

Тестирование конкурентных структур  
данных на соответствие моделям  
согласованности

Морковкин Василий

2020

# Мотивация

Чтение данных с записывающего устройства, проведение сетевых запросов и одновременная обработка поступающих команд - одни из задач, стоящих перед программным обеспечением. Результатом оптимизации затрат процессорного времени на решение этих задач стало появление ряда техник *конкурентного программирования*. Однако необходимость использования ресурсов, не поддерживающих одновременный доступ, зависимость от системного планировщика задач, а также особенности устройства процессорных кешей могут приводить к неопределенным или нежелательным исполнениям компьютерных программ. Целью конкурентного программирования является предоставление механизмов и методик, которые бы позволили писать корректный, обладающий определенным поведением код. Одним из важнейших механизмов являются *структуры данных*, пригодные для использования в конкурентной среде. Для описания их свойств и предоставляемых гарантий был разработан ряд *моделей согласованности*.

Проверка алгоритмов на соответствие этим моделям может быть осуществлена с помощью методов формальной верификации. Однако для проверки конкретных их реализаций на языках программирования доступно лишь тестирование с перебором некоторого подмножества всех сценариев исполнения. В этой работе будут рассмотрены существующие модели согласованности конкурентных структур данных, разработаны варианты их тестирования, и, наконец, реализованы в виде библиотеки на языке программирования Scala.

# Оглавление

<b>1</b>	<b>Введение</b>	<b>4</b>
1.1	Основные понятия . . . . .	4
1.2	Цели работы . . . . .	4
1.3	План работы . . . . .	4
<b>2</b>	<b>Конкурентное программирование</b>	<b>5</b>
2.1	Проблемы . . . . .	5
2.1.1	Состояние гонки . . . . .	5
2.1.2	Гонка на данных . . . . .	5
2.1.3	Взаимная блокировка . . . . .	5
2.2	Модели согласованности . . . . .	5
2.2.1	Последовательная согласованность . . . . .	5
2.2.2	Линеаризуемость . . . . .	5
2.2.3	Ослабленная линеаризуемость . . . . .	5
<b>3</b>	<b>Тестирование моделей согласованности</b>	<b>6</b>
3.1	Алгоритмическая сложность тестирования . . . . .	6
3.2	Алгоритмы тестирования . . . . .	6
3.2.1	Последовательная согласованность . . . . .	6
3.2.2	Линеаризуемость . . . . .	6
3.2.3	Ослабленная линеаризуемость . . . . .	6
3.3	Разработка требований к библиотеке . . . . .	6

3.3.1	Интерфейс . . . . .	6
3.3.2	Конфигурация . . . . .	6
<b>4</b>	<b>Реализация библиотеки</b>	<b>7</b>
4.1	Интерфейс . . . . .	7
4.2	Генерация тестовых сценариев . . . . .	7
4.3	Запуск тестовых сценариев . . . . .	7
4.4	Валидация результатов исполнения . . . . .	7
4.5	Генерация отчета тестирования . . . . .	7
<b>5</b>	<b>Выводы</b>	<b>8</b>
<b>6</b>	<b>Литература</b>	<b>9</b>

# **Введение**

## **1.1 Основные понятия**

## **1.2 Цели работы**

## **1.3 План работы**

# **Конкурентное программирование**

## **2.1 Проблемы**

### **2.1.1 Состояние гонки**

### **2.1.2 Гонка на данных**

### **2.1.3 Взаимная блокировка**

## **2.2 Модели согласованности**

### **2.2.1 Последовательная согласованность**

### **2.2.2 Линеаризуемость**

### **2.2.3 Ослабленная линеаризуемость**

# **Тестирование моделей согласованности**

## **3.1 Алгоритмическая сложность тестирования**

## **3.2 Алгоритмы тестирования**

### **3.2.1 Последовательная согласованность**

### **3.2.2 Линеаризуемость**

### **3.2.3 Ослабленная линеаризуемость**

## **3.3 Разработка требований к библиотеке**

### **3.3.1 Интерфейс**

### **3.3.2 Конфигурация**

# **Реализация библиотеки**

**4.1 Интерфейс**

**4.2 Генерация тестовых сценариев**

**4.3 Запуск тестовых сценариев**

**4.4 Валидация результатов исполнения**

**4.5 Генерация отчета тестирования**



# **Выводы**

# **Литература**