http: //www. acabice. cn acabice@163. com

引用格式: 陈立前，吴国福，姜加红 航天嵌入式软件静态分析技术[J]. 空间控制技术与应用，2021，47(2): 86-92. CHEN L Q，WU G F，JIANG J H. Static analysis technique for aerospace embedded software [J]. Aerospace Control and Application，2021，47(2): 86-92 (in Chinese). doi: 10.3969/j.issn.1674-1579.2021.02.012

# 航天嵌入式软件静态分析技术

陈立前[1*]，吴国福[2]，姜加红[3]

1. 国防科技大学计算机学院，湖南 长沙 410073
2. 国防科技大学空天科学学院，湖南 长沙 410073
3. 北京跟踪与通信技术研究所，北京 海淀 100095

**摘 要**: 软件故障已成为航天系统失败的重要因素. 源代码级程序错误仍是航天嵌入式软件中最突出的问题之一，数组越界、算术溢出、除以零、指针错误、数据竞争等问题仍经常发生. 静态分析能够在编译时通过分析源代码来推断程序运行时性质，是提高航天嵌入式软件安全性与可靠性的一种重要技术. 首先将分析航天嵌入式软件的代码特征及常见错误. 在此基础上，介绍了适合于航天嵌入式软件错误检测的静态分析技术，包括抽象解释、符号执行、数据竞争检测等. 然后，介绍了国内近年来在面向航天嵌入式软件的静态分析技术与工具方面的研究进展. 最后，对未来发展方向做了简单展望.

**关键词**: 嵌入式软件; 程序分析; 静态分析; 缺陷检测

**中图分类号**: TP311 **文献标志码**: A **文章编号**: 1674-1579(2021)02-0086-07

## 0 引 言

随着我国卫星、大运载火箭、载人航天、探月计划的发展，航天嵌入式软件的规模和复杂程度日益增大，系统并发性不断增强，软件系统变得日趋庞大和难以驾驭，软件系统中潜在的缺陷也越来越难以检测、定位和控制[1]. 软件故障已成为航天系统失败的重要因素[2]. 源代码级程序错误仍是航天嵌入式软件中最突出的问题之一，数组越界、算术溢出、被零除、空指针解引用、数据竞争等问题仍经常发生.

静态分析是一种在编译时通过分析程序源代码来推断程序运行时性质的程序分析技术，是检测代码缺陷、提高代码安全性与可靠性的一种重要技术. 尽管程序分析是一类不可判定问题，然而通过适当的抽象，静态分析能够构造程序行为的某种近似，并得出有意义的分析结果. 程序静态分析的目标是在程序运行前尽可能多地发现其中隐含的错误，以提高程序的可靠性和安全性. 对于一些常见的源代码级程序错误，虽已有较成熟商业化静态分析工具(如 Coverity、Klocwork、Parasoft 等) 支持其检测，但是这些工具主要基于语法规则、缺陷模式等开展分析，程序语义信息利用不够，且对于航天嵌入式软件中常见的中断并发机制、浮点运算、面向可靠性的容错设计等代码特征和要素考虑不够，因此这些工具还不能很好地适用于航天嵌入式软件的分析.

为此，本文首先将分析航天嵌入式软件代码特征及常见错误. 在此基础上，介绍适合于航天嵌入式软件错误检测的静态分析技术，包括抽象解释、符号执行、数据竞争检测等. 这些静态分析技术都通过不同方式自动分析并利用了程序的语义信息.

收稿日期: 2020-11-16; 录用日期: 2021-01-22
基金项目: 国家自然科学基金资助项目(61872445)
* 通信作者: E-mail: lqchen@ nudt. edu. cn

## 1　航天嵌入式软件代码特征及常见错误

航天嵌入式软件对可靠性要求非常高,软件开发过程有严格的评审环节,软件设计考虑容错与冗余设计,软件编码实现遵循严格的安全编程规范. 遵循严格的编程规范,使用程序语言的安全子集编程,从某些方面规避了程序错误的引入,也简化了程序分析的难度,如无需考虑递归函数、非结构化程序结构( goto 语句) 等. 但是,航天嵌入式软件中也有一些特征和要素,也给程序分析带来了挑战.

航天嵌入式软件在本质上都是基于物理模型的,需要实现导航、姿轨控等模块,因此不可避免地会包含大量数值计算,涉及许多整型、浮点型变量及其上的运算,从而容易导致除以零、整数溢出、浮点溢出等运行时错误[1]. 在涉及姿态轨道控制的功能实现方面,航天软件会大量使用浮点运算. 这些计算模块的源数据来自敏感器模数( AD) 转换后的数据,如果不对数据的有效性进行检查,数据处理过程中可能出现非法浮点数等问题,导致整个系统数据错误. 同时,由于浮点数的有限二进质制表示,在处理极小数时,可能产生除以零的问题,导致无法得到正确结果. 浮点的误差累积可能对结果的计算精度造成较大影响.

为了更好封装数据,航天嵌入式软件会大量使用数组、结构体、联合体等数据结构,从而容易出现初始化不完全、数组越界等数据使用问题. 航天嵌入式软件大部分使用 C 语言编程,程序代码中会大量使用指针,经常会使用指针访问内存绝对地址、数组、结构体或传递函数参数等. 指针的使用增加了编程实现的灵活性,但 C 语言中指针不受约束,指针访问可能超出所定义内存区域范围,造成越界访问,导致不相干数据被破坏、函数调用堆栈被覆盖等严重问题.

航天嵌入式软件功能复杂,往往采用多任务( 多进程) 设计,并且其执行会受外部中断源的影响. 多任务调度、中断处理带来的并发可能导致数据竞争、数据访问原子性违反等问题. 数据竞争问题仅在特定外部环境和输入下才会出现,难以测试、调试和复现,问题遗漏率较高,是航天嵌入式软件可信性保障中最迫切需要解决的问题之一.

航天嵌入式软件中可能包含符合设计的无限循环( 如用于事件处理、采集数据等) . 但是,不符合预期设计的无限循环可能给系统带来严重的后果. 代码设计逻辑不当会导致在某些条件下对循环变量进行更改,导致无法满足循环退出条件. 程序进入无限循环后,可能导致系统崩溃或者长时间占用 CPU 无法为其他模块服务.

航天嵌入式软件中往往融入三模冗余等容错技术来提高系统的可靠性. 比如,为了防止外空间高能粒子对 RAM 数据造成的翻转破坏,对关键数据采用三取二的方式,即数据保存三份副本,使用时采用三取二的表决方式得到正确的值. 数据保存三份副本,新数据产生时需要同步更新三份数据,可能由于人为疏忽造成数据更新不同步,带来关键数据错误等问题. 因此,需要从代码层面检查同时更新了三份数据以及表决程序本身代码实现的正确性. 此外,看门狗程序、数据缓冲模块( 乒乓存储、环形缓存等) 、数字滤波器等航天嵌入式软件中常见模块的正确性,也需要保证.

## 2　面向航天嵌入式软件的抽象解释技术

抽象解释是一种对程序语义进行可靠近似的通用理论,可用于构造和逼近程序不动点,为静态分析提供了一个通用的框架[3]. 基于抽象解释的静态程序分析框架一般包括编译前端、不动点求解器和抽象域库三个模块. 编译前端主要负责将源程序建模成迁移系统,不动点求解器通过迭代的方式调用抽象域的操作来计算程序不变式,而抽象域库一方面为静态分析过程中需处理的值、表达式、约束等元素提供统一的数据类型,另一方面为区间抽象域、八边形抽象域、多面体抽象域等抽象域的域操作提供了通用的接口.

上世纪九十年代末,基于抽象解释的静态分析工具已被用来识别导致阿丽亚娜 5 号火箭失效的程序错误和类似错误[4]. 近年来,基于抽象解释的静态分析工具在工业界大规模嵌入式软件尤其是航空航天嵌入式软件的分析与验证中得到了成功应用[5-6].

PolySpace 是最早采用抽象解释的商业化静态分析工具之一. 目前,其产品 Polyspace Code Prover,支持形式化地证明软件中不存在严重的运行时错误( 如不存在溢出、除以零、数组访问越界等运行时错误) ; 其产品 Polyspace Bug Finder,能够检查编码规则、安全标准、代码质量指标,并能发现嵌入式软

件代码中的运行时错误、并发问题、安全漏洞等缺陷[7]. 最近，Alenia Aermacchi 公司开发了符合 DO-178B A 级认证的 M-346 教练机飞行控制系统自动驾驶仪软件. 在该自动驾驶仪软件的开发和认证过程中，该公司使用了 Polyspace 静态分析工具来检查代码中是否存在运行时错误，确保符合 MISRA C 编码标准，并为 DO-178B A 级认证的取证提供了证据[8].

Cousot 等[5]基于抽象解释开发了程序静态分析工具 ASTREE，并以航空航天嵌入式安全关键软件系统作为主要研究对象，分析和验证这些软件系统的一些关键性质，并自动检测其中的运行时错误[6]. 这些错误包括数组越界、算术溢出、除以零等. ASTREE 被成功应用于空客 A340(约 13.2 万行 C 代码)、A380(约 35 万行 C 代码)等系列飞机的飞行控制软件的分析，并实现了“零误报”[9]. 在航天领域，ASTREE 证明了欧空局“儒勒·凡尔纳”号自动货运飞船 ATV (Automated Transfer Vehicle) 的自动对接软件(约 19 万行 C 代码)不含运行时错误[10]. 最近，作为 ASTREE 的并发扩展版本，Miné 等[11]基于抽象解释开发了面向并发程序分析的商业化静态分析工具 AstreeA，并实际应用于空客飞行系统软件的分析，被分析程序的最大规模达到百万行 C 代码.

NASA 开发了基于抽象解释的 C 程序运行时错误检查工具 C Global Surveyor(CGS)[12]，并成功地应用于 Mars Path-Finder(13.5 万行代码)、Deep Space One(28 万行代码)、Mars Exploration Rover(65 万行代码)等 NASA 的火星探测项目的软件质量保证. CGS 重点检查 C 程序的三种运行时错误：访问未初始化变量、访问空指针、数组越界访问. 经过针对 NASA 软件的优化，其误报率被控制在 10% 左右. 同时 CGS 利用多处理器平台支持对程序的并行分析，从而进一步提高了其可扩展性.

另外，Goubault 等[13]基于抽象解释开发了用来分析浮点程序中舍入误差的传播情况并发现舍入误差的阶及其源头的静态分析工具 Fluctuat，并在 ATV 自动货运飞船的监控与安全保障单元(MSU)软件等工业界航空航天控制软件的分析与验证中也得到了成功应用[10].

## 3　面向航天嵌入式软件的符号执行技术

符号执行技术使用符号化的值(而非具体值)作为输入，来(抽象)执行程序. 这样，每次(抽象)执行就可以覆盖具有相同执行路径的多个输入，分析器可以搜集每次执行对应的路径约束，然后通过约束求解器求解得到可触发该路径的具体输入值. 总体而言，相比抽象解释符号执行可以避免误报. 符号执行发现的错误对应一条真实可行的路径，从而可以生成触发该错误的测试用例. 符号执行技术最初在上世纪 70 年代提出，但是当时的硬件和算法能力尚不够，所以应用受限. 近年来，随着 SAT 和 SMT 求解技术的快速发展、相关判定算法的研究进展以及现代计算机运算速度的提升，符号执行在越来越多的实际场景中变得适用可行，成为近十余年的研究热点之一. 基于符号执行技术构建的分析工具和测试工具也逐渐被工业界所采纳并使用. 目前，比较著名的符号执行工具有 KLEE、SAGE、Pex、SPF、DART、CUTE、$S^2E$、Cloud9 等.

符号执行技术对程序的路径空间进行遍历，是一种路径敏感的分析技术，因此分析精度高. 但是，当被分析程序具有较大规模时，符号执行面临路径空间爆炸问题. 为了提高符号执行的可扩展性和效率，近年来，研究人员开展了大量研究，并提出了多种优化技术. 其中，将符号执行与具体执行相结合(称为 concolic execution)的思想的提出，对推动符号执行的发展起了重要作用. Godefroid、Sen 等人提出了动态符号执行(dynamic symbolic execution)技术，将具体执行(如随机测试)和静态符号执行方法结合起来，使用具体执行中的信息辅助符号执行，增加覆盖率，并能减少符号执行的约束求解开销. 动态符号执行有效缓解了第三方库源码不可见、复杂路径条件(如非线性表达式、超越函数等)超出约束求解器能力范围等静态符号执行面临的挑战性问题.

Java PathFinder(JPF)是美国 NASA 开发的面向 Java 字节码程序的分析和验证工具[14]. 该工具最早是基于模型检验思想设计的，用于检测并发程序中的数据竞争、死锁等问题. 早期的 JPF，将 Java 代码翻译为 Promela 代码，然后使用模型检验工具 SPIN 来检验性质，并应用于航空领域实时操作系统 DE-OS 的验证中[15]. 后来，该工具逐渐引入静态分析的思想，采用基于符号执行的方法对软件进行分析和验证[16]. JPF 是 NASA 航天器控制软件验证的重要工具，如勇气号、机遇号火星探测器的控制软件都采用 JPF 进行了分析和验证.

## 4　面向航天嵌入式软件的数据竞争检测技术

随着航天软件并发性不断增强,数据访问冲突问题已成为航天嵌入式软件质量保障中最迫切需要解决的问题之一[17]. 数据访问冲突是指多个并发执行流(如任务、中断、线程)对同一数据单元进行同时读写、且其中至少有一个操作是写操作. 由于数据访问冲突中涉及的两次访问之间的次序不可确定,程序可能会因此产生异常行为,严重时甚至会导致软件或系统失效. 这类问题的产生是由于航天器的不同器件之间、多个并发运行的软件之间存在复杂的数据交互,如果同步、互斥机制安排不当,很容易产生数据竞争、原子性破坏等数据访问冲突问题,从而导致软件和系统故障. 已往的航天嵌入式软件研制经验表明,数据竞争问题仅在特定外部环境和输入下才会出现,难以测试、调试和复现,问题遗漏率较高. 同时,因为这类问题涉及到并发软件之间的复杂交叠和时序关系,所以其检测比较困难. 基于模型检验和符号执行的 Java PathFinder(JPF)、基于抽象解释的静态分析工具 AstreeA 都支持多线程程序的数据竞争检测.

航天嵌入式软件的典型并发特征是中断驱动. 由于中断导致的数据竞争问题,曾经导致多起系统故障,比如某卫星帆板驱动线路盒转角跳变导致控制偏差[18]. 据中国空间技术研究院软件产品保证中心统计,在航天器总装测试(AIT.)阶段发现的软件质量问题中,约 80% 都与中断密切相关[18],这已经成为影响我国航天任务完成的技术障碍. 因此,研究针对中断并发、中断与任务共存相关数据竞争分析方法和工具具有重要现实意义. 中断驱动型软件包含常规任务、中断处理函数等成分,是一种特殊的并发软件. 一方面,中断驱动型软件的分析面临一般并发程序的共性问题,即因为状态的并发组合导致的整体状态空间爆炸问题; 另一方面,与一般的多线程程序不同,因为中断触发的不确定性和中断控制的动态性,中断驱动型程序的执行过程,需要进行特殊的语义解释.

除了航天领域,汽车电子领域的软件也常常包含中断. Schwarz 等[19] 提出了一种针对中断驱动型程序的数据竞争检测方法,其主要针对汽车电子领域使用天花板协议(priority ceiling protocal)的中断程序. 其主要思想是在抽象解释框架下,使用任务与中断间的全局不变式,并基于该不变式计算共享变量的资源锁,然后通过资源锁来判定程序中是否存在数据竞争,并开发了工具 Goblint[20]. 目前,Goblint 已成功应用于 OSEK 程序、Linux 设备驱动程序等程序的数据竞争检测和锁错误检测.

## 5　国内航天嵌入式软件静态分析工具研究

近年来,国内学术界和工业界在将静态分析技术与工具应用于航天嵌入式软件的质量保证方面开展了不少研究. 尤其,国内航天软件评测部门采用静态分析技术也开发了一些面向航天软件的静态分析工具,并在实际航天软件进行了应用,为航天软件质量保证提供了有力支撑. 北京轩宇信息技术有限公司研发的 C/C++ 静态缺陷检测工具 Sunwise SpecChecker 支持安全编码标准符合性检查、运行时缺陷检测和代码质量度量等,采用跨函数、跨中断的抽象解释技术对程序进行高效的并发语义分析,除了支持如空指针、未初始化变量、数组越界、除零错、整数溢出等运行时错误的检测,还支持中断数据访问冲突缺陷的检测. 该公司研制的针对 C 语言程序的单元与集成测试平台 Sunwise AUnit,采用了动态符号执行和约束求解技术来加速 C 程序测试用例的快速生成,提高了测试效率. 这些工具已经在中国空间技术研究院全面应用,促进了型号软件质量,并推广至航空、电力等多个领域[21]. 王崑声等[22] 针对航天嵌入式软件提出了一种基于属性模型的运行时错误静态分析方法,并开发了相应的工具,在航天嵌入式软件上开展了实验.

在航天嵌入式软件数据竞争检测方面,吴学光等[17] 针对多重中断航天嵌入式 C 程序开展了数据竞争及原子性检测相关研究,并开发了多重中断环境下数据竞争和原子性检查工具 MIDAC. 该工具支持共享变量分析、数据竞争检测、原子性违反检测等. 进一步,WU 等[23] 在有界模型检验框架下提出了一种面向多任务、多中断嵌入式软件的数据竞争检测方法,将可能产生数据竞争的路径条件编码为 SAT/SMT 公式,然后通过判断公式的不可满足性来消除数据竞争误报. 进一步,WU 等[24] 提出了基于顺序化的中断驱动型程序的数值性质分析方法,将中断驱动型程序顺序化为带非确定性的顺序程序,然后使用面向顺序程序的数值性质分析方法来分析顺序化中断驱动型程序,以检查中断驱动型程序

中的数值相关运行时错误，并在真实航天软件上开展了实验. 陈睿等[18]以航天嵌入式软件数据竞争案例库为基础，经过系统分析，提出了刻画有害中断数据竞争的 7 种缺陷模式，并针对其中最常见且最难解决的单变量访问序模式，提出一种支持过程间分析、中断并发分析的高效检测方法，设计并实现了相应的检测工具 SpaceDRC，在多个航天重点型号中进行了应用. 进一步，陈睿等[25]设计开发了航天嵌入式软件中断数据访问冲突基准测试集程序 RaceBench，对 SpaceDRC 工具进行了评估.

表 1 典型静态分析技术及其在航天嵌入式软件中的适用方面与可改进方面

Tab. 1 Static analysis techniques and its applications to aerospace embedded software

| 技术 | 抽象解释 | 符号执行 |
|---|---|---|
| 优点 | 可扩展性好、可保证分析的终止性 | 分析精度高、可产生触发错误的输入 |
| 缺点 | 误报率较高、难以产生反例 | 路径空间爆炸问题、约束不可解或求解代价大 |
| 典型工具 | ASTREE、PolySpace、CGS、FramaC-Eva、Sparrow、CRAB-LLVM、IKOS 等 | KLEE、JPF、SAGE、Pex、SPF、DART、CUTE、$S^2E$ 等 |
| 适合分析的航天软件性质或错误 | 数组越界、算术溢出、除以零等数值相关错误，误差累计，最差执行时间，资源使用量上界等 | 缓冲区溢出、整数溢出、非法指针访问、内存泄漏、时序控制错误等 |
| 面向航天软件的可改进方面 | 提高分析精度，降低误报，支持时序等更多功能相关性质的分析 | 支持浮点程序的分析及浮点约束求解优化，支持中断、多核等并发行为 |

## 6 结 论

本文首先分析了航天嵌入式软件的代码特征及常见错误. 在此基础上，介绍了适合于航天嵌入式软件错误检测的静态分析技术，包括抽象解释、符号执行、数据竞争检测等. 然后，介绍了国内近年来在面向航天嵌入式软件的静态分析技术与工具方面的研究进展. 表 1 给出了本文主要介绍的抽象解释、符号执行两种典型静态分析技术及其优缺点，以及在航天嵌入式软件中的适用方面与可改进方面. 限于篇幅，本文只是简述了面向航天嵌入式软件的一部分静态分析技术及其进展[26].

虽然近年来国内在面向航天嵌入式软件的静态分析技术与工具方面取得了不少进展，未来需要进一步研究和开发适合于航天嵌入式软件特征、支持领域关键模块(如数据缓冲区等)、面向更多类型性质与缺陷、面向航天嵌入式软件未来发展方向(如多核等)的静态分析技术与工具，为载人航天、探月工程和卫星等国家重大科技专项中的软件质量保障提供支撑.

## 参 考 文 献


[1] 蔡铭，程胜，王瑞. 航天型号高可信软件系统调试原理与技术[M]. 北京: 中国宇航出版社，2008.

[2] 杨海成，乔永强，许胜，等. 航天型号软件工程[M]. 北京: 中国宇航出版社，2011.

[3] COUSOT P，COUSOT R. Abstract interpretation: a unified lattice mode for static analysis of programs by construction or approximation of fixpoints[C]// The 4th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Texas: ACM，1977.

[4] LACAN P，MONFORT J N，RIBAL L V Q，et al. ARIANE 5—the software reliability verification process[C]// ESA SP-422: The Conference on Data Systems in Aerospace. ESA，1998.

[5] BLANCHET B，COUSOT P，COUSOT R，et al. A static analyzer for large safety-critical software[C]// The ACM SIGPLAN 2003 Conference on Programming language design and implementation. Texas: ACM，2003.

[6] SOUYRIS J，DELMAS D. Experimental assessment of astree on safety-critical avionics software[C]// The 26th International Conference on Computer Safety，Reliability，and Security. New York: Springer，2007.

[7] THE MATHWORKS INC. Polyspace [EB/OL]. [2020-11-23]. https://ww2.mathworks.cn/products/polyspace.html.

[8] THE MATHWORKS INC. Alenia Aermacchi 开发符合 DO-178B A 级认证的自动驾驶仪软件[EB/OL]. [2020-11-23]. https://ww2.mathworks.cn/company/user_stories/alenia-aermacchi-develops-autopilot-software-for-do-178b-level-a-certification.html.

[9] BERTRANE J，COUSOT P，COUSOT R，et al. Static analysis and verification of aerospace software by ab-

stract interpretation [J]. Found. Trends Program. Lang., 2015, 2(2-3): 71-190.

[10] BOUISSOU O, CONQUET E, COUSOT P, et al. Space software validation using Abstract Interpretation [C] // The 13th Data Systems in Aerospace. 2009.

[11] MINé A, DELMAS D. Towards an industrial use of sound static analysis for the verification of concurrent embedded avionics software [C] // The 15 th International Conference on Embedded Software. New York: IEEE, 2015.

[12] NASA. C Global Surveyor [EB/OL]. [2020-11-23]. https: // ti. arc. nasa. gov/tech/rse/vandv/cgs/.

[13] DELMAS D, GOUBAULT E, PUTOT S, et al. Towards an industrial use of fluctuat on safety-critical avionics software [C] // The 14th International Workshop on Formal Methods for Industrial Critical Systems. New York: Springer, 2009.

[14] HAVELUND K, PRESSBURGER T. Model checking Java programs using Java pathfinder [J]. STTT, 2000, 2(4): 366-381.

[15] VISSER W, HAVELUND K, BRAT G P, et al. Model checking programs [C] // The 15th IEEE International Conference on Automated Software Engineering. New York: IEEE, 2000.

[16] PASAREANU C S, VISSER W, BUSHNELL D H, et al. Symbolic pathfinder: integrating symbolic execution with model checking for Java bytecode analysis [J]. Autom. Softwares Eng., 2013, 20(3): 391-425.

[17] 吴学光,文艳军,王戟,等. 多重中断 C 程序中数据竞争及原子性检测 [J]. 计算机科学与探索,2011,5(12):1085-1093.
WU X, WEN Y, WANG J,et al. Data race and atomicity checking for c programs with multiple interruptions [J]. Journal of Frontiers of Computer Science and Technology, 2011, 5(12): 1085-1093.

[18] 陈睿, 杨孟飞, 郭向英. 一种基于变量访问序模式的中断数据竞争检测方法 [J]. 软件学报, 2016, 27(3): 547-561.
CHEN R, YANG M F, GUO X Y. Interrupt data race detection based on shared variable access order pattern [J]. Journal of Software, 2016,27(3):547-561.

[19] SCHWARZ M D, SEIDL H, VOJDANI V, et al. Static analysis of interrupt-driven programs synchronized via the priority ceiling protocol [C] // The 38 th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Texas: ACM, 2011.

[20] VOJDANI V, APINIS K, RÕTOV V, et al. Static race detection for device drivers: the Goblint approach [C] // The 31st IEEE/ACM International Conference on Automated Software Engineering. Texas: ACM, 2016.

[21] 北京轩宇信息技术有限公司. C/C++静态缺陷检测工具 SUNWISE SpecChecker [EB/OL]. [2020-11-23]. http: // www. sunwiseinfo. com/product/278050290.

[22] 王崑声,詹海潭,经小川,等. 航天嵌入式软件运行时错误静态分析方法 [J]. 北京理工大学学报. 2013, 33(2): 160-165.
WANG K, ZHAN H, JING X, et al. Program static analysis on runtime error for aerospace embedded software [J]. Transactions of Beijing Institute of Technology. 2013, 33(2): 160-165.

[23] WU X, WEN Y, CHEN L, et al. Data race detection for interrupt-driven programs via bounded model checking [C] // The 7th International Conference on Software Security and IEEE Computer Society. New York: IEEE,2013.

[24] WU X, CHEN L, MINE A, et al. Numerical static analysis of interrupt-driven programs via sequentialization [C] // The 15 th ACM SIGBED International Conference on Embedded Software. New York: IEEE Computer Society, 2015.

[25] 陈睿, 杨孟飞. 航天嵌入式软件数据访问冲突基准测试集研究 [J]. 中国空间科学技术, 2017, 37(3): 62-70.
CHEN R, YANG M. Study on aerospace embedded software data race benchmark [J]. Chinese Space Science and Technology. 2017, 37(3): 62-70.

[26] 张健, 张超, 玄跻峰, 等. 程序分析研究进展 [J]. 软件学报, 2019, 30(1): 80-109.
ZHANG J, ZHANG C, XUAN J F, et al. Recent progress in program analysis [J]. Journal of Software, 2019, 30(1): 80-109.

**作者简介:** 陈立前(1982—),男,副教授,研究方向为高可信软件技术、软件分析与验证; 吴国福(1980—),男,副研究员,研究方向为卫星高可靠软件设计技术、卫星地面快速测试技术; 姜加红(1989—),男,助理研究员,研究方向为航天软件分析与软件测试.

# Static Analysis Technique for Aerospace Embedded Software

CHEN Liqian[1*], WU Guofu[2], JIANG Jiahong[3]

1. College of Computer, National University of Defense Technology, Changsha 410073, China;
2. College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China;
3. Beijing Institute of Tracking and Telecommunication Technology, Beijing 100095, China

**Abstract:** Software faults have become one of the main causes of the failure of aerospace systems. Bugs in source code are still one of the most prominent kinds of problems in aerospace embedded software, such as array out of bounds, arithmetic overflow, divided by zero, data race, etc. Static analysis, which can infer runtime properties of a program at compile time through analyzing its source code, is an important technique to improve safety and reliability of aerospace embedded software. The code features and common kinds of program errors in aerospace embedded software are reviewed in this paper. After that, the static analysis techniques that are fit for analyzing aerospace embedded software are presented, including abstract interpretation, symbolic execution, data race detecting, etc. Then, some recent advance on applying static analysis techniques to aerospace embedded software in China are reviewed. Finally, some future directions on this topic are discussed.

**Keywords:** embedded software; program analysis; static analysis; defect detection

Received: 2020-11-16; Accepted: 2021-01-22
Foundation items: National Nature Science Foundation of China( 61872445)
* Corresponding author. E-mail: lqchen@nudt.edu.cn