

**学士学位论文中期报告**

**论文题目**：B\*语言解释器的实现

**专 业**：计算机科学与技术

**本 科 生**：赵继伟

**学 号**：10061118

**指导教师**：马殿富

**北京航空航天大学计算机学院**

2014年04月16日

目 录

[1 题目背景与意义 3](#_Toc385191444)

[1.1 课题来源 3](#_Toc385191445)

[1.2 选题的背景与意义 3](#_Toc385191446)

[2 课题研究内容回顾 4](#_Toc385191447)

[2.1 研究目标 4](#_Toc385191448)

[2.2 研究核心内容 4](#_Toc385191449)

[3 工作进度 6](#_Toc385191450)

[3.1 工作进度概况图 6](#_Toc385191451)

[3.2 工作进度描述 6](#_Toc385191452)

[3.3 已完成部分工作流程 7](#_Toc385191453)

[3.4 已完成部分软件模型 8](#_Toc385191454)

[3.5 软件效果图 8](#_Toc385191455)

[4 待完成工作及问题 8](#_Toc385191456)

[4.1 待完成工作 8](#_Toc385191457)

[4.2 存在的问题 9](#_Toc385191458)

[5 后期进度安排 9](#_Toc385191459)

[6 参考文献 9](#_Toc385191460)

# 题目背景与意义

## 课题来源

民用飞机专项科研项目“多核处理的机载实时操作系统关键技术研究”

## 选题的背景与意义

随着航空工业的不断发展，机载软件的规模、复杂程度日趋提高，并且是呈倍级增加的趋势。根据美国相关报道，美国空军新一代联合攻击机F-35的Final Block 3 Software计划约860万行代码[1]。根据相关资料，空中客车公司的历代飞机的在板软件大小（Volume of onboard software）也是逐渐提升，如下表[2]

|  |  |  |  |
| --- | --- | --- | --- |
| 飞机型号 | A310 | A320 | A340 |
| 在板软件大小（Mbytes） | 4 | 10 | 20 |

同时，在规模和复杂度提高的前提下，机载软件的安全性与可靠性都被做出了极高的要求。机载软件的标准于2011年从旧的DO-178B提升到了DO-178C[3],使得我们需要对于机载软件这类安全关键软件做出更加可靠的验证，对软件的质量提出了更高的要求。在DO-178C标准中，着重强调了采用形式化方法开展基于模型的开发和验证。

另外，于2009年提出的ISO/IEC 15408国际信息技术安全评估标准[4]，对于软件的形式化验证也提出了严格的要求。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CC级别 | 需求 | 功能规约 | 高层设计 | 低层设计 | 代码实现 |
| EAL1 | 非形式化 | 非形式化 | 非形式化 | 非形式化 | 非形式化 |
| EAL2 | 非形式化 | 非形式化 | 非形式化 | 非形式化 | 非形式化 |
| EAL3 | 非形式化 | 非形式化 | 非形式化 | 非形式化 | 非形式化 |
| EAL4 | 非形式化 | 非形式化 | 非形式化 | 非形式化 | 非形式化 |
| EAL5 | 形式化 | 部分形式化 | 部分形式化 | 非形式化 | 非形式化 |
| EAL6 | 形式化 | 部分形式化 | 部分形式化 | 部分形式化 | 非形式化 |
| EAL7 | 形式化 | 形式化 | 形式化 | 部分形式化 | 非形式化 |
| 完全验证 | 形式化 | 形式化 | 形式化 | 形式化 | 形式化 |

为了能够实现对于航空机载软件这类安全关键性软件的形式化验证，我们需要一种工具，可以将验证过程中机械化的过程交给软件自动完成，并且针对不可自动完成的部分，优化数据呈现模式，使得验证者可以更加清晰、方便的完成人工验证过程。最终达到可以加快形式化验证速度，提升形式化验证效率，提高形式化验证准确率的目标。

本人的毕业设计即是要完成上述目标，完成一个软件供形式化验证人员使用。

# 课题研究内容回顾

## 研究目标

根据上文的理论模型，完成一个形式化验证工具，将用B\*语言代码编写的低层规格描述，通过交互式的方式，实现逻辑语句的生成，完成整个形式化验证的过程，取代过去完全人工验证的方式，加快形式化验证速度，提高形式化验证准确度。

## 研究核心内容

为了完成研究目标，需要一个B\*语言的解释器，来自动化处理用B\*语言代码。然而，该项目需要的解释器不是一个传统的解释器，它核心目的是进行形式化验证，导出有关于低层需求的逻辑语句。所以，本解释器虽然在前端语法分析方面与传统解释器上相似，但是其后端的语言虚拟机部分却有很大区别。

传统解释器每个变量在某一个时刻只会存在一个值，也就是说整个语言的语境（在具体的程序中，指的是全部变量的值）的变化是线性的，每个分支语句（if）只会完成一种true/false的分支并执行。根据上文提到的算法验证模型的分支语句部分，本解释器需要对分支语句的每种运行可能进行检查，也就是语境的变化是非线性的，是一个树状展开的过程，这便是本解释器面临的第二个问题。

再者，传统解释器整个的解释过程是全自动不需要人工干预的，但是根据上文提到的算法验证模型，一旦遇到循环语句，其逻辑语句的导出方法是基于数学归纳法的，这个过程目前还是必须由人工来完成，所以说，如何在利用自动化的手段，让这个唯一的人工干预的过程变得更加快速而准确，便是本解释器面临的一个问题。

另外，根据上文中提到的数据结构验证模型，本解释器面临的第三个问题就是如何实现集合数据结构与线性表、二叉树，以及其他的数据结构之间的对等的映射关系。

综上所述，我们研究的核心内容，也是研究中的难点，为以下三点

* 非线性变化的语境
* 便利快捷的循环语句的逻辑语句导出
* 常用数据结构的集合化

# 工作进度

## 工作进度概况图

|  |
| --- |
|  |
| 图 3-1-1 图中白色底色的模块为已基本完成的模块，灰色底色的模块为尚未完成的模块。 |

## 工作进度描述

到目前为止，本人的毕设的编码工作已经完成了大半。软件的总体框架已经搭建完成，而框架中的8个主要模块已经完成了6个。

以代码量计算，本人自己编写的Java代码行数已达到4500行以上，自己编写的BNF范式为200行，共完成了42个类。

## 已完成部分工作流程



## 已完成部分软件模型

在曾经的开题报告中提到过，循环部分是三个主要研究内容中最为核心的部分。为解决这个部分，我提出了一套自己的方法。

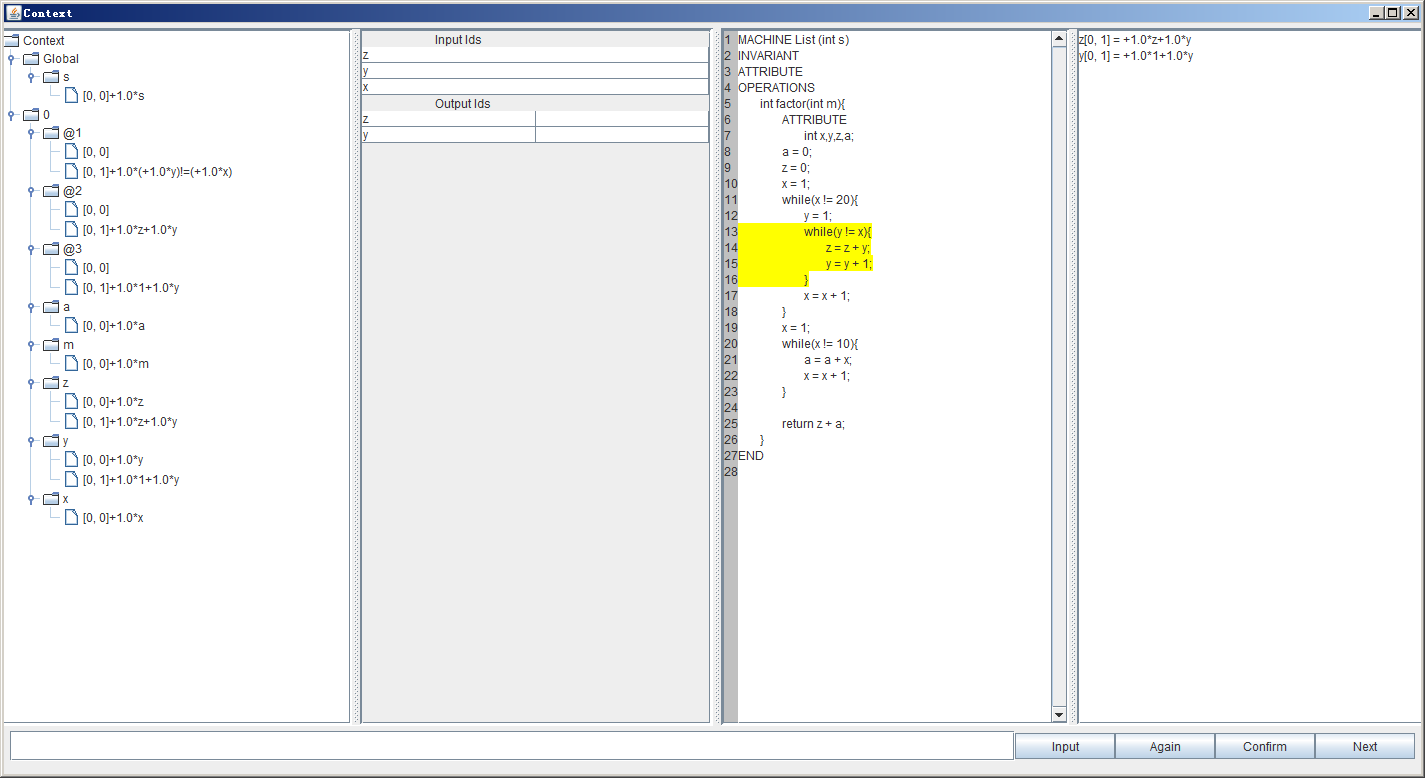
其中，我们把每个循环语句作为一个循环块来处理。对于每一个循环块，它都有一组输入参数和一组输出参数，我们需要求解的是输入参数与输出参数之间的关系，以及在产生输出的过程中生成的逻辑语句。

通过分析代码以及研究验证过程我们可以发现，对于每一个循环块而言，它在输入与输出参数之间构成的关系，是上下文无关的，也就是说我们可以把每一个循环块单独来处理。

在求解出循环块所形成的有关于输入输出参数之间的公式之后，我们可以通过代码替换的方式，将该循环块的代码替换为求解出的公式。这样，在后来的处理中，该循环块代码从需要人工处理的代码，就变为了可以自动化验证的代码。

重复上述工作，直至所有循环语句全部被替换，这样，全部的代码都可以自动化验证，也就解决了本软件中最关键的部分。

## 软件效果图



# 待完成工作及问题

## 待完成工作

* 完成剩余的两个模块
* 系统整体测试
* 翻译论文
* 编写毕业设计论文

## 存在的问题

* 系统中还有一些零散的Bug
* 语法树的测试不够全面

# 后期进度安排

|  |  |
| --- | --- |
| 时间 | 工作 |
| 2014年4月16日至2014年5月1日 | 完成代码编写 |
| 2014年5月2日至2014年5月8日 | 完成论文翻译 |
| 2014年5月9日至2014年6月2日 | 完成毕设论文编写 |

# 参考文献

1. Warwick, Graham. "Flight Tests Of Next F-35 Block Underway." Aviation Week, 12 June 2010.
2. Jean-Louis Camus. “Efficient Development of Airborne Software with SCADE Suite™.” Handbook. 2003
3. <http://www.rtca.org/store_list.asp>
4. ISO/IEC 15408