**贵州大学研究生中间环节培养系列表**

**学术报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓 名** | 冯仁艳 | | **导 师** | 王以松 | |
| **年 级** | 2018级 | | **院、系** | 计算机科学与技术学院 | |
| **专 业** | 软件工程 | | **研究方向** | 软件工程技术与人工智能 | |
| **学术报告名称** | | 1. Using the Knowledge Expression and Reasonging method to solve the problems in Formal Verification | | | |
| **学术报告时间** | | 2020.09.14 | **学术报告地点** | | 线上会议 |
| **学术报告主要内容(至少2000字)** | | | | | |
| 形式化验证是一种广泛应用在硬件和软件系统中有别于测试的、采用数学方法证明系统满足给定特性的验证（verification）技术。软件和硬件系统的缺陷会导致严重的后果。近年来，为了减少系统（尤其是像火箭发射系统和卫星发射系统等关键领域的系统）错误带来的损失，形式化方法的研究与应用越来越受到人们的重视。INTEL、AMD、IBM、NVIDIA、CADENCE、 Motorola、西门子和微软等大型公司纷纷引入了形式化验证方法。与此同时，学术界也在形式化验证领域取得了突破性的成果，比如：剑桥大学对ARM6处理器进行了验证， 为类似于ARM这样的处理器提供了潜在的形式化验证方法，德国的Verisoft项目验证了一个一万行的操作系统内核。  形式化验证的模型检测首先由Clarke提出，并用于解决并发系统验证问题。Clarke和Emerson指出，在有限状态的并发系统中，使用时态逻辑推演系统（deductive system）中的公理和推理规则进行构造性证明（proof construction）的方法证明系统是否满足给定的规范是不必要的。因为在有限状态并发系统中，并发系统可以被抽象为一个Kripke结构M，规范被表示成一个逻辑公式F；此时，该验证问题就变成检验一个Kripke结构是否满足该公式，即模型检测: 判断M是否是F的一个模型。  Dijkstra在1976年的文章中说：显著提高程序置信度的唯一有效方法是对其正确性给出令人信服的证明。程序的正确性的证明从Dijkstra的数学证明发展到McCarthy、Floyd、Hoare等人的系统的形式化证明。研究表明，由于在真实的软件系统中不仅包括复杂的数据结构还包含有代码之间的上下文关系[18]，这些方法都不能付诸于实践当中。因此，采用自动的方法来验证像反应式系统这类不终止且非确定的复杂系统得到迅速的发展，其中时态逻辑模型检测是一种较为有效的方法。  随着时态逻辑的发展，Allen Emerson和Clarke于1981年结合了解决状态爆炸的方法和时态逻辑，正式将模型检测应用到验证有限状态反应式系统是否满足一个用时态逻辑描述的规范中。也就是将并发系统用一个有限的Kripke结构来表示，而其规范用一个分支时态逻辑公式来表示，然后验证该结构是否为此公式的 | | | | | |
| 模型以验证其正确性。  因此，从KRR的角度探索模型检测下的SNC和WSC的计算是重要的。这不仅提供了计算反应式系统定义在某个符号集上的SNC和WSC的方法，还架起了形式化验证与KRR之间的桥梁。这在理论和实践上都是一项有意义的工作。  本文从遗忘理论的角度出发，拟研究反应式系统的SNC和WSC的计算，从而为计算不终止类系统下的定义在某个符号集上的SNC和WSC提供了新的方法，架起形式化验证与KRR之间的桥梁。特别地，本文将规范的描述语言限制到CTL和μ-演算下。其主要研究内容集中在以下几个方面:  （1）CTL和μ-演算的遗忘理论  本文将研究CTL中遗忘理论的方法和性质，特别是其遗忘结果的存在性、复杂性等等，为探索用遗忘理论计算SNC和WSC提供理论基础。具体说来，遗忘理论具有削弱(Weakening，(W))、正维持（Positive Persistence，(PP)）、负维持（Negative Persistence，(NP)）、无关性（Irrelevance，(IR)）等基本性质。本文将探索CTL和μ-演算的遗忘理论的以上性质，并探讨其与存在性之间的关系。此外，本文将深入研究 CTL和μ-演算 遗忘理论的基本准则、发现计算 CTL和μ-演算遗忘结果的算法以及探讨CTL和μ-演算子类与遗忘相关问题的计算复杂度，为研究计算SNC和WSC的性质、算法以及基本准则等作好铺垫。  计算复杂性理论致力于将可计算问题根据它们本身的复杂性分类。研究表明，在经典命题逻辑中：CNF（Conjunctive normal form）公式的遗忘的推理问题最难是Π\_2^P-完全的，DNF（Disjunctive normal form）公式的遗忘的推理问题是co-NP-完全的；在模态逻辑S5中，遗忘的模型检测问题是NP-完全的，对应的推理问题是Π\_2^P-完全的。基于此，本文拟研究CTL和𝝁-演算的遗忘在模型检测和推理问题上的复杂性。  （2）CTL和μ-演算的遗忘理论的算法的研究及实现  基于上述研究结果，我们拟分别设计并实现一个计算CTL和μ-演算遗忘结果的原型系统，并从实验角度研究其计算代价以启发快速的计算遗忘结果的算法。尽管经典命题逻辑、模态逻辑和一些描述逻辑都有对应的计算遗忘结果的算法和实现，但是CTL中引入了多种时态连接词，μ-演算中引入了不动点操作，如能找到类似于前面所说的那些逻辑中采用的计算方法，将会为CTL和μ-演算的遗忘理论找到牢固的立足点。  CTL不具有均匀插值性质[28]，即：不存在一个算法使得对于任意的CTL公式，其遗忘掉任意原子命题的集合得到的结果仍然是CTL公式。在这种情况下，本文将研究CTL子类的遗忘理论，特别是能保证其遗忘结果仍然是CTL可表达的子类，这些问题非常关键且具有挑战性。  与CTL不同，μ-演算虽然表达能力比CTL强，其可满足性问题也比CTL的复杂，但是μ-演算具有均匀插值性质[51]。这意味着对于在任意的μ-演算句子中遗忘掉任意的原子命题的集合得到的结果仍然是μ-演算公式。此时，探索μ-演算遗忘理论及相关性质显得很重要。 | | | | | |
| （3）结合反应式系统的模型检测中的实例探索SNC和WSC的具体应用。在形式化验证中，SNC和WSC通常用于验证代码段的正确性，也用于负例生成和系统精化。本文拟探讨使用遗忘理论求解反应式系统的SNC和WSC，再结合具体实例探索SNC和WSC在负例生成和系统精化上的应用。在经典命题逻辑和一阶逻辑中，遗忘理论与SNC和WSC的关系分别已经被Lin和Doherty等人分别提出[22,23]。特别地，经典命题逻辑中的SNC和WSC被用于规划问题中的后继状态公理的计算。这里所说的SNC和WSC都是在给定的命题公式或一阶公式下的，当给定一个反应式系统（Kripke structure）时，如何使用上述所说的CTL和μ-演算的遗忘理论计算SNC和WSC是本课题的一个关键问题。 | | | | | |

|  |
| --- |
| **主持学术活动专家评语(参加省级以上会议的由会议组织者提供证明论文进行公开学术交流)**  专家组或会议秘书组签(章)：  (至少3人以上签字方可)  年 月 日 |