基于特征切片的软件产品线模型检测

刘玉梅 魏 欧 黄鸣宇
(南京航空航天大学计算机科学与技术学院 南京 210016)

摘 要 特征模型是一种描述软件产品线中共性和可变性特征的通用形式。特征模型象征着所有可能的应用程序配置空间,是实现个性化产品定制的基础。随着软件产品线的规模和复杂程度的增加,如何有效支持以用户需求为基础以及根据特定需求和利益相关者的目标进行个性定制开发是亟待解决的实际问题。提出一种根据用户需求对特征模型进行切片,进一步结合三值逻辑对行为模型进行抽象,最后利用模型检测技术对软件产品线进行验证的方法。实验结果证实了该方法的有效性。

关键词 软件产品线,特征切片,三值模型,模型检测
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.4.034

Model Checking Software Product Line Based on Feature Slicing

LIU Yu-mei WEI Ou HUANG Ming-yu
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract Feature models are a popular formalism for describing the commonality and variability of software product line in terms of features. Feature models symbolize a representation of the possible application configuration space, and can be customized based on specific domain requirements and stakeholder goals. As feature models are becoming increasingly complex, it is desired to provide automatic support for customized analysis and verification based on the strategic goals and requirements of stakeholders. This paper firstly presented feature model slicing based on the requirements of the users. It then introduced three-valued abstraction of behavior models based on the slicing unit. Finally, based on multi-valued model checker, a case study was conducted to illustrate the effectiveness of our approach.

Keywords Software product line, Feature slicing, Three-valued model, Model checking

1 引言

软件产品线(Software Product Line, SPL)是在公共核心资源的基础上,按照规定方式开发的软件密集系统的集合。这些系统共享一组公共的、可管理的、能够满足特定市场或者任务需求的功能集合[1]。

自特征[2]的概念被引入到软件产品线后,基于特征的领域分析与建模技术在实际中得到广泛应用。在软件产品线中,基于特征的建模方法[3]主要包括两部分:特征建模(Feature Modeling)和行为建模(Behavioral Modeling)。特征建模主要侧重于定义软件产品线中特征之间的层次关系。例如,特征模型[4](Feature Model, FM)是一种描述产品线中共性和可变性特征以及特征之间关系的模型。软件产品线的行为建模主要侧重于描述产品线上所有产品根据不同特征所能发生的行为。例如,特征迁移系统[5](Featured Transition System, FTS)是一个以特征为基本单元来描述整个系统行为的形式化模型。

随着软件产品线广泛应用于航天、汽车以及航空电子等安全关键系统,保证软件产品线的正确性的要求越来越重要。模型检测[6]是一种自动形式化验证技术,用于对计算机系统的正确行为属性进行判断。软件产品线模型检测的基本方法是用一个特征模型和特征迁移系统来表示所要检测的系统的模型,并用时序逻辑(如CTL、LTL)公式 φ 来描述系统的正确行为属性,然后通过对模型状态空间进行穷举搜索来判断该公式是否能够在产品线模型上被满足。如果公式在模型上被满足,则系统的正确性得到证实,也就是说,每个产品均满足该公式所对应的属性;否则,就表明系统中存在错误,系统正确性被证伪,即存在着不满足给定属性的软件产品。

然而,在实际应用中,很多因素制约着软件产品线的自动化分析和验证。其一,软件产品线的特征模型变得越来越复杂[7],例如特征数量不断增加使家族产品呈指数级增长,同时,用户需求、技术、环境等因素使特征间的约束关系更加复杂;其二,越来越多的软件部件由外部软件提供商提供[8,9],因此支持为不同利益相关者提供的产品给系统开发人员带了

到稿日期:2015-03-05 返修日期:2015-06-17 本文受国家自然科学基金项目(61170043),国家重点基础研究发展计划(973)项目(2014CB744904)资助。

刘玉梅(1990-),女,硕士生,CCF会员,主要研究领域为模型检测、软件产品线,E-mail:lymworks@126.com;魏 欧(1974-),男,博士,副教授,主要研究领域为形式化方法、软件自动验证,E-mail:owei@nuaa.edu.cn;黄鸣宇(1994-),男,硕士生,主要研究领域为软件验证,E-mail:hmy189@163.com。

很大的困难；其三，庞大的特征模型带来了状态空间爆炸问题，这也是软件产品线模型检测应用的主要瓶颈。

针对以上问题，本文提出基于特征切片的软件产品线模型检测方法。该方法根据不同用户需求对特征模型进行分解切片，利用三值逻辑对特征迁移系统进行抽象，通过多值模型检测技术对软件产品线进行正确性验证。具体方法如下：首先根据用户需求定义切片准则，利用特征模型的语义抽取出与切片准则相关的最小依赖特征集和最小排斥特征集。然后，利用三值逻辑对特征迁移系统的迁移关系进行选择。其中，将属于依赖集中的特征行为属性标记为真(true)，表示依赖该特征标记的迁移；将属于排斥集中的特征行为属性标记为假(false)，表示排斥该特征标记的迁移；将既不属于依赖集也不属于排斥集中的特征行为属性标记为未知(maybe)，即对该特征标记的迁移选择无法确定。进一步，按照三值逻辑对迁移关系上的特征表达式进行逻辑运算，得到一个抽象的三值特征迁移系统。最后，利用多值模型检测工具 χchek[10] 对软件产品线进行模型检测，给出系统行为属性的满足程度。本文提出的基于特征切片的软件产品线模型检测方法不仅仅是一个对特征模型进行自动分析的方法[11]，从产品配置角度来看，该方法提供了一个满足用户特定需求的基础单元，为后续的可变性分析验证奠定了基础。与此同时，利用三值逻辑对行为模型进行抽象从一定程度上缓解了模型检测在空间和运行时间上的瓶颈问题。

本文第 2 节介绍涉及到的理论知识；第 3 节描述产品线特征模型切片的思想与实现；第 4 节介绍对产品线特征迁移系统的抽象；第 5 节利用多值模型检测工具进行验证；第 6 节讨论相关工作；最后进行总结和展望。

2 基本概念

本节主要介绍产品线特征模型、特征迁移系统以及计算树逻辑等预备知识。

2.1 特征模型

特征模型是领域工程中描述软件产品线共性和可变性特征以及特征之间关系的需求模型。特征模型一般由树形结构图表示，称为特征图(Feature Diagram，FD)[12]。特征图中有且仅有一个根节点——表示一个系统。特征则由树形结构中的一般节点表示，并通过特征名称来识别。边表示父特征与子特征间的约束分解关系。

定义 1[7] 特征图(Feature Diagram)是一个六元组：$FD=(G,r,E_{mand},E_{opt},F_{xor},F_{or})$，其中

1) $G=(F,E)$ 为树，其中 F 是软件产品线中的有限特征集，$E\subseteq F\times F$ 是有限边集；

2) $r\in F$ 是唯一的根节点；

3) $E_{mand}\subseteq E$ 是父特征与子特征间存在的必选特征分解模式的边集；

4) $E_{opt}\subseteq E$ 是父特征与子特征间存在的可选特征分解模式的边集；

5) $F_{xor}\subseteq P(F)\times F$ 是子特征与父特征间满足多选一分解模式的特征组；

6) $F_{or}\subseteq P(F)\times F$ 是子特征与父特征间满足多选多分解模式的特征组。

定义 2 软件产品线中某个特定产品是指满足约束条件的特征集；一个特征模型的产品集指所有产品的集合。

对于任一个特征模型 d，本文采用符号 $|d|$ 表示产品集。

图 1 描述的是挡风玻璃雨刷控制器 WWD(Windscreen Wiper Controller)特征图[5]。该控制器包含 5 个特征 Wiper-Family、Sensor、Permanent、Low 和 High，每个特征都用一个小写字母表示。其中根节点特征 w 表示雨刷系统。特征 s 表示雨量传感器；根据质量高低，该传感器分为高品质 h 和低品质 l 两种，其中高品质传感器 h 可以识别雨量的大小，而 l 不可以。特征 p 表示一个永久雨刷。根据定义 1，p 是可选特征，即该控制器系统中的产品可以选择是否包含该特征。l 和 h 之间是多选一的分解模式，即当父特征 s 出现在某产品中时，l 和 h 之间有且仅有一个出现在该产品中。根据定义 2，可以得到此特征模型对应的所有满足约束条件的有效产品配置集为 $|WWD|=\{\{w,s,l\},\{w,s,h\},\{w,s,l,p\},\{w,s,h,p\}\}$。

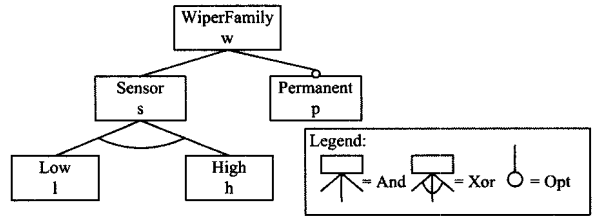

图 1 雨刷控制器特征模型 WWD

2.2 特征迁移系统

一般来说，迁移系统是一个有向图，用于描述计算机系统的行为模型，其中迁移关系用动作标记，状态用原子命题标记。面向软件产品线的特征迁移系统(FTS)是一种扩展的迁移系统[6]，其以特征为基本单元描述整个软件产品线中所有产品的系统行为。在 FTS 中，迁移关系以“动作/特征表达式”的形式标记。例如 $s\xrightarrow{up/a\wedge\neg b}t$ 表示从状态 s 出发，做 up 这个动作，到达状态 t，同时该迁移依赖于特征 a 且排斥特征 b。特征迁移系统的具体定义如下：

定义 3 特征迁移系统(FTS)是一个六元组：$FTS=(S,Act,I,trans,AP,L,d,\gamma)$，其中

1) S 是有穷状态集；

2) Act 是动作集；

3) $I\subseteq S$ 是初始状态集；

4) $trans\subseteq S\times Act\times S$ 是有穷迁移集；

5) AP 表示原子命题集；

6) $L:S\to P(AP)$ 表示用原子命题标记每一个状态；

7) d 是特征模型；

8) $\gamma:trans\to(\{0,1\}^{|N|}\to\{0,1\})$ 是以特征为变量的布尔函数，用特征表达式标记迁移关系。

图 2 所描述的是雨刷控制器特征迁移系统，该系统的基本工作流程是：拉起控制杆启动系统→激活传感器→传感器根据雨量大小调整雨刷速度。

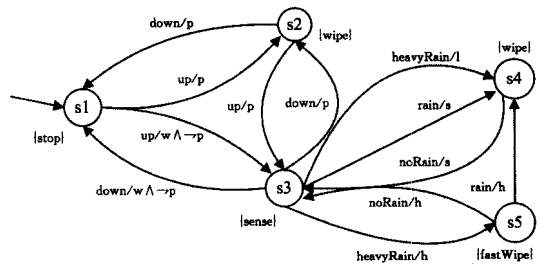

图 2 雨刷控制器产品线的 FTS

图中初始状态为 $s1$，拉起控制杆(动作 up)启动系统。若产品中包含特征 p，系统首先到达状态 $s2$，再一次拉起控制杆，系统到达状态 $s3$，激活传感器；另一方面，若产品中不包含特征 p，则从初始状态 $s1$ 拉起控制杆后，系统将直接到达状态 $s3$，激活传感器。在状态 $s3$ 下，若产品包含低品质传感器，即特征 l，则无论大雨(heavyRain)还是小雨(rain)，系统都到达状态 $s4$，使得控制器以相同的速度控制雨刷；而高品质传感器即特征 h 能够区分 heavyRain 和 rain，若雨量为 heavyRain，系统到达状态 $s5$；若为 rain，则到达状态 $s4$；若雨量由 heavyRain 变为 rain，则系统状态从 $s5$ 迁移到 $s4$。当雨停止时，通过拉下控制杆(动作 down)使得系统回到初始状态。

2.3 ACTL 逻辑

基于特征迁移系统对软件产品线进行模型检测，需要选择合适的时序逻辑描述系统属性。常见的时序逻辑有计算树逻辑(Computation Tree Logic, CTL)[13] 和线性时序逻辑(Linear-time Temporal Logic, LTL)[14]。由于 CTL 和 LTL 都是以 Kripke 结构为计算模型，并不直接适用于对特征迁移系统属性的描述，因此本文采用一种基于动作的 CTL 逻辑-ACTL[15] 描述系统的时序逻辑属性。

定义 4 ACTL 公式由下列规则进行归纳：

$\varphi ::= \text{true} \mid \text{false} \mid p \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi' \mid (\varphi \rightarrow \varphi') \mid AF\varphi \mid EF\varphi \mid EG\varphi \mid AG\varphi \mid A[\varphi U \varphi'] \mid E[\varphi U \varphi'] \mid ([a]\varphi) \mid \langle a\rangle\varphi$

其中，*true*、*false* 为逻辑常量，分别表示“真”、“假”；p 表示原子命题；$\wedge$、$\vee$、$\neg$、$\rightarrow$ 为基本逻辑连接词，分别表示“且”、“或”、“非”、“蕴含”；A 表示“所有路径”；E 表示“存在一条路径”；F 表示“将来某个状态”；G 表示“将来所有状态”；U 表示“直到”；a 动作(action)；$\langle a\rangle$ 表示“存在某条迁移做 a 动作到达下一状态”；$[a]$ 表示“所有做 a 这个动作的迁移到达下一状态”。除了 $\langle\rangle$ 和 $[]$，每个 ACTL 时态连接词都是一对符号。符号对中的第一个是 A 或 E，符号对中的第二个符号是 F、G 或 U。

以图 2 中的雨刷控制器特征迁移系统为例，ACTL 公式 $AG[\text{rain}]AF\text{wipe}$ 描述的属性是“任何时候只要天气下雨，雨刷总会启动工作”。

3 特征模型切片

特征模型是软件产品线应用系统开发定制的基础。在一个包含大量特征并且约束复杂的特征模型基础上，如何帮助系统开发人员根据用户需求对所对应的产品实现验证具有重要的实际意义。为此，基于程序切片的思想，本文提出对特征模型进行软件产品线模型检测，其中切片准则由用户指定的特征集组成；切片结果是满足用户需求的最小配置集，由依赖特征集和排斥特征集组成。

3.1 示例分析

本节以图 1 所示的雨刷控制器特征模型 WWD 来说明切片操作的基本思想。

假设用户选定特征 s，则定义切片准则为 $S1=\{s\}$。根据定义 1 中的约束依赖关系得到包含特征 s 的产品配置集为 $\{w,s,l\}$、$\{w,s,h\}$、$\{w,s,l,p\}$ 和 $\{w,s,h,p\}$。观察发现，用户的特定需求往往对应一组产品变体(尤其是大而复杂的特征模型)，其公共部分往往对应用户的硬性需求，如特征 w 和 s；非公共部分的特征则可以根据用户的软性需求或技术、环境等约束进行后续绑定，如特征 p、l 和 h。通过对公共特征部分及其对应的系统行为进行验证，可为后续依据用户软性需求进行可变性动态绑定奠定基础。基于此，首先将与用户需求对应的产品配置集的公共部分定义为特征模型切片的依赖特征集。

再考虑另一个切片准则 $S2=\{s,h\}$。根据上面的分析得到依赖特征集是 $\{w,s,h\}$。同时，考虑到在特征模型中，特征 l 和 h 是满足多选一的分解模式，而依赖特征集 $\{w,s,h\}$ 只体现特征间的依赖关系，并未体现特征 l 与 h 间隐含的排斥关系。由于特征模型是通过特征间的组合与约束来实现对软件产品线中所有产品的配置，用户特定需求反映了目标产品的部分特征组合。而特征依赖集仅仅表示了相关产品公共部分，整体上的约束还需要通过一个排斥特征集进行描述，为后续实现部分特征集在整个行为模型的抽象提供更精确的分析基础。因此，进一步考虑描述所有与依赖集中特征存在互斥关系的排斥特征集合。综上，定义特征模型切片操作的结果包含依赖特征集和排斥特征集两部分信息。

3.2 语义

根据以上的示例分析，下面从特征模型语义的角度给出具体的操作方法定义。

本文定义的特征模型切片是一个作用在特征模型 d 上的一元操作，记为 $\Pi_{f_{slice}}(d)$，其中 $f_{slice}=\{f_1,f_2,\cdots,f_n\}\subseteq F$ 指用户特定需求对应的特征集，称为切片准则。

定义 5(特征模型切片) 给定特征模型 d 与切片准则 f_{slice}，切片结果集 $\Pi_{f_{slice}}(d)=\langle\phi_{impl},\phi_{excl}\rangle$ 定义如下：

1) $\phi_{impl}=\{\bigcap x \mid x\in |f_{slice}|\}$

2) $\phi_{excl}=\{\bigcap (F\backslash y) \mid y\in |f_{slice}|\}$

其中 $|f_{slice}|=\{z \mid z\in |d| \wedge f_{slice}\subseteq z\}$ 指所有包含切片准则的产品集，F 指软件产品线中的有限特征集，ϕ_{impl} 为特依赖征集，ϕ_{excl} 为排斥特征集。

如图 1 中的特征模型 WWD，对于切片准则 $f_{slice}=\{s,l\}$，包含 f_{slice} 并且满足特征约束的产品集是：

$|f_{slice}|=\{\{w,s,l\},\{w,s,l,p\}\}$

根据定义 5，对以上的产品集进行运算，得到特征依赖集为 $\{w,s,l\}$，特征排斥集为 $\{h\}$，即 $\Pi_l(fm)=\langle\{w,s,l\},\{h\}\rangle$。

特征切片的结果反映了与用户特定需求最紧密相关的特征集合，基于此可进一步对产品线的行为模型进行抽象。

4 特征迁移系统抽象

软件产品线的系统行为模型采用特征迁移系统来描述，本节根据特征切片结果通过三值逻辑对特征迁移系统进行抽象。

根据特征模型的切片结果对特征迁移系统进行抽象的一种直接的想法是对特征迁移系统进行投影操作，即将上节得到的切片结果集作为行为模型的切片准则，遍历搜索特征迁移系统，通过对迁移关系的删除和保留操作，得到一个仅依赖于切片结果集的子特征迁移系统。例如，考虑特征切片结果集 $P1=\langle\{w,s,l\},\{h\}\rangle$，根据上述想法将其作为行为模型的切片准则，对图 2 所示的特征迁移系统进行操作：注意到从状态 $s1$ 出发，到达状态 $s2$ 依赖于特征 p，而到达状态 $s3$ 排斥特征 p；但切片准则 $P1$ 中既没有依赖也没有排斥特征 p，即对特征 p 的选择未知。如果删除这些迁移关系，得到的子模型将是一个不连通的系统模型，这样的切片对后续开发没有意

义；然而，如果保留这些迁移关系，则会强制这些迁移关系必须发生，与所期望的语义不符。

另一方面，考虑以 $P2=\langle\{w,s,p\},\emptyset\rangle$ 为切片准则对特征迁移系统进行上述投影操作，将得到图 3 对应的子模型。观察发现，该模型不满足系统行为属性 $EF(fastWipe)$，但这并不能说明所有包含 $\{w,s,p\}$ 单元的产品都不满足该属性，例如，包含 $\{w,s,p,h\}$ 的产品就满足该属性。同样，如果子模型满足某个属性，也不能保证整个产品线系统中的每个产品都满足该属性。因此，对采用投影操作得到的子系统模型的验证结果无法帮助判断对整个产品线系统模型的验证结果。

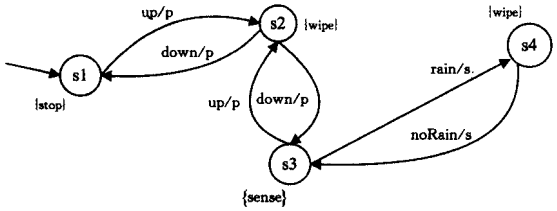

图 3　子特征迁移系统

根据以上的分析，为了能够从对子模型的验证结果中推导出对整个产品线系统的验证结果，与特征切片结果对应的子模型应该是基于产品线模型的抽象，而不是简单的投影操作。为此，考虑采用三值逻辑[16]来描述与特征切片对应的子特征迁移系统，将三值逻辑扩展应用到对未选特征的建模，使用 maybe 表示既未出现在依赖特征集又未出现在排斥特征集中的特征行为属性。相应地，true 表示出现在依赖特征集中的特征行为属性，false 表示出现在排斥特征集中的特征行为属性。具体定义如下：

定义 6（三值特征迁移系统）　设一个产品线的特征迁移系统为 $M_1=(S,Act,I,trans,AP,L,d,\gamma)$，给定切片结果集 $P=\langle\phi_{impl},\phi_{excl}\rangle$，则由 P 在 M_1 上抽象得到的三值特征迁移系统 $M_2=(S,Act,I,trans,AP,L,B,\gamma')$ 定义如下：

1）S、Act、I、$trans$、AP、L 同定义 3；

2）$$B(f)=\begin{cases}T, & f\in\phi_{impl}\\ M, & \text{else}\\ F, & f\in\phi_{excl}\end{cases}$$

其中，B 表示一个赋值函数，即将依赖特征集中特征行为属性标记为 T(true)，排斥特征集中的特征行为属性标记为 F(false)，其他特征行为属性标记为 M(maybe)；

3）γ'：$trans\rightarrow(\{T,M,F\}^{|N|}\rightarrow\{T,M,F\})$ 是一个全函数，用三值逻辑标记迁移关系。

根据定义 6，利用第 3 节的切片结果 $P1=\langle\{w,s,l\},\{h\}\rangle$ 以及 $P2=\langle\{w,s,p\},\emptyset\rangle$ 对图 2 的特征迁移系统进行抽象，分别得到图 4 和图 5 对应的三值特征迁移系统。

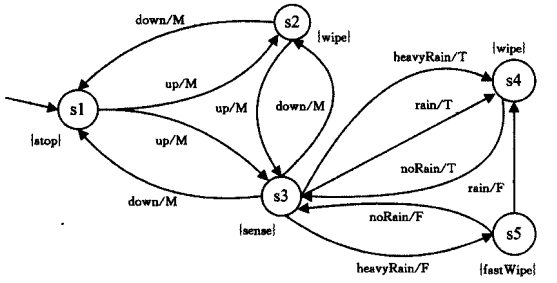

图 4　基于 $P1=\langle\{w,s,l\},\{h\}\rangle$ 对特征迁移系统的抽象

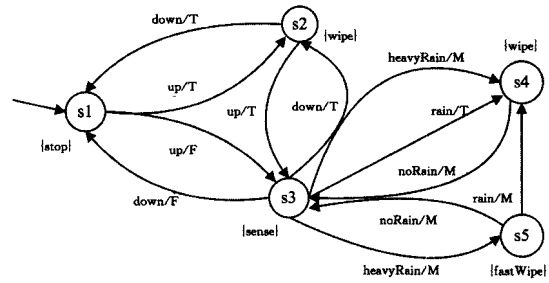

图 5　基于 $P2=\langle\{w,s,p\},\emptyset\rangle$ 对特征迁移系统的抽象

同时，将基于三值逻辑的模型检测用于不完全的状态模型以进行验证分析：在模型检测返回结果中，若属性满足则返回真(true)，不满足则返回假(false)，若因信息不完全而无法判断属性满足与否则返回未知(maybe)。对于特征迁移系统上的迁移，产品线的行为模型包含的迁移是所有特征依赖的迁移的总和。而基于三值逻辑的特征迁移系统，迁移是特征依赖的迁移、特征排斥的迁移以及特征未知迁移的总和。因此，对于某个属性，如果在特征切片对应的三值逻辑特征迁移系统上的模型检测为真，则所有包含该特征切片中特征的产品都满足该属性；如果模型检测结果为假，则所有包含该特征切片中特征的产品都不满足该属性；如果模型检测结果为未知，则表示包含该特征切片中特征的产品并非都满足或者都不满足该属性，进一步的验证将依赖于对其他特征的选择。

5　实验与分析

本节的主要目的：(1)验证本文提出的基于特征切片的软件产品线模型检测方法的可行性；(2)从验证时间方面，将本文提出的方法与非切片方法进行比较。

5.1　实验环境

本文运用 χchek 对第 4 节中雨刷控制器产品线的两个抽象模型(见图 4 和图 5)进行初步模型检测，分别验证以下 3 个属性。

(#1)AG[rain]AFwipe：该属性描述雨刷控制系统最基本的功能，即任意时刻，只要天气下雨，控制器总会启动雨刷；

(#2)EF[up]AXsense：表示只要司机做 up 这个动作，系统下一步将激活传感器；

(#3)EF(fastWipe)：表示系统永远不会启动快速雨刷。

验证属性及分析结果如表 1 所列。

表 1　实验设计与结果

ACTL 公式	抽象模型	
	图 4	图 5
#1	T	T
#2	M	F
#3	F	M

5.2　实验结果及分析

以图 5 对应的抽象模型为例，采用 XML 对其构建Kripke结构模型，运用 χchek 对表 1 中 3 个属性进行验证，运行界面如图 6 所示。由于 χchek 中验证的属性采用 CTL 公式描述，在 FTS 模型检测中用以描述系统属性的 ACTL 公式需要转换成与其语义等价的 CTL 公式。公式 AG[rain]AFwipe 在语义上与 CTL 公式 AG(rain → AFwipe) 等价；公式 EF[up]AXsense在语义上与 CTL 公式 EF(up∧AXsense)等价；公式 EF(fastWipe) 本身是一个 CTL 公式。属性 AG

[rain]AFwipe 验证结果为 true，说明在对应的子模型上该属性成立，因此任意包含相应特征切片中特征的软件产品均满足该属性；属性 EF(up $\wedge$ AXsense)验证结果为 false，说明在对应的子模型上该属性不成立，因此任意包含相应特征切片中特征的软件产品均不满足该属性；属性 EF(fastWipe)验证结果为 maybe，说明在对应的子模型上该属性成立与否依赖于其他特征信息。

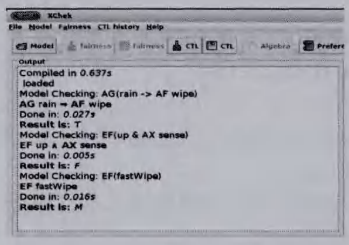

图 6　基于 $P2=\langle\{w,s,p\},\emptyset\rangle$抽象模型的验证结果

5.3　时间性能比较

为评估本文方法与非切片模型检测方法的时间性能，本文对雨刷控制器产品线进行非切片的模型检测。表 2 给出了两种方法对相同的属性进行验证的时间消耗。

表 2　时间性能比较(单位：s)

方法	编译时间	属性验证时间			综合
		#1	#2	#3	
切片方法	0.65	0.028	0.007	0.009	0.694
非切片方法	0.799	0.036	0.01	0.022	0.867

由图 6 和表 2 可以看出，基于特征切片的软件产品线模型检测方法是可行的，同时该方法在时间性能上相对于非切片方法提高了约 20%。

6　相关工作

基于特征模型的模型检测技术是软件产品线领域的研究热点。文献[11]总结了近 20 年以来基于特征模型的操作、技术、工具和经验等方面的研究成果。文献[7]提出特征模型切片操作，该操作以任意一组特征集作为切片准则，根据特征模型语义和命题逻辑得到一个子特征模型。该模型是一个仅包含切片准则中特征并同时保留原模型中特征间约束关系的树形结构。该切片操作已经实现并集成到 FAMILIAR 中，对特征模型大而复杂的问题提供了自动化分析方法。然而，该方法并没有将对特征模型的分析结果与系统功能行为相关联。本文通过对特征模型进行分解切片，抽取出包含切片准则的最小依赖特征集和最小排斥特征集，并将切片结果作用在系统行为模型上，实现对部分特征的行为属性的验证。文献[17]采用行为与特征相结合的可变性建模方式，提出 FTS，并开发了相应的验证工具 SNIP。但该方法并没有对部分特征的行为属性实现模型检测。本文在其提出的特征迁移系统的基础上，实现在整个产品线系统行为模型上对部分特征的行为属性进行抽象验证。多值模型是传统布尔模型的扩展，与布尔模型相比，多值模型更适合对包含不确定和不一致信息的软件系统进行建模[18]。石玉峰等人[19]将 FTS 扩展到多值逻辑上，提出基于 BFTS 多值模型对软件产品线进行建模的方法，进而将特征与行为的关系从二值变为多值，解决了对软件产品线中出现的信息不确定和不一致情况的描述，但是，该方法没有针对用户的特定需求进行特征模型的分析和产品线验证。本文从系统开发定制的角度，以用户的特定需求为导向，将三值逻辑扩展到对特征行为的选择，实现对软件产品线系统行为模型的多值抽象和验证。

结束语　特征模型是软件产品线应用系统开发定制的基础。为了有效地处理特征模型复杂性给系统开发人员带来的分析以及个性定制开发等方面的问题，本文从需求定制的角度出发，对特征模型进行分解切片，帮助系统开发人员根据用户需求个性定制开发单元。同时应用多值逻辑对特征迁移系统进行抽象，通过模型检测技术对系统进行验证，实验结果证实了该方法的有效性。到目前为止，已经实现了对特征模型的切片并将其集成到 TVL[20] 工具中，但对特征迁移系统的自动抽象还有待实现。在未来的工作中，将继续研究自动化工具以及算法的设计与实现，采用更多、更复杂的实例对方法进行改进和完善。

参 考 文 献

[1] Pohl K, Bockle G, Linden V D. Software Product Line Engineering: Foundations, Principles and Techniques[M]. Berlin Heidelberg: Springer-Verlag, 2005

[2] Classen A, Heymans P, Schobbens P Y. What's in a feature: A requirements engineering perspective[C]// Proceedings of the 11th International Conference on Fundamental Approaches to Software Engineering (FASE 08) in Conjuncation with ETAPS' 08. 2008: 16-30

[3] Nie K M, Zhang L, Fan Z Q. Systematic literarture review of software product line variability modeling techniques[J]. Journal of Software, 2013, 24(9): 2001-2019(in Chinese)
聂坤明，张莉，樊志强. 软件产品线可变性建模技术系统综述[J]. 软件学报，2013，24(9)：2001-2019

[4] Schobbens P Y, Heymans P, Trigaux J C, et al. Generic semantics of feature diagrams[J]. Computer Networks, 2007, 51(2): 456-479

[5] Classen A, Heymans P, Schobbens P Y, et al. Symbolic model checking of software product lines[C]// Proceeding of the 33rd International Conference on Software Engineering. New York: ACM, 2011: 321-330

[6] Baier C, Katoen J P. Principles of model checking [M]. Cambridge: MIT press, 2008

[7] Acher M, Collet P, Lahire P, et al. Slicing feature models[C]// Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering. Washington: IEEE, 2011: 424-427

[8] Hartmann H, Trew T. Using feature diagrams with context variability to model multiple product lines for software supply chains[C]// Proceedings of the 12th International Software Product Line Conference. Washington: IEEE, 2008: 12-21

[9] Hartmann H, Trew T, Matsinger A. Supplier independent feature modeling[C]// Proceedings of the 13th International Software Product Line Conference. Pittsburgh: Carnegie Mellon University, 2009: 191-200

[10] Easterbrook S, Chechik M, Devereux B, et al. χChek: A model checker for multi-valued reasoning[C]// Proceedings of the 25th International Conference on Software Engineering. Washington: IEEE Computer Society, 2003: 804-805

[11] Benavides D, Segura S, Ruiz-Cortés A. Automated analysis of feature models 20 years later: A literature review[J]. Information Systems, 2010, 35(6): 615-636

[12] Schobbens P, Heymans P, Trigaux J. Feature Diagrams: A Survey and A Formal Semantics[M]// Proceedings of 14th IEEE International Conference on Requirements Engineering. Washington: IEEE Computer Society, 2006: 139-148

[13] Clarke E M, Emerson E A. Design and synthesis of synchronization skeletons using branching time temporal logic[M]// Logic of Programs: Workshop, Yorktown Heights, New York, May 1981. London: Springer-Verlag, 1981: 52-71

[14] Pnueli A. The temporal logic of programs[C]// Proceedings of the 18th Annual Symposium on Foundations of Computer Science. Washington: IEEE, 1977: 46-57

[15] Nicola R D, Vaandrager F. Action versus state based logics for transition systems[J]// Lecture Notes in Computer Science, 1990, 469: 407-419

[16] Glenn B, Patrice G. Model Checking Partial State Spaces with 3-Valued Temporal Logics[J]. 11th International Conference on Computer Aided Verification(CAV'99). 1999: 274-287

[17] Classen A, Cordy M, Heymans P, et al. Model checking software product lines with SNIP[J]. International Journal on Software Tools for Technology Transfer, 2012, 14(5): 589-612

[18] Chen J J, Wei O. Approximation of multi-valued models via reduction[J]. Computer Science, 2014, 41(6): 125-130(in Chinese)
陈娟娟, 魏欧. 基于分解的多值模型的逼近关系[J]. 计算机科学, 2014, 41(6): 125-130

[19] Shi Y F, Wei O, Zhou Y. Model checking of software product line based on bilattices[J]. Computer Science, 2015, 42(2): 167-172(in Chinese)
石玉峰, 魏欧, 周宇. 基于双格的软件产品线模型检测[J]. 计算机科学, 2015, 42(2): 167-172

[20] Classen A, Boucher Q, Heymans P. A text-based approach to feature modelling: Syntax and semantics of TVL[J]. Science of Computer Programming, 2011, 76(12): 1130-1143

(上接第 162 页)

本文协议与其他几种超轻量认证协议的性能比较结果见表 1。

表 1 超轻量级 RFID 认证协议性能的比较

协议	计算需求	存储空间	通信量
SASI	∧ ∨ ⊕ + Rot	7L	2L
Gorssamer	⊕ + Rot MixBits	7L	2L
CURAP	⊕ Rot Cro	4L	2L
本协议	⊕ CRC Cro	5L	2L

结束语 本文提出了一种低成本超轻量级 RFID 双向认证协议，通过在标签端使用简单的 CRC 运算和交叉位运算实现了标签和阅读器之间的双向认证，并且在降低计算代价的同时，使协议能够有效抵抗拒绝服务攻击、去同步化攻击、假冒攻击等多种恶意攻击，确保在资源受限的低成本标签上完成双向认证。通过 BAN 逻辑形式化分析，验证了本协议的正确性和安全性。通过性能分析说明与已有的超轻量级 RFID 认证协议相比，所提协议减少了标签所需的存储空间。

参 考 文 献

[1] Zhou Y B, Feng D G. Design and Analysis of Cryptographic Protocols for RFID[J]. Chinese Journal of Computers, 2006, 29(4): 581-589(in Chinese)
周永彬, 冯登国. RFID 安全协议的设计与分析[J]. 计算机学报, 2006, 29(4): 581-589

[2] Sun H M, Ting W C. A Gen2-based RFID authentication protocol for security and privacy[J]. IEEE Trans Mob Comput, 2009, 8(8): 1052-1062

[3] Peris-Lopez P, Hernandez-Castro J C, Estevez-Tapiador J M, et al. LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags[C]// Proceedings of the 2nd Workshop on RFID Security. New Jersey, USA: IEEE Press, 2006: 137-148

[4] Chien H Y. SASI: A New Ultra-lightweight RFID Authentication Protocol Providing Strong Authentication and Strong Intergrity[J]. IEEE Trans. Dependable and Secure Computing, 2007, 4(4): 337-340

[5] Bagheri N, Safkhani M, Naderi M, et al. Security Analysis of LMAP++, an RFID Authentication Protocol[C]// Abu Dhabi. 6th International Conference on Internet Technology and Secured Transactions. 2011: 689-694

[6] Phan R C W. Cryptanalysis of a New Ultralightweight RFID Authentication Protocol—SASI[J]. IEEE Trans. on Dependable and Secure Computing, 2009, 6(4): 316-320

[7] Peris-Lopez P, Hernandez-Castro J C, Estevez-Tapiador J M, et al. Advances in Ultra-Lightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol[C]// 9th International Workshop on Information Security Applications. 2009: 56-68

[8] Peng P, Zhao Y M, Han W L, et al. Ultra-lightweight RFID Mutual Authenticaiton Protocol[J]. Computer Engineering, 2011, 37(16): 140-142(in Chinese)
彭朋, 赵一鸣, 韩伟力, 等. 一种超轻量级 RFID 双向认证协议[J]. 计算机工程, 2011, 37(16): 140-142

[9] Du Z Y, Zhang G A, Yuan H L. Crossover Based Ultra-lightweight RFID Authentication Protocol[J]. Computer Science, 2013, 40(11): 35-37(in Chinese)
杜宗印, 章国安, 袁红林. 基于交叉位运算的超轻量 RFID 认证协议[J]. 计算机科学, 2013, 40(11): 35-37

[10] Gao L J, Ma M D, Shu Y T, et al. An ultralightweight RFID authentication protocol with CRC and permutation[J]. Journal of Network and Computer Applications, 2014, 41: 37-46

[11] Yang S P. Analysis and Research of Security Protocol with BAN Logic[D]. Guiyang: Guizhou University, 2007: 54-73(in Chinese)
杨世平. 安全协议及其 BAN 逻辑分析研究[D]. 贵阳: 贵州大学, 2007: 54-73

[12] Tian Y, Chen G L, Li J H. A new ultralightweight RFID authentication protocol with permutation[J]. Communications Letters IEEE, 2012, 16(5): 702-705