在需求分析的时候经常碰到. 另外, 建立在经典逻辑上的验证方法同样不适用于包含不一致信息的系统. 例如, 在需求工程中, 模型往往因为集成由不同开发人员所开发的组件而引入不一致信息.

通常, 对包含非确定或不一致信息系统的推理是在多值逻辑的框架下进行的. 因为多值逻辑可以允许 “真” 和 “假” 以外的逻辑值来显式的刻画非确定性或不一致性. 例如, 可

国家 973 计划 “数学机械化及其在信息技术中的应用”(2004CB318000) 、 “需求工程 – 对复杂系统的软件工程的基础研究 ” (2007CB310800) 和国家 863 计划 “基于代数符号计算的新型软件形式化验证技术和支持工具 ” (2007AA01Z143) 项目资助.

收稿日期: 2007-12-30.

以用 “不知道 (Don’t know)” 、 “可能 (Maybe)” 作为逻辑真值. 事实上, 已经有人研究了三值逻辑和四值逻辑上的模型检验. 文 [2] 利用三值逻辑来解释模型抽象后的静态分析结果, 而文 [3] 使用四值逻辑对门级电路的抽象模型进行推理验证. Chechik 等人 [4] 提出使用统一的逻辑簇 – 拟布尔逻辑 (Quasi-Boolean Logics) [5−6] 来进行推理, 将符号模型检验推广到任意的多值逻辑. 文 [7] 中讨论了一个基于多路判定图 (MDDs)[8] 的多值符号模型检验软件 χChek [9]. 多路判定图是二叉判定图 (BDD)[10] 的扩展, 以压缩的方式存储多值逻辑函数而且可以很高效的实现各种逻辑运算. 然而, 基于 MDD 实现的模型检验软件存在一些固有的缺点. 比如, 存在状态空间爆炸问题; 对某些系统, 尤其是软件系统的验证效果不理想等. 因此, 有必要对多值逻辑的符号表示方法做进一步的分析和研究.

符号计算是数学与计算机科学的交叉学科, 在几何定理证明、机器人学、计算机视觉和信号处理等很多重要的科研领域有着非常成功的应用. 吴方法 [11] 是吴文俊先生由中国传统数学中的机械化思想出发, 借助 30 年代 Ritt[12] 的理论工作, 针对几何定理机器证明问题研究和发展的数学机械化方法. 吴方法在几何定理的机器证明领域的成功应用, 为国际自动推理的研究开辟了新的前景.

多值模型检验从代数的角度看也是符号计算, 因而可以使用吴方法的总体思想和框架. 本文提出将多值模型检验问题转化为吴方法等符号计算算法适用的代数问题, 在多项式代数系统中研究和开发验证算法. 我们给出了一种系统化的多值逻辑 (涵盖经典逻辑) 的多项式表示方法, 可以根据给定的逻辑公式自动地构造其对应的多项式表示形式. 其基本思想是把对多值逻辑公式的赋值集合视为有理数域上的一组多项式的公共零点集. 例如, 多项式组 $\{x_0, x_1-2, x_2-1\}$ 可以用来描述逻辑赋值 $\langle x_0=0, x_1=2, x_2=1\rangle$. 基于多值逻辑的这种代数表示, 多值模型检验问题转化为通过吴方法计算系统特征列, 并判断初始状态所表示的零点是否在满足某性质的零点集合中, 从而判断系统是否具有该性质.

Avrunin 等人 [13] 最早将 Groebner 基理论应用于布尔模型检验, 通过布尔环上多项式理想的 Groebner 基来表示系统模型和待验证的性质, 从而将模型检验问题转化为理想的归属问题. 而本文作者 [14] 最早提出利用特征列方法来构造表示系统模型和待验证的性质的多项式的公共零点集, 从多项式零点集的角度来研究基于布尔逻辑的模型检验问题. 由于多项式的零点被限制在 $\{0,1\}^n$(n 为变量的个数) 内, 上述方法都必须在逻辑公式的多项式表示中加入额外的一组多项式 $\{x_1^2+x_1, x_2^2+x_2, \cdots, x_n^2+x_n\}$. 基于布尔逻辑的模型检验可以看作是多值模型检验的一个特例. 本文提出的代数化的多值模型检验方法在采用布尔逻辑进行推理时, 可以退化为文 [13–14] 的工作. 另外, 使用本文提出的逻辑函数的代数表示方法来表示布尔逻辑, 可以避免 $\{x_1^2+x_1, x_2^2+x_2, \cdots, x_n^2+x_n\}$ 类型多项式的引入.

2 多值模型检验

引入多值逻辑来精确刻画系统的不确定性和不一致性是一种有效的途径, 多值模型检验就是针对带有不确定和不一致信息的系统而建立起来的一套形式验证理论, 是经典模型检验的一种扩展. 本节简单介绍一下多值模型检验的基本概念和原理.

2.1 拟布尔逻辑

多值逻辑通常定义在格上, 这使得经典逻辑的许多性质得以保持, 例如结合律、分配律、德 · 摩根律和双重否定律 $(\neg\neg a = a)$.

定义 1 设 $(\mathcal{L}, \sqsubseteq)$ 是一个偏序集，如果 $\mathcal{L}$ 中任意两个元素 (a, b) 都有最小上界和最大下界，分别记为 $a \sqcup b$ 和 $a \sqcap b$, 则称 $(\mathcal{L}, \sqsubseteq)$ 为格.

定义 2 一个有限格 $(\mathcal{L}, \sqsubseteq)$ 是拟布尔格，当且仅当在其上可以定义满足如下性质的单目运算 $\neg$.

$$\neg(a \sqcap b) = \neg a \sqcup \neg b, \quad \neg\neg a = a,$$

$$\neg(a \sqcup b) = \neg a \sqcap \neg b, \quad a \sqsubseteq b \Leftrightarrow \neg a \sqsupseteq \neg b.$$

如果一个逻辑系统，其真值可以构成一个拟布尔格，则称这种逻辑为拟布尔逻辑. 格上的最小上界和最大下界运算分别对应逻辑中的析取和合取运算，而满足定义 2 的单目运算 $\neg$ 对应逻辑非运算. 多值模型检验选择拟布尔逻辑作为推理的基础，一个重要的原因就是格上定义的单调函数存在最小不动点和最大不动点 [15], 而不动点运算是符号化的模型检验算法的核心. 图 1 给出了几个逻辑格的例子，不难看出，定义在 2- 值拟布尔格上的逻辑实际上就是经典布尔逻辑.

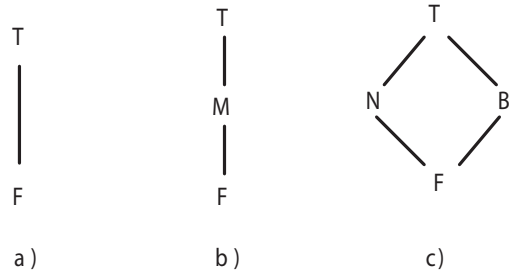

图 1 a) 2 值拟布尔格 b) 3 值拟布尔格 c) 4 值拟布尔格

2.2 多值 CTL 模型检验

分支时态逻辑 CTL[1] 是由 Eermson 和 Clarke 在 1979 年首先提出的一种时态逻辑，当时用于表示程序的时态性质. 在分支时态逻辑中，时间结构是一种无限树结构，称之为 Kripke 结构. 多值模型检验中，模型用 Kripke 结构的多值扩展 (mv–Kripke)[6] 来刻画. 在 mv–Kripke 结构中，状态之间的迁移关系可以取 “真 ” 和 “假 ” 之外的逻辑值，每个状态包含的原子命题也可以是多值的，图 2 给出了一个 mv–Kripke 结构和其相应展开树的例子. 待检验的性质用扩展了多值语义的 CTL(mv-CTL) 公式表示. Chechik 等人 [4] 证明了 mv-CTL 公式满足如下性质.

定理 1 对于一个有限 mv–Kripke 结构 $\mathcal{M}$, 下述公式成立

$$AG\varphi = \upsilon y.\ (\varphi \wedge AXy), \quad EG\varphi = \upsilon y.\ (\varphi \wedge EXy),$$

$$AF\varphi = \mu y.\ (\varphi \vee AXy), \quad EF\varphi = \mu y.\ (\varphi \vee EXy),$$

$$A[\varphi U \psi] = \mu y.\ (\psi \vee (\varphi \wedge AXy), \quad E[\varphi U \psi] = \mu y.\ (\psi \vee (\varphi \wedge EXy)),$$

其中，逻辑连接词 $\neg$, $\wedge$ 和 $\vee$ 的定义与经典逻辑相同. 路径量词 A 表示对于所有路径， E

表示存在某个路径. 时序算子 X 指示下一个状态， F 指示将来某个状态， G 指示所有状态， U 指示直到某个状态.

根据定理 1, mv-CTL 公式的值可以通过计算相应函数的最小 (大) 不动点来获得. 例如, 检验某个性质 $E[\varphi U\psi]$, 我们所需要做的就是计算函数 $f(y)=\psi\vee(\varphi\wedge EXy)$ 的最小不动点. 要获得更多与符号多值模型检验算法相关的内容，读者可以参阅文 [6].

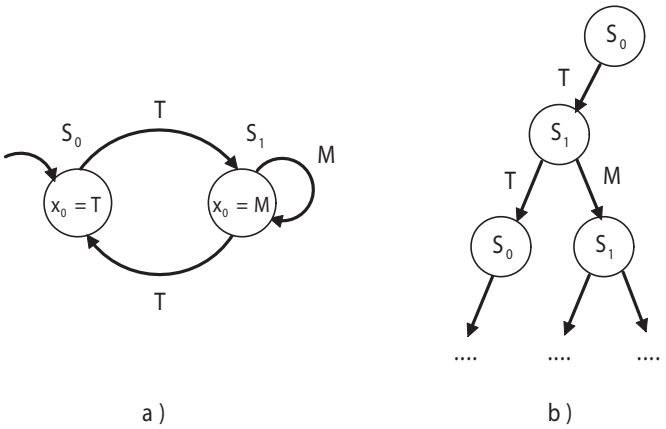

图 2　a) mv–Kripke 结构　b) 无限计算树

3　基于吴方法的多值模型检验

在多值模型检验中, 模型用 mv-Kripke 结构来刻画, 而 mv-CTL 公式表达对待验证的模型的功能正确性要求. 在符号化的算法中， mv-Kripke 结构和 mv-CTL 公式都用 MDD 作为数据结构, 通过计算相关公式的不动点来达到验证的目的. 本节提出了多值逻辑的一种系统化的代数表示方法, 并将系统的状态、转移关系以及要验证的性质都转化成多项式, 而不是 MDD, 然后在多项式代数系统中研究验证算法.

3.1　拟布尔逻辑的多项式表示

记 $\mathbb{Q}[x_1,x_2,\cdots,x_n]$ 为有理数域上的多项式环, $PS=\{p_1,p_2,\cdots,p_m\}$ 为 $\mathbb{Q}[x_1,x_2,\cdots,x_n]$ 的一个有限非空子集.

定义 3　令 $PS_1=\{p_1,p_2,\cdots,p_r\}$,$PS_2=\{q_1,q_2,\cdots,q_s\}$, 多项式集合上的加法 $(+)$ 和乘法 $(\times)$ 操作定义如下

$$PS_1+PS_2=\{\ p_1,p_2,\cdots,p_r,q_1,q_2,\cdots,q_s\ \},\quad PS_1\times PS_2=\{\ p_iq_j\mid 1\le i\le r,1\le j\le s\ \}.$$

为了进行符号化的编码, 需要为格 $(\mathcal{L},\sqsubseteq)$ 上的元素定义一个序函数 $\mathrm{Ord}:\mathcal{L}\to\mathbb{Q}$, 它将 $l\in\mathcal{L}$ 映射到 $\{0,1,\cdots,|\mathcal{L}|-1\}$. 例如, 可以为三值的拟布尔格 $(3\text{-QBool},\sqsubseteq)$ 定义如下的序关系

$$\mathrm{Ord}(T)=2,\quad \mathrm{Ord}(M)=1,\quad \mathrm{Ord}(F)=0.$$

定义 4 令 L 为格 $(\mathcal{L},\sqsubseteq)$ 上的拟布尔逻辑，$f\colon \mathcal{L}^n\to\mathcal{L}$ 为 L 上的任意 n 变量逻辑公式，$\{a_1,a_2,\cdots,a_n\}\in\{0,1,\cdots,|\mathcal{L}|-1\}^n$ 为对公式 f 的一个赋值．$V(ps)$ 表示多项式集合 $ps\subset\mathbb{Q}[x_1,x_2,\cdots,x_n]$ 的代数簇，而 Ord^{-1} 为序函数 Ord 的逆．

f 的多项式表示形式是一个 $|\mathcal{L}|$ - 元组

$$[\![f]\!]=\langle\,[\![f]\!]_0,\,[\![f]\!]_1,\,\cdots,\,[\![f]\!]_{|\mathcal{L}|-1}\rangle,$$

其中，$[\![f]\!]_i$ 定义为满足如下条件的多项式集合

$$\{a_1,a_2,\cdots,a_n\}\in V([\![f]\!]_i)\text{ 当且仅当 }f(\mathrm{Ord}^{-1}(a_1),\mathrm{Ord}^{-1}(a_2),\cdots,\mathrm{Ord}^{-1}(a_n))=\mathrm{Ord}^{-1}(i),$$

逻辑真值 $l\in\mathcal{L}$ 定义为

$$[\![l]\!]=\{[\![l]\!]_0,[\![l]\!]_1,\,\cdots,\,[\![l]\!]_i,\,\cdots,\,[\![l]\!]_{|\mathcal{L}|-1}\},\ \text{其中 }\mathrm{Ord}(l)=i,\ [\![l]\!]_i=\{0\},\ [\![l]\!]_{j\neq i}=\{1\}.$$

在我们的应用中，$V([\![f]\!]_i)$ 是有限簇，因此 $[\![f]\!]_i$ 存在，并且可以通过给定逻辑的真值表进行构造．注意定义 4 中常数多项式 $\{1\}$ 表示的代数簇是空集，而常数多项式 $\{0\}$ 表示的代数簇是 $\{0,1,\cdots,|\mathcal{L}|-1\}^n$ ．逻辑操作合取 $(\vee)$ 、析取 $(\wedge)$ 和非 $(\neg)$ 的定义取决于给定逻辑的真值表．下面，我们给出 3-QBool 逻辑的完整代数定义，其真值表如表 1 所示．

表 1 3-QBool 逻辑的真值表

$\wedge$	T	M	F
T	T	M	F
M	M	M	F
F	F	F	F

$\vee$	T	M	F
T	T	T	T
M	T	M	M
F	T	M	F

$\neg$	
T	F
M	M
F	T

定义 5 令 x 为任意的逻辑变量，φ 和 ψ 为任意的逻辑公式，3-QBool 逻辑的多项式表述形式如下

$$\mathrm{Ord}(T)=2,\quad \mathrm{Ord}(M)=1,\quad \mathrm{Ord}(F)=0,$$

$$[\![T]\!]=\langle\,\{1\},\,\{1\},\,\{0\}\,\rangle,\quad [\![M]\!]=\langle\,\{1\},\,\{0\},\,\{1\}\,\rangle,\quad [\![F]\!]=\langle\,\{0\},\,\{1\},\,\{1\}\,\rangle,$$

$$[\![x]\!]=\langle\,\{x\},\,\{x-1\},\,\{x-2\}\,\rangle,\quad [\![\neg x]\!]=\langle\,\{x-2\},\,\{x-1\},\,\{x\}\,\rangle,$$

$$[\![\varphi]\!]=\langle\,[\![\varphi]\!]_0,\,[\![\varphi]\!]_1,\,[\![\varphi]\!]_2\,\rangle,\quad [\![\neg\varphi]\!]=\langle\,[\![\varphi]\!]_2,\,[\![\varphi]\!]_1,\,[\![\varphi]\!]_0\,\rangle,$$

$$\begin{aligned}[\![\varphi\wedge\psi]\!]=\langle\,&([\![\varphi]\!]_0+[\![\psi]\!]_0)\times([\![\varphi]\!]_0+[\![\psi]\!]_1)\times([\![\varphi]\!]_0+[\![\psi]\!]_2)\times([\![\varphi]\!]_1+[\![\psi]\!]_0)\times([\![\varphi]\!]_2+[\![\psi]\!]_0),\\ &([\![\varphi]\!]_1+[\![\psi]\!]_1)\times([\![\varphi]\!]_1+[\![\psi]\!]_2)\times([\![\varphi]\!]_2+[\![\psi]\!]_1),\ [\![\varphi]\!]_2+[\![\psi]\!]_2\,\rangle,\end{aligned}$$

$$\begin{aligned}[\![\varphi\vee\psi]\!]=\langle\,&[\![\varphi]\!]_0+[\![\psi]\!]_0,\ ([\![\varphi]\!]_1+[\![\psi]\!]_1)\times([\![\varphi]\!]_1+[\![\psi]\!]_0)\times([\![\varphi]\!]_0+[\![\psi]\!]_1),\\ &([\![\varphi]\!]_2+[\![\psi]\!]_0)\times([\![\varphi]\!]_2+[\![\psi]\!]_1)\times([\![\varphi]\!]_2+[\![\psi]\!]_2)\times([\![\varphi]\!]_1+[\![\psi]\!]_2)\times([\![\varphi]\!]_0+[\![\psi]\!]_2)\,\rangle.\end{aligned}$$

根据定义 5, 可以将任意的 3-QBool 逻辑公式转化为多项式形式的表示，以公式 $x_1\wedge x_2$ 为例，有

$$[\![x_1\wedge x_2]\!]=\langle\,\{x_1^3-3x_1^2+2x_1,x_1x_2,x_2^3-3x_2^2+2x_2\},\ \{x_1^2-3x_1+2,x_1x_2-x_2-x_1+1,x_2^2-3x_2+2\},\ \{x_1-2,x_2-2\}\,\rangle.$$

定义 6 令 x 为任意的逻辑变量，φ 和 ψ 为任意的逻辑公式，布尔逻辑的多项式表述

形式如下

$$\begin{aligned}
&\mathrm{Ord}(T)=1,\quad \mathrm{Ord}(F)=0,\\
&[\![T]\!]=\langle\ \{1\},\ \{0\}\ \rangle,\quad [\![F]\!]=\langle\ \{0\},\ \{1\}\ \rangle,\\
&[\![x]\!]=\langle\ \{x\},\ \{x-1\}\ \rangle,\quad [\![\neg x]\!]=\neg[\![x]\!]=\langle\ \{x-1\},\ \{x\}\ \rangle,\\
&[\![\varphi]\!]=\langle\ [\![\varphi]\!]_0,\ [\![\varphi]\!]_1\ \rangle,\quad [\![\neg\varphi]\!]=\neg[\![\varphi]\!]=\langle\ [\![\varphi]\!]_1,\ [\![\varphi]\!]_0\ \rangle,\\
&[\![\varphi\vee\psi]\!]=\langle\ [\![\varphi]\!]_0+[\![\psi]\!]_0,\ [\![\varphi]\!]_1\times[\![\psi]\!]_1\ \rangle,\\
&[\![\varphi\wedge\psi]\!]=\langle\ [\![\varphi]\!]_0\times[\![\psi]\!]_0,\ [\![\varphi]\!]_1+[\![\psi]\!]_1\ \rangle.
\end{aligned}$$

定义 6 给出的是经典布尔逻辑的多项式表述形式. 采用布尔逻辑的这种代数表示, 经典的模型检验问题可以转化为吴方法、 Groebner 基理论等符号计算算法适用的代数问题. 文献 [13–14] 的工作中必须引入的多项式组 $\{x_1^2+x_1, x_2^2+x_2, \cdots, x_n^2+x_n\}$ 在这种表示下是冗余的, 可以省略. 当验证过程中涉及的变元个数很多 (即 n 很大) 时, 这种优化是很明显的. 另外, 柴风娟、高小山等人 [16] 最近提出了关于求解布尔函数的新的特征列方法, 与原有方法相比效率提升非常大, 将他们的方法应用于代数化的布尔模型检验, 能在很大程度上改善验证算法的运行时效率.

3.2　代数化的多值模型检验算法

下面, 我们解释如何将模型 (mv-Kripke 结构) 和待验证的性质 (mv-CTL) 公式转化成多项式表示, 以及从算法上判断待验证的性质是否在该模型下成立.

在符号化的多值模型检验算法中, mv-Kripke 结构 $\mathcal{M}$ 通常由若干个状态变量来描述, $\mathcal{M}$ 的一个状态表示为多个状态变量组成的向量. 代数化的多值模型检验算法中, 我们将 $\mathcal{M}$ 的一个状态视为一组多项式的公共零点集. 例如在 3-QBool 逻辑上, 状态 $\langle\ x_1=T, x_2=M, x_3=F\ \rangle$ 表示为 $\{\ x_1-2, x_2-1, x_3\ \}$. 在这种表示下, 一组多项式可以用来表示一个状态集合. 例如, 多项式组 $\{\ x_1(x_2-1)(x_3-2)\ \}$ 就可以代表所有满足 $x_1=F$ 或者满足 $x_2=M$ 或者满足 $x_1=T$ 的状态.

既然多项式可以用来表示 mv-Kripke 结构的状态, 那么状态之间的迁移关系也可以用多项式来描述. 令 $\mathcal{M}$ 的一条迁移关系为 $(\langle x_1, x_2, \cdots x_n\rangle, \langle x_1', x_2', \cdots, x_n'\rangle)\in R$, 表示从初态 $\langle x_1, x_2, \cdots x_n\rangle$ 到次态 $\langle x_1', x_2', \cdots, x_n'\rangle$ 的迁移. 初态和次态都可以用多项式来表示, 分别记为 F 和 G, 则迁移关系的多项式表示为 $F+G$ (多项式集合的加法见定义 3).

为了将 mv-CTL 公式转化成代数形式的表示, 需要定义拟布尔逻辑的量词 (Quantification Operators) 的多项式表示.

定义 7　令 $[\![f]\!]$ 为多项式表示形式的拟布尔逻辑公式, $x_1, x_2, \cdots, x_n$ 为逻辑变量, 全称量词 $\forall$ 和存在量词 $\exists$ 的多项式表示如下

$$\exists x_n\ f(x_1, x_2, \cdots, x_n)]\!] = \left[\!\!\left[\ \bigvee_{l\in\mathcal{L}} f(x_1, x_2, \cdots, x_n)|_{x_n\leftarrow l}\right]\!\!\right],$$

$$\forall x_n\ f(x_1, x_2, \cdots, x_n)]\!] = \left[\!\!\left[\ \bigwedge_{l\in\mathcal{L}} f(x_1, x_2, \cdots, x_n)|_{x_n\leftarrow l}\right]\!\!\right].$$

通过这样的转化，各种类型的 mv-CTL 公式都可以转化成多项式形式的表示．我们首先来看 EX 类型的公式的转化．令 $\overline{x}$ 和 $\overline{x}'$ 分别为代表初态和次态的状态向量， $R(\overline{x},\overline{x}')$ 为迁移关系集合，则 mv-CTL 公式 $EX\varphi(\overline{x})$ 的多项式表示为

$$[\![EX\varphi(\overline{x})]\!] = [\![\exists\, \overline{x}'\ (\ R(\overline{x},\overline{x}') \wedge \varphi(\overline{x}')\ )]\!].$$

其它类型的 mv-CTL 公式的表示可以基于 $[\![EX\varphi(\overline{x})]\!]$ 和定理 1 给出相关的不动点运算进行转化．

这样， mv-Kripke 结构和 mv-CTL 公式都表示成了多项式的形式，多值模型检验问题转化为通过吴方法计算系统特征列，并判断初始状态所表示的零点是否在满足待验证性质的零点集合中，从而判断模型是否具有该性质．

4 实例分析

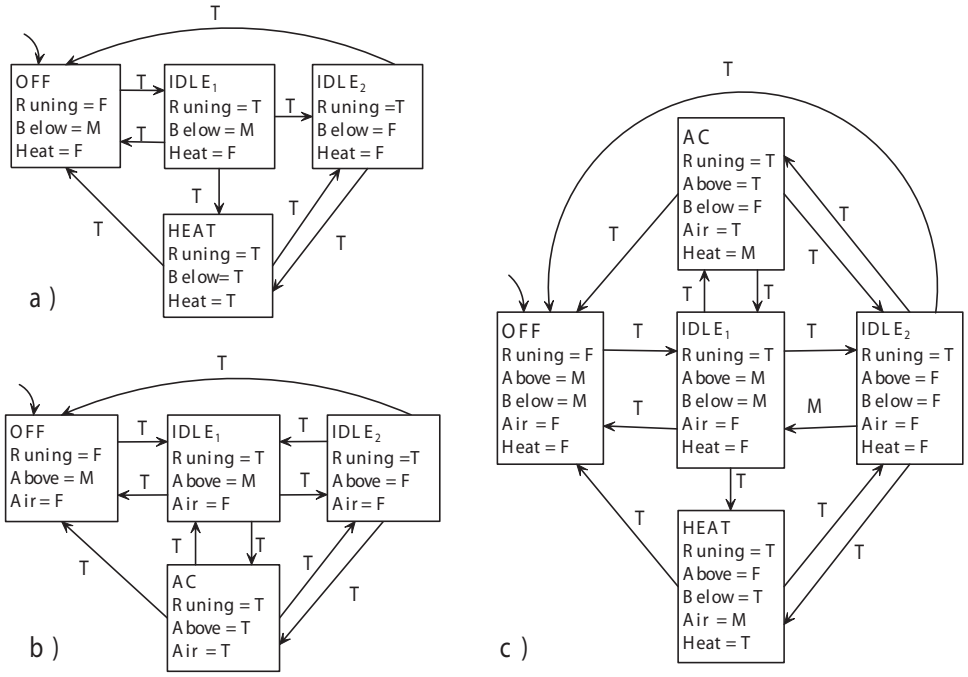

图 3 温控系统 a) 加热系统模型 b) 制冷系统模型 c) 组合模型

本节我们通过一个例子来演示代数化的多值模型检验方法如何验证包含非确定和不一致信息的系统．图 3 c) 给出了一个温控系统的行为描述，它包括一个加热系统图 3 a) 和制冷系统图 3 b). 温控系统包含一个电源开关 (Running), 两个温度传感器 (Below) 和 (Above) 用来感应外界温度，还有两个指示器 (Heat) 和 (Air) 分别指示加热器和制冷器的当前开关状态．系统启动时从状态 OFF 转移到状态 IDLE_1, 等待温度传感器感应外界温度．一旦传感器的读数确定下来，系统状态转移 IDLE_2, AC 或 HEAT. 然而，系统刚刚启动时，传感器的读数是未知的，因此，该系统的模型中包含非确定信息．另外，假设加热系统和制冷系统分

别由不同的开发人员设计，加热系统的设计者认为系统不存在从状态 IDLE_2 到状态 IDLE_1 的转移，而制冷系统的设计者不这么认为，因此，在组合模型还中包含不一致信息.

首先，对状态变量进行编码. 温控系统模型是用三值逻辑表示的，根据定义 5, 逻辑真值 F, M, 和 T 分别与 0,1, 和 2 对应. 用变量 $x_0,\ x_1,\ x_2,\ x_3,\ x_4$ 分别表示状态变量 Running, Above, Below, Air, Heat, 而 $x_0',\ x_1',\ x_2',\ x_3',\ x_4'$ 表示次态变量. 这样，温控系统模型的所有状态都可以用多项式组来表示. 例如，状态 OFF (Running = F, Above = M, Below = M, Air = F, Heat = F) 的多项式表示为 $\{x_0, x_1-1, x_2-1, x_3, x_4\}$. 如果状态 OFF 是某条迁移关系的目的地，那么它应该被表示成 $\{x_0', x_1'-1, x_2'-1, x_3', x_4'\}$.

接下来，将温控系统的迁移关系也表示成多项式的形式. 以从状态 OFF 到状态 IDLE_1 的 T 迁移为例，该迁移 (记为 f_1) 可以表示成 $\{x_0,\ x_1-1,\ x_2-1,\ x_3,\ x_4,\ x_0{}'-2,\ x_1{}'-1,\ x_2{}'-1,\ x_3{}',\ x_4{}'\}$. 类似地，可以得到系统其它的 T 迁移

$$\begin{aligned}
&\mathrm{AC}\rightarrow\mathrm{OFF}\\
&f_2=\{x_0-2,x_1-2,x_2,x_3-2,x_4-1,x_0',x_1'-1,x_2'-1,x_3',x_4'\},\\
&\quad\vdots\\
&\mathrm{HEAT}\rightarrow\mathrm{IDLE}_2\\
&f_{13}=\{x_0-2,x_1,x_2-2,x_3-1,x_4-2,x_0'-2,x_1',x_2',x_3',x_4'\}.
\end{aligned}$$

为了书写的简洁，这里没有显示地给出系统所有的 M 迁移和 F 迁移. 将唯一的 M 迁移记为 f_{14}, 而 F 迁移分别用 $f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20},\ f_{21}, f_{22}, f_{23}, f_{24}, f_{25}$ 表示. 那么，系统整体的迁移关系可以表示为 $[\![R]\!]=\langle\ [\![R]\!]_0,\ [\![R]\!]_1,\ [\![R]\!]_2\ \rangle$, 其中

$$\begin{aligned}
[\![R]\!]_2 &= f_1\times f_2\times\cdots\times f_{12}\times f_{13},\\
[\![R]\!]_1 &= f_{14},\\
[\![R]\!]_0 &= f_{15}\times f_{16}\times\cdots\times f_{24}\times f_{25}.
\end{aligned}$$

既然我们已经将温控系统的模型转化成了多项式形式的表示，那么在此基础上可以开始进行验证. 这里，将要验证的性质为 $E[\neg\mathrm{Below}\ U\ \mathrm{Heat}]$, 也就是 $E[\neg x_2\ U\ x_4]$. 根据定理 1, 验证公式 $E[\neg x_2\ U\ x_4]$ 就是计算 $f(y)=x_4\vee(\neg x_2\wedge EXy)$ 的最小不动点. 模型检验算法第 0 次循环计算 $E^0[\neg x_2\ U\ x_4]$; 第 i 次循环计算 $E^i[\neg x_2\ U\ x_4]$. 算法终止条件为 $E^i[\neg x_2\ U\ x_4]=E^{i+1}[\neg x_2\ U\ x_4]$, 并将 $E^i[\neg x_2\ U\ x_4]$ 作为返回值. 计算的中间结果为

$$\begin{aligned}
E^0[\neg x_2\ U\ x_4]&=\langle\ \{x_4\},\{x_4-1\},\{x_4-2\}\ \rangle,\\
E^1[\neg x_2\ U\ x_4]&=\langle\ \{1\},\{x_0^2-2x_0,x_1-1,x_2-1,x_3,x_4\},\\
&\qquad\{x_0-2,x_1^2-2x_1,x_2x_1,x_2^2-2x_2,2x_3-x_2-2x_1,2x_4-2x_2-x_1\}\ \rangle,\\
E^2[\neg x_2\ U\ x_4]&=\langle\ \{1\},\{x_0^2-2x_0,x_1-1,x_2-1,x_3,x_4\},\\
&\qquad\{x_0-2,x_1^2-2x_1,x_2x_1,x_2^2-2x_2,2x_3-x_2-2x_1,2x_4-2x_2-x_1\}\ \rangle.
\end{aligned}$$

现在，$E^1[\neg x_2\ U\ x_4]=E^2[\neg x_2\ U\ x_4]$. 因此，算法终止，并返回 $E[\neg x_2\ U\ x_4]$ 的多项式表示，即 $\langle\ \{1\},\{x_0^2-2x_0,x_1-1,x_2-1,x_3,x_4\},\{x_0-2,x_1^2-2x_1,x_2x_1,x_2^2-2x_2,2x_3-$

$x_2 - 2x_1, 2x_4 - 2x_2 - x_1\}$ $\rangle$. 对系统的初始状态 OFF 进行零点分解， WSolve(OFF) = $\{$ $[x_0, x_1 - 1, x_2 - 1, x_3, x_4]$ $\}$(函数 WSolve(OFF) 通过一组有限多项式集合的特征列对其进行零点分解). 显然，WSolve(OFF) $\subseteq$ WSolve($[\![E[\neg x_2\ U\ x_4]]\!]_1$) = $\{$ $[x_0, x_1-1, x_2-1, x_3, x_4]$, $[x_0-2, x_1 - 1, x_2 - 1, x_3, x_4]$ $\}$. 因此，对于温控系统模型，性质 $E[\neg x_2\ U\ x_4]$ 的取值既不是 “真 (T)”, 也不是 “假 (F)”, 而是 “可能 (M)”. 也就是说，系统的该性质是与其包含的非确定和不一致的信息密切相关的.

上面的计算是在 Maple 10 中进行的，用来验证上述性质的 Maple 程序在 PC(1.9 GHz Pentium 4 和 256 MB 内存) 上的运行时间是 15 秒，消耗内存 4 MB. 然而， Maple 是用于一般符号计算的软件，其数据结构和算法没有针对模型检验问题进行优化，因此，其运行结果并不能准确地对我们的方法进行评估. 针对模型检验实例的特点，设计专门的数据结构和优化策略，进行大规模的实例分析将是我们下一阶段的研究重点之一.

5 结论和展望

本文提出了一种系统化的多值逻辑 (涵盖经典逻辑) 的多项式表示方法，该方法可以根据给定的逻辑公式自动地构造其对应的多项式表示形式. 把系统的状态、转移关系以及要验证的性质都转化成多项式，而不是逻辑函数，然后在多项式代数系统中研究和开发验证算法. 使用吴方法的基本思想和框架实现复杂系统形式验证中基于多值逻辑的模型检验的代数化，建立了通过吴方法实现多值模型检验技术的整体框架.

下一阶段我们的研究方向将集中在以下几个方面. 1) 寻找自动而有效的对系统描述进行编码的方法. 如何使得转化后的代数形式简洁并保持等价，这是提高基于符号计算的模型检验算法运算效率的关键； 2) 吴方法针对一般的多项式算法效率不高，因此我们下一步工作的重点是研究模型检验下多项式的特点，重新设计吴方法在特定多项式环上的算法，提高算法效率. 高小山等人 [16] 的工作给了我们很大的启示，他们利用有限域 $\mathbb{F}_2$ 的特点给出的布尔环上的特征列方法相对原有方法效率提升非常明显； 3) 自动分析转化后的代数系统的域及理想等特点，根据前面研究的结果，采用启发式方法以达到简化问题的目的.

参 考 文 献

[1] Clarke E, Grumberg O and Peled D. Model Checking. Cambridge, MA: The MIT Press, 1999.
[2] Sagiv M, Reps T and Wilhelm R. Parametric shape analysis via 3-valued logic. Proceedings of Symposium on Principles of Programming Languages, 1999, 105–118.
[3] Hazelhurst S. Generating and Model Checking a Hierarchy of Abstract Models. Technical Report TR-Wits- CS-1999-0, 1999.
[4] Chechik M, Easterbrook S and Petrovykh V. Model-checking over multi-valued logics. Proceedings of Formal Methods for Increasing Software Productivity International Symposium of Formal Methods Europe, 2001, 72–98.
[5] Bolc L and Borowik P. Many-Valued Logics. Springer Verlag, 1992.

[6] Chechik M, Devereux B, Easterbrook S and Gurfinkel A. Multi-Valued Symbolic Model-Checking. ACM Transactions on Software Engineering and Methodology, 2003, 371–408.
[7] Chechik M, Devereux B and Easterbrook S. Implementing a Multi-Valued Symbolic Model Checker. Proceedings of Tools and Algorithms for the Construction and Analysis of Systems, 2001, 404–419.
[8] Srinivasan A, Kam T and Brayton R. Algorithms for Discrete Function Manipulation. IEEE/ACM International Conference on Computer-Aided Design, 1990, 92–95.
[9] Chechik M, Devereux B and Gurfinkel A. χChek: A Multi-Valued Model-Checker. Proceedings of 14th International Conference on Computer-Aided Verification, 2002, 505–509.
[10] Bryant R E. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 1992, **24**(3): 293–318.
[11] Wu W T. Basic principles of mechanical theorem proving in geometries. *J. of Automated Reasoning*, 1986, **2**(4): 221–252.
[12] Ritt J F. Differential Algebra. Amer. Math. Sco., 1950.
[13] Avrunin G S. Symbolic model checking using algebraic geometry. *Lecture Notes in Computer Science*, 1996, **1102**: 26–37.
[14] Mao W B and Wu J Z. Application of Wu's method to symbolic model checking. Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation, 2005, 237–244.
[15] Tarski A. A Lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 1955, **5**: 285–309.
[16] Gao X S, Chai F J and Yuan C. A characteristic set method for equation solving in $\mathbb{F}_2$ and applications in cryptanalysis of stream ciphers. *The Mathematics-Mechanization Research Preprints*, 2006, **25**: 42–56.

APPLICATION OF WU'S METHOD TO MULTI-VALUED MODEL CHECKING

ZHAO Lin

(*Chengdu Institute of Computer Applications, CAS, Chengdu* 610041; *Beijing Jiaotong University State Key Laboratory of Rail Traffic Control and Safety, Beijing* 100044)

WU Jinzhao

(*Chengdu Institute of Computer Applications, CAS, Chengdu* 610041)

Abstract The development of most large and complex systems is necessarily involved with the treatment of uncertainty and inconsistency. Multi-valued model checking is very useful for analyzing models that contain such information. Based on the algebraic representation of multi-valued logics, we present a framework to apply Wu's method to multi-valued model checking. This algebraic approach to multi-valued model checking can be used as a powerful supplement to the existing methods.

Key words Multi-valued model checking, quasi-boolean logics, polynomials, Wu's method.