基于Büchi自动机化简的JavaMOP监控器构造方法

汇报人：

2024-01-27

contents

目录

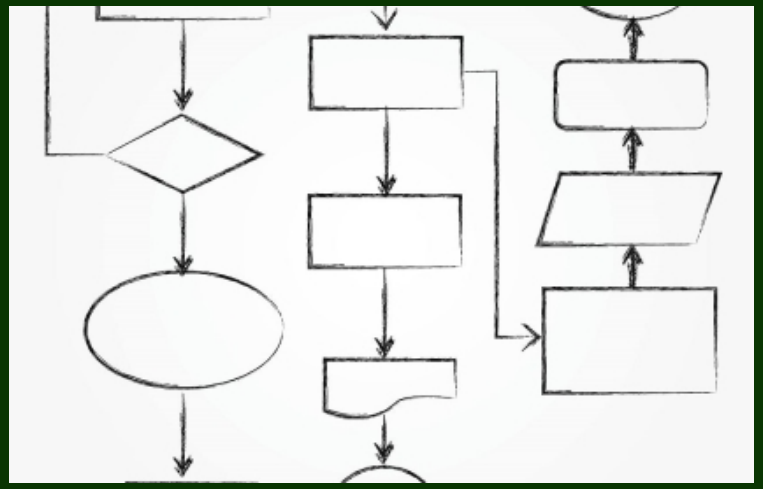
01

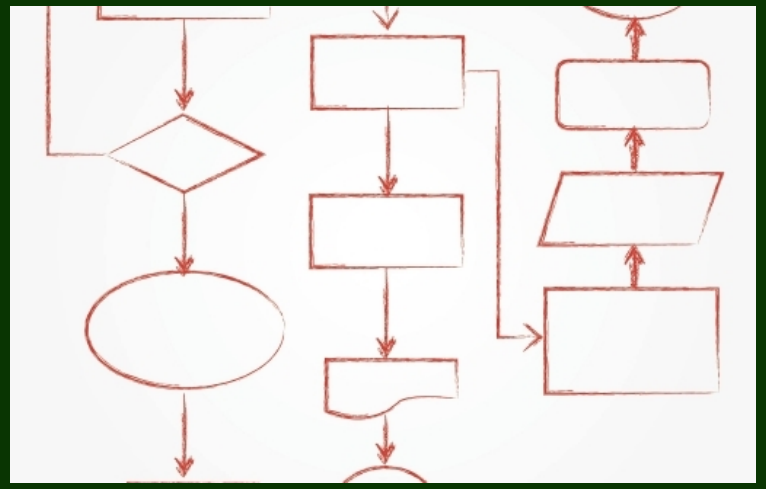
引言

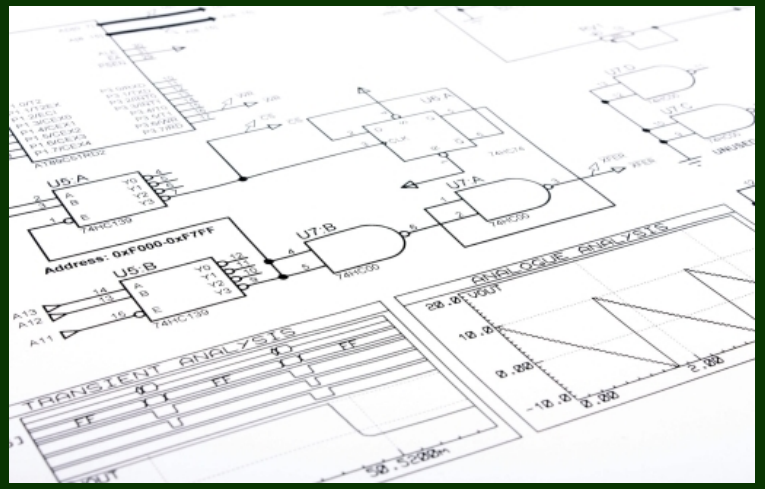
CHAPTER

研究背景与意义

- 监控器在运行时验证系统行为是否符合预期规格方面发挥着重要作用。
- Büchi自动机是一种用于描述无限状态系统的形式化工具，在验证和测试领域具有广泛应用。
- JavaMOP是一种基于Java的监控器构造工具，支持运行时验证和测试。
- 研究基于Büchi自动机化简的JavaMOP监控器构造方法，对于提高监控器的效率和准确性具有重要意义。







国内外研究现状及发展趋势

国内外研究现状

目前，已有一些基于Büchi自动机的监控器构造方法，但存在效率不高、准确性不足等问题。同时，JavaMOP作为一种流行的监控器构造工具，也面临着类似的问题。

发展趋势

随着形式化方法和运行时验证技术的不断发展，基于Büchi自动机的监控器构造方法将更加注重效率和准确性的提升。同时，随着JavaMOP等工具的不断完善，基于这些工具的监控器构造方法也将更加成熟和实用。

研究内容、目的和方法

1

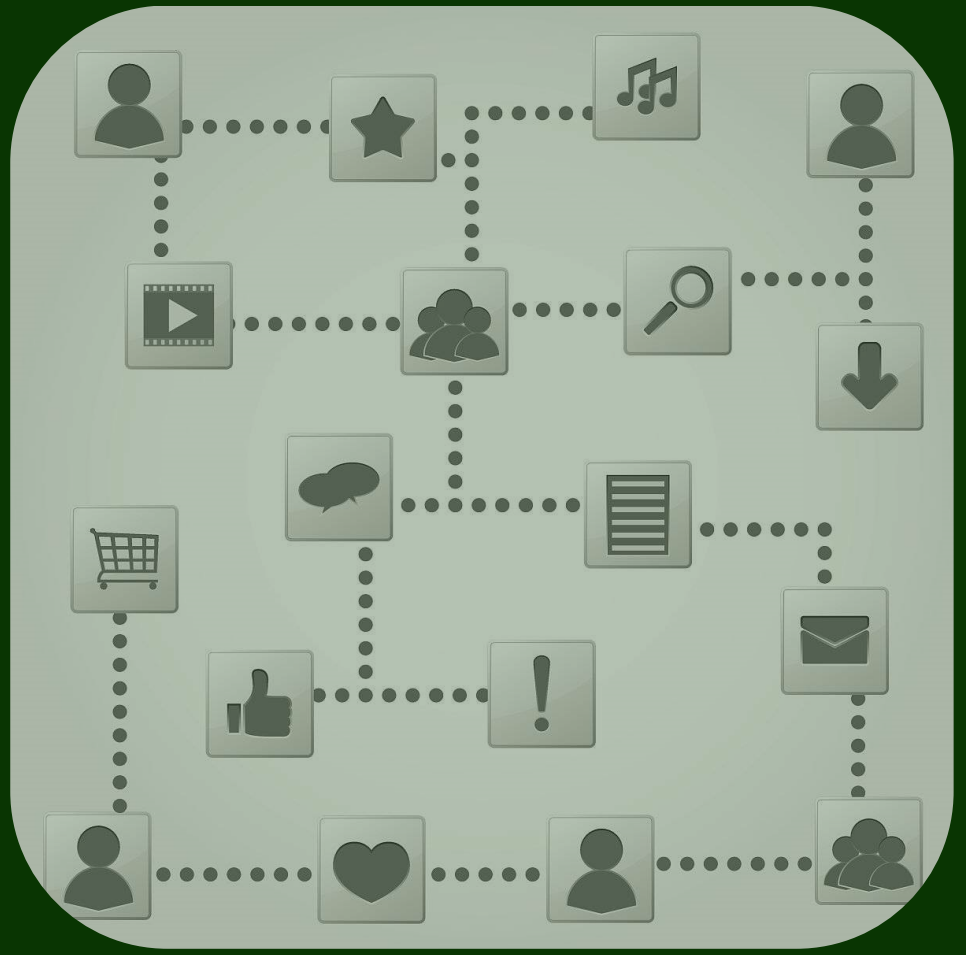
研究内容：本研究旨在探讨基于Büchi自动机化简的JavaMOP监控器构造方法。具体内容包括

2

分析Büchi自动机在监控器构造中的应用及其存在的问题；

3

研究基于Büchi自动机化简的监控器构造算法；



研究内容、目的和方法

研究目的

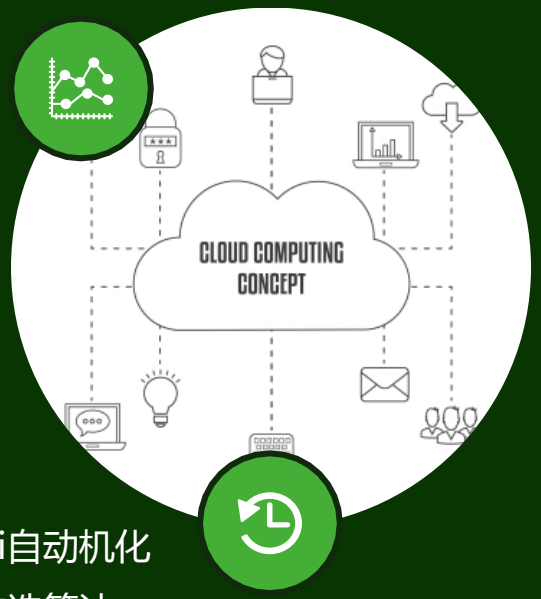
本研究旨在提高监控器的效率和准确性，为运行时验证和测试提供更加可靠的工具支持。同时，通过探讨基于Büchi自动机化简的监控器构造方法，为相关领域的研究提供新的思路和方法。

研究方法

本研究将采用理论分析和实证研究相结合的方法。具体包括

研究内容、目的和方法

对Büchi自动机和JavaMOP等相关理论进行深入分析；



开发基于JavaMOP的监控器构造工具，并进行实验验证；

设计并实现基于Büchi自动机化简的监控器构造算法；

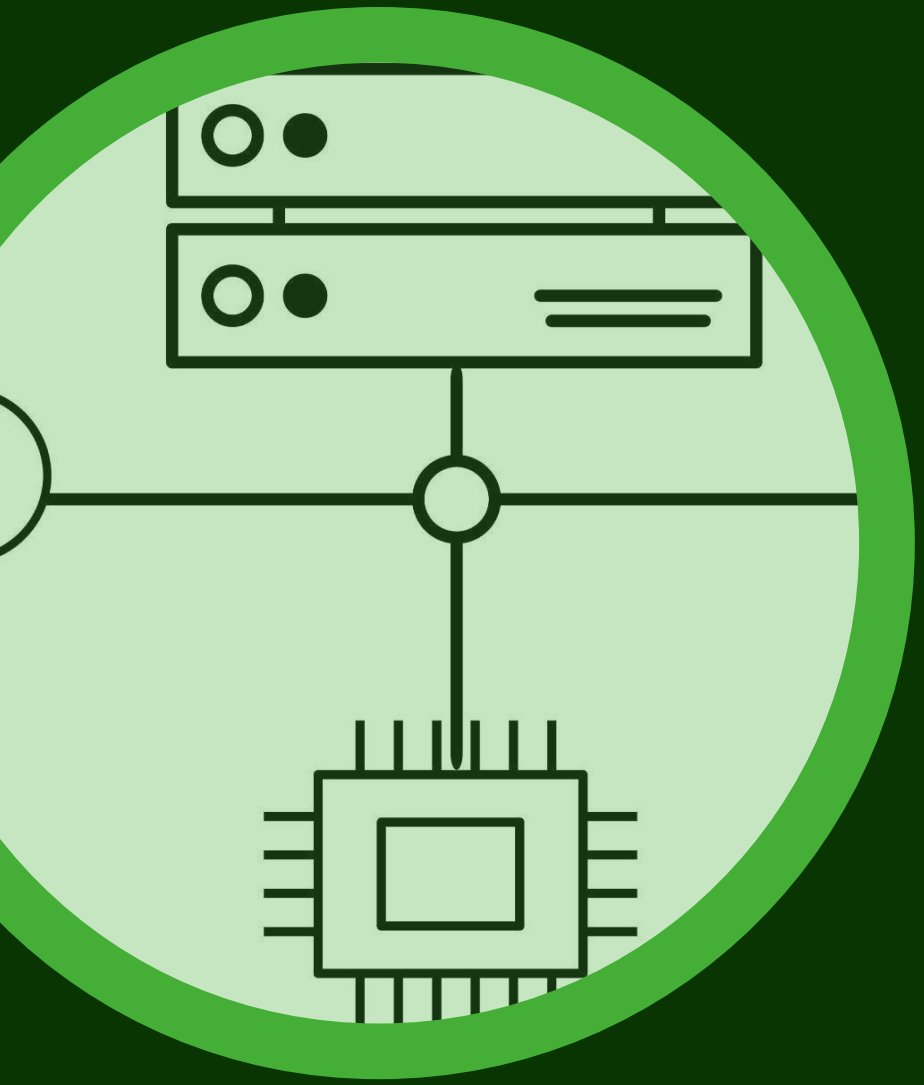
对实验结果进行分析和讨论，评估所提方法的有效性和效率。

02

Büchi自动机化简理论

CHAPTER

Büchi自动机基本概念



01

Büchi自动机是一种无限状态自动机，用于描述和验证无限序列上的性质。

02

Büchi自动机由一组状态、一组输入符号、一组转移函数和一个接受状态集组成。

03

Büchi自动机通过读取输入序列并根据转移函数进行状态转移，来判断输入序列是否满足特定的性质。

Büchi自动机化简方法

状态合并

将具有相同行为的状态合并为一个状态，减少状态数量。

转移函数化简

通过分析和优化转移函数，减少不必要的状态转移。

接受状态集化简

对接受状态集进行化简，使得自动机更加简洁。

Büchi自动机化简算法

Paige-Tarjan算法

一种基于路径压缩和并查集的化简算法，具有较快的化简速度。

Hopcroft算法

一种基于等价关系的化简算法，通过计算状态间的等价关系来合并状态。

其他算法

如基于模拟退火、遗传算法等的启发式化简算法，适用于特定场景下的Büchi自动机化简。

03

JavaMOP监控器构造方法

CHAPTER

JavaMOP监控器基本概念

监控器（Monitor）

在JavaMOP中，监控器是用于观察和记录程序运行行为的实体，它可以捕获和响应程序中的特定事件。

事件（Event）

事件是程序运行过程中发生的特定行为或状态变化，如方法调用、对象创建等。

JavaMOP通过定义事件来指定监控器的触发条件。

切点（Pointcut）

切点是用于定义事件发生的上下文和条件的表达式，它描述了监控器应该在哪些程序点上被触发。

JavaMOP监控器构造流程

定义事件

首先，需要明确需要监控的程序行为，并定义相应的事件来表示这些行为。

编写切点表达式

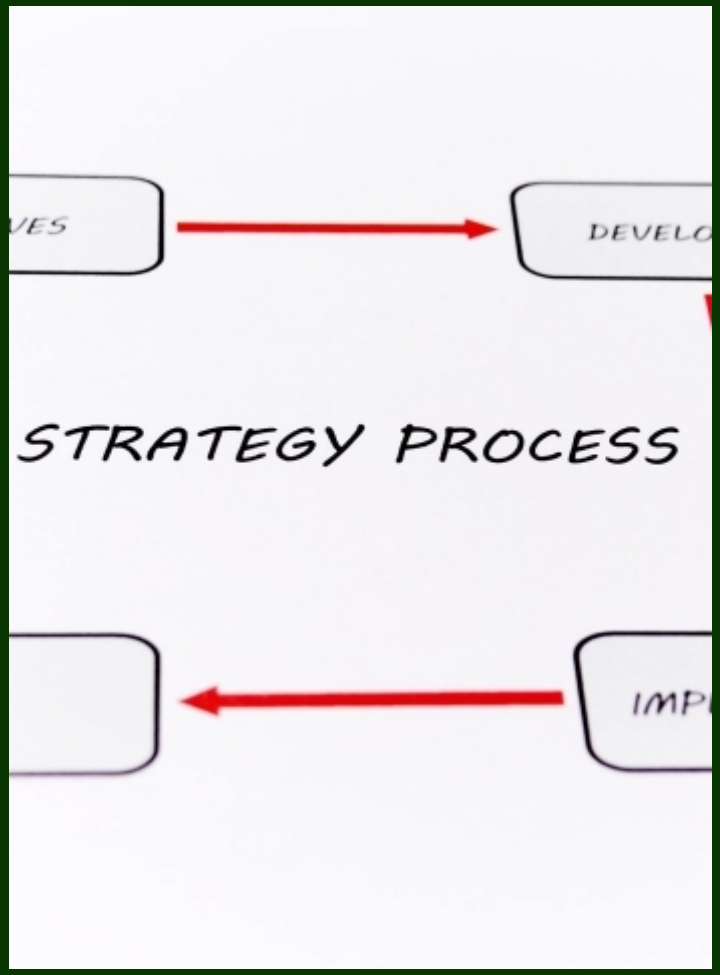
根据监控需求，编写切点表达式来指定监控器触发的条件和上下文。

实现监控器逻辑

在监控器中编写处理逻辑，定义在事件发生时应该执行的操作，如记录日志、发送通知等。

注册监控器

将编写好的监控器注册到JavaMOP框架中，以便在程序运行时自动触发和执行。



JavaMOP监控器实现技术

反射（Reflection）

JavaMOP利用Java反射机制来动态获取程序运行时的信息，如类、方法、字段等，以便实现灵活的监控。

字节码操作

JavaMOP使用字节码操作技术来修改被监控程序的字节码，从而实现在程序运行时动态插入监控逻辑。

事件驱动编程

JavaMOP采用事件驱动编程模型，允许开发者定义事件和监听器来处理程序中的特定行为，简化了监控器的编写过程。

上下文感知

JavaMOP提供了上下文感知功能，使得监控器能够获取事件发生时的程序上下文信息，如调用栈、变量值等，有助于更准确地分析和诊断问题。

04

基于Büchi自动机化简的JavaMOP监控器构造方法

CHAPTER