南華大學
UNIVERSITY OF SOUTH CHINA

毕业设计(文献综述)

题　　目	模型验证软件的归纳推理算法的设计与实现
学院名称	计算机学院
指导教师	刘杰
职　　称	副教授
班　　级	本 20 软卓 01 班
学　　号	20200440717
学生姓名	李奕星

2023 年 12 月 17 日

摘要：随着软件和硬件系统的日益复杂化，确保这些系统的可靠性和安全性的需求也不断增长。领域特定语言（DSL）例如 L2C，因其能够提供更贴近特定应用领域的抽象而受到青睐。然而，这些语言的特殊性也带来了验证和验证工具可用性的挑战。Lustre 语言，作为一种在安全关键系统领域广泛使用的同步数据流语言，通过其严格的形式化语义提供了一个强大的平台，用于实现这些系统的模型验证。因此，将 L2C 代码转换为 Lustre 代码，不仅有助于提高系统的可靠性，还可以利用 Lustre 的强大验证工具，如 Kind2。本文综述了当前关于 L2C 到 Lustre 转换与验证的研究，特别是归纳推理算法在模型验证中的应用。同时描述了基于 CMake 开发的可在 Windows 环境下运行的 Lustre 验证工具 QKind 的基于 K-induction 算法开发模型检测模块内容编写。

关键词：领域特定语言；L2C；Lustre；归纳推理；模型验证；

1 研究背景

1.1 同步数据流语言研究现状

同步数据流语言（如 Lustre、Signal）已在航空、高铁、核电等安全关键领域得到广泛应用[1]。举例来说，适用于这些领域实时控制系统建模和开发的 Scade 工具是基于类似 Lustre 语言的。这类语言相关的开发工具，尤其是编译器的安全性问题备受关注。近年来，采用形式化验证实现可信编译器的研究成为程序设计语言领域的焦点之一，取得了显著成果，例如 CompCert 项目成功实现了产品级的可信 C 编译器[2]。类似地，人们也采用这种方法开展了同步数据流语言可信编译器的研究工作。

以核领域为例，高性能计算和数据处理一直是至关重要的任务。随着科学计算和工程应用的复杂性不断增加，对于并行计算和数据流处理的需求也日益迫切。传统的并行编程模型往往需要开发人员深入理解硬件架构和并发控制原理，并手动管理并发性和同步，这增加了开发的复杂性和错误率。然而，随着大数据和人工智能等领域的快速发展，数据密集型应用的需求也越来越显著。这些应用对于高效的数据处理和并行计算提出了新的挑战。

而在核领域中，一些应用对实时性有较高的要求，例如控制系统和数据采集系统，这使得更加高效和可靠的编程范式变得至关重要。同步数据流语言的使用为解决这些挑战提供了一种新的思路。这种编程模型将计算视为数据流的传递和

转换过程，使用者只需关注数据流之间的依赖关系，而不需要显式地管理并发和同步。这大大简化了开发过程，降低了出错的概率，并有利于优化并行性和性能。虽然已经有一些同步数据流语言被提出并应用于核领域，但在实际应用中仍然存在一些挑战。例如，如何优化编译器以提高代码的执行效率，如何进行有效的调试和测试以确保程序的正确性等问题仍然需要深入研究。同步数据流语言在核领域的应用具有重要的意义和潜在的挑战。

1.2 模型验证软件现状

随着科技的迅猛进步，复杂软硬件系统在日常生活和关键行业中扮演着愈发重要的角色。这些系统，如交通控制、医疗监护、航空航天及自动化制造等，均承担着至关重要的任务，其可靠性和安全性直接关系到人类生活与财产安全。因此，如何验证和保障这些复杂系统的正确性，已成为研究与工业界共同关注的焦点。

领域特定语言（DSL）因具备为特定领域提供定制化抽象和表达方式的特性，在复杂系统开发中扮演着愈发关键的角色。L2C 作为一种基于 Lustre 语言改良的 DSL，以其简洁高效的语言特性，使得开发者能够迅速实现系统功能，同时确保代码的可读性与可维护性。然而，随着系统复杂度的攀升，单纯依赖人工审核代码以确保系统正确性已变得愈发困难，这凸显了自动化形式化验证工具的必要性。

Lustre[3,4,5]语言，作为一种成熟的同步数据流语言，最早出现在 P.Caspi 的论文中，多用于嵌入式控制系统和信号处理系统[6]。其已在实时和安全关键系统开发中得到广泛应用[7]。其强大的形式化语义为系统模型验证提供了理想平台[8]。可是，当前市面上并没有针对 L2C 语言的解析验证工具，因此开发一款软件来对 L2C 语言进行解析与验证便成了当下的一大需求。

尽管 L2C 到 Lustre 的转换带来了诸多潜在益处，但转换过程本身亦充满挑战，尤其是在确保转换后代码与原始代码在逻辑上的一致性方面。此外，鉴于复杂系统的多样性与实现细节，开发一个通用且高效的转换与验证工具极具挑战性。这要求我们在转换过程中引入更先进的技术与方法，如归纳推理算法，以确保转换的正确性与有效性。

同时，Kind2 软件在运行环境等方面存在一定局限性，为开发者带来诸多不

便。因此，开发一款更加实用的Lustre验证软件以满足当前市场需求显得尤为重要。

2 基本定义及特点

为了深入理解本文综述的主题和范围，首先必须明确几个关键概念的基本定义及其特点。以下是L2C语言、Lustre语言以及归纳推理算法等重点概念的基本定义和特点概述。

2.1 L2C语言

L2C语言，全称Lustre to C Language。是由中国核动力研究设计院研发的同步数据流语言，它允许将Lustre程序转换为C语言代码。它以Lustre语法为基础，通过增添一系列实用的语法功能，进一步丰富了Lustre的表达能力。相较于传统的Lustre语言，L2C语言在现实工作中的适用性得到了显著增强，能够更好地满足复杂系统的建模与仿真需求。

2.2 Lustre语言

Lustre是一种用于并行和分布式计算的声明式编程语言，特别适用于数据流编程模型。它最初由法国国家研究中心INRIA开发，用于实时系统和嵌入式系统的开发。Lustre的主要特点是其清晰、数学化的语法和语义，这使得它成为高可靠性、高可预测性应用的理想选择，特别是在需要确保程序正确性和可维护性的场景中。在Lustre中，程序由数据流方程（equations）组成，这些方程定义了系统的状态随时间的演变。这种声明式的编程范式有助于提高程序的可读性和可维护性，同时也方便进行形式化验证，以确保程序的正确性。

2.3 归纳推理算法

归纳推理算法是一种形式化验证方法，通过对特定案例的验证来推广整体系统的正确性。这种方法尤其适用于证明程序在所有可能的输入和状态下的行为[9,10,11]。其特点有以下3点：

1.普适性： 能够处理各种类型的程序和系统，不限于特定的语言或平台。

2.自动化： 通过自动化的工具和技术，归纳推理可以减少人工干预，提高验证过程的效率和可靠性。

3.适应性： 可以根据不同的需求和条件调整推理策略和深度，具有很高的灵活性。

2.4 Qkind

Qkind 是一款由中国核动力研究设计院委托南华大学计算机学院研发的模型验证软件。其主要设计目标是借助 SMT 求解器，对使用同步数据流语言 L2C 编写的模型进行量词约束安全性属性的证明或反证。在此过程中，这些属性将通过特定的注解语言进行表达，并以不变式或"假设-保证"契约的形式得以体现。为了实现这一目标，Qkind 采用形式化方法分析系统行为，将 L2C 程序和约束属性转化为符号系统，再利用 SMT 求解器进行验证。

2.5 CMake

CMake 是一个跨平台的安装（编译）工具，可以用简单的语句来描述所有平台的安装(编译过程)[12]。他可以完成各种的 makefile 以及 project 文件的输出，并且可以测试编译器所支持的C++特性,就像UNIX下的automake。只是 CMake 的组态档取名为 CMakeLists.txt。同时，Cmake 并不会直接建构出最终的软件，而是会产生标准的建构档（如 Unix 的 Makefile 或 Windows Visual C++ 的 projects/workspaces），然后再根据一般的建构方式使用。这让熟悉某个集成开发环境（IDE）的开发者可以用标准方式建构自己的软件，这种允许使用各平台的原生建构系统的能力是 CMake 和 SCons 等其他类似系统的区别之处。Cmake 在实际开发中有着开放源代码、跨平台、能够管理大型项目、能简化编译构建过程和编译过程、高效率、可扩展等一系列优点[13]。

2.6 ANTLR

ANTLR（全名：ANother Tool for Language Recognition）是基于 LL(*)算法实现的语法解析器生成器（parser generator），用 Java 语言编写，使用自上而下（top-down）的递归下降 LL 剖析器方法。由旧金山大学的 Terence Parr 博士等人于 1989 年开始发展。

2.7 Z3 库

本程序在归纳算法的实现上使用 Z3 库实现。Z3 是一个开源的高性能定理证明器，由微软研究院开发[14]。它提供了一个功能强大的库，用于解决多种形式的自动定理证明问题。该库支持多种语言，包括 C、C++、Python 等，使得它在各种领域的应用变得更加广泛和灵活。Z3 库的核心功能包括布尔逻辑、线性整数和实数算术、位向量和数组理论等。它可以用于解决各种类型的问题，如程序验

证、软件分析、形式化验证等。其强大的求解能力和高效的性能使得它成为了学术界和工业界的研究人员首选的自动定理证明工具之一。除了基本的定理证明功能外，Z3 库还提供了丰富的 API 和文档，以便用户可以轻松地集成和扩展其功能。同时，它还支持各种算法和优化技术，以提高求解效率和准确性。综上，Z3 库是一个功能强大、灵活易用的自动定理证明工具，为研究人员提供了一个强大的工具，用于解决各种复杂的逻辑和数学问题，同时也适合用来完成归纳推理算法的代码实现工作。

2.8 Kind2

Kind2 是一款开源的模型验证工具，专注于连续和混合系统的验证[15]。其独特之处在于，它集成了多种验证技术，如抽象解释、SMT 求解器以及模型检测等。这些技术的融合使得 Kind2 在处理复杂系统模型时具有更高的灵活性和适应性。同时，得益于优化的算法和数据结构，Kind2 能够应对大规模和复杂的系统模型，展现出卓越的验证效率。

2.9 模型验证

模型检测技术[16,17]是一种自动判断一个程序是否满足其规范的方法[18]。模型检测的经典范式包括建模、规范和算法[19],其中建模是在保留实际系统特点的情况下对其进行抽象,而规范则是以时态逻辑对待验证性质进行的刻画,算法则用于对给定性质和模型进行自动化的判断，并且在发现违反规定性质的反例后可以将其输出，对纠正软件设计与发现软件缺陷上能够起到很好的参考作用[20]。同时模型检测技术作为自动化验证技术的核心环节，致力于确保系统或模型与既定规范标准的一致性[21,22]。在软件工程、硬件设计、通信协议等关键领域，模型检测技术的运用至关重要，为系统正确性与可靠性的提升提供了坚实支撑。该技术通过全面且系统的分析，深入探索系统模型的各个层面，自动揭示潜在错误或违规行为。这一过程不仅帮助开发者在设计与实施阶段及时发现并修正问题，还显著降低了软件或系统运行时的错误和故障率，极大地增强了系统的稳定性与可靠性。此外，模型检测技术的价值不容忽视，对于保障系统质量与提升用户体验具有关键性作用。随着技术的持续进步与完善，我们有充分理由期待，模型检测将在软件工程、硬件设计等未来领域中发挥更加重要的角色。

3 功能模块

3.1 L2C 解析模块，用于对 L2C 程序进行建模并解析为符号系统

3.2 不变式生成模块，基于给定的系统模型和属性，自动地生成不变式，以增强验证过程的效果。

3.3 模型检测模块，通过对系统的状态空间进行遍历，检查属性规约是否满足。

4 结论

在本文综述中，我们探讨了 L2C 到 Lustre 的转换及其验证的关键技术和方法，特别是归纳推理算法在确保转换过程正确性方面的应用。虽然这一过程提出了不少挑战，包括保持语法和语义的一致性以及处理转换过程中的复杂性，但通过现有的技术进展，我们可以看到实现高效、准确转换的可能性。未来的工作将集中在优化转换策略、提高算法效率，以及扩展工具支持的范围上，以推动复杂系统开发实践的创新与改进。总之，L2C 到 Lustre 的转换及其验证技术的发展为提高复杂系统的可靠性和安全性提供了有力的支持，并展现出在软件和硬件开发领域的广泛应用前景。

参考文献

[1] 石刚，王生原，董渊，等.同步数据流语言可信编译器的构造[J].软件学报,2014,25(02):341-356.DOI:10.13328/j.cnki.jos.004542.

[2] 康跃馨,甘元科,王生原.同步数据流语言可信编译器 Vélus 与 L2C 的比较[J].软件学报,2019,30(07):2003-2017.DOI:10.13328/j.cnki.jos.005755.

[3] Caspi P, Pilaud D, Halbwachs N, Plaice J. Lustre: A declarative language for programming synchronous systems. In: Proc. of the 14th ACM Symp. on Principles of Programming Languages (POPL'87). Munchen, 1987. 178-188.

[4] Halbwachs N, Caspi P, Raymond P, Pilaud D. The synchronous dataflow programming language LUSTRE. Proc. of the IEEE, 1991, 79(9): 1305－1320 . [doi:10.1109/5.97300]

[5] Shang S, Gan YK, Shi G, Wang SY, Dong Y. Key Translations of the Trustworthy Compiler L2C and Its Design and Implementation[J]. Journal of Software, 2017, 28(5): 1233-1246(in Chinese).http://www.jos.org.cn/1000-9825/5213.htm

[6] Champion, A., Mebsout, A., Sticksel, C., Tinelli, C. (2016). The KIND 2 Model Checker. In: Chaudhuri, S., Farzan, A. (eds) Computer Aided Verification. CAV 2016. Lecture Notes in Computer Science(), vol 9780. Springer, Cham.

[7] ^Gadelha M R, Monteiro F, Cordeiro L, et al. ESBMC v6. 0: Verifying C programs using k-induction and invariant inference[C]//International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, Cham, 2019: 209-213. https://doi.org/10.1007/978-3-030-17502-3_15

[8] 尚书, 甘元科, 石刚, 王生原, 董渊. 可信编译器 L2C 的核心翻译步骤及其设计与实现[J]. 软件学报, 2017, 28(5): 1233-1246.

[9] ^Donaldson A F, Haller L, Kroening D, et al. Software verification using k-induction[C]//International Static Analysis Symposium. Springer, Berlin, Heidelberg, 2011: 351-368. https://doi.org/10.1007/978-3-642-23702-7_26

[10] ^Gadelha M R, Monteiro F, Cordeiro L, et al. ESBMC v6. 0: Verifying C programs using k-induction and invariant inference[C]//International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, Cham, 2019: 209-213. https://doi.org/10.1007/978-3-030-17502-3_15

[11] 刘帅, 魏峰玉, 黄怡桐, 江建国. k-归纳模型检测结果的认证器[J].应用数学进展, 2022,11(11).

[12] 谢小小 XH.CMake 入门实践(一)什么是 cmake[EB/OL].[2024.4.19].http://t.csdnimg.cn/xjFyJ.

[13] 孙凯. 基于嵌入式平台的深度学习人脸识别技术研究[D].南京邮电大学,2023.DOI:10.27251/d.cnki.gnjdc.2022.000364.

[14] 张赛. 基于静态分析的 c 语言程序安全验证方法研究[D].天津理工大学,2023.DOI:10.27360/d.cnki.gtlgy.2023.000210.

[15] Champion, A., Mebsout, A., Sticksel, C., Tinelli, C. (2016). The KIND 2 Model Checker. In: Chaudhuri, S., Farzan, A. (eds) Computer Aided Verification. CAV 2016. Lecture Notes in Computer Science(), vol 9780. Springer, Cham.

[16] Edmund M. Clarke;Thomas A. Henzinger;Helmut Veith;Roderick Bloem.Handbook of Model Checking[J].Springer, Cham,2018.

[17] Halbwachs N.;Caspi P..The synchronous data flow programming language LUSTRE[J].Proceedings of the IEEE,1991(9).

[18] Lina Marsso.Specifying a Cryptographical Protocol in Lustre and SCADE[J].Electronic Proceedings in Theoretical Computer Science,2020.

[19] Temesghen Kahsai;;Cesare Tinelli.PKind: A parallel k-induction based model checker[J].Electronic Proceedings in Theoretical Computer Science,2011.

[20] 卫俊杰. Lustre 语言安全性及活性模型检测工具的研究与实现[D].华东师范大学,2024.DOI:10.27149/d.cnki.ghdsu.2023.000630.

[21] Bošnački, D., & Wijs, A. (2018). Model checking: recent improvements and applications. International journal on software tools for technology transfer : STTT, 20(5), 493 - 497. https://doi.org/10.1007/s10009-018-0501-x
[22] Merz, S. (2000). Model checking: A tutorial overview. Summer School on Modeling and Verification of Parallel Processes, 3-38.