

**Ministry of Education and Science of the Russian Federation**

THE FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER EDUCATION

**"SAINT PETERSBURG NATIONAL RESEARCH UNIVERSITY OF  
INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS"**

**ANNOTATION**

**OF THE GRADUATION QUALIFICATION WORK**

**Student** Artem Yushkovskiy

**Thesis title:** Automated Analysis of Weak Memory Models

**Institution:** Aalto University (Espoo, Finland), ITMO University (St Petersburg, Russia)

**SPECIFICATION OF THE GRADUATION QUALIFICATION WORK**

1 Research goal Redesign of the architecture of the static code analyser Porthos in order to extend the input language, enhance performance, extensibility, reliability and maintainability.

2 Addressed challenges (1) examining existing approaches for the analysis of the behaviour of parallel programs executed on hardware with weak memory models; (2) extension of the support for syntactic constructions of the input language of static portability analyser Porthos; (3) design the low-level representations used by Porthos (the event-flow graph); (4) development of an extensible C interpreter as a part of the non-optimising compiler infrastructure for static portability analyser PorthosC.

3 Number of sources used for review 53

4 Total number of sources, used in the thesis 65

5 Including sources below, listed by year of publication

Local (Russian)			Foreign		
Last 5 years	From 5 to 10 years	More than 10 years ago	Last 5 years	From 5 to 10 years	More than 10 years ago
0	0	0	27	20	18

6 Use of Internet information resources yes, 1

7 Use of packages of computer programs and technologies

Packages of computer programs and technologies	Chapter
HERD (memory model-aware symbolic checker)	2
ANTLR (parser generator)	4
Z3 (SMT solver)	4
IntelliJ IDEA (Java IDE)	4
Graphviz (graph visualiser)	4

8 Short summary of obtained results (1) basic approaches and tools for analysis of the correctness of parallel programs under a hardware architecture with weak memory model were studied

(Chapters 1-3); (2) the Porthos analyzer architecture was revised, the necessary components and internal code representations were identified (Chapter 4); (3) to support the instructions of unconditional jumps of C language (goto, break, continue) the encoding scheme for the program control flow was modified (Section 3.2.1); (4) a new DFS-based loop unrolling algorithm for an arbitrary control flow graph is proposed; the algorithm produces all possible program executions within the user-defined bound (Sections 4.3.2.6, 5.1.2); (5) a complete infrastructure of non-optimizing compiler with extensible knowledge base of semantics of functions invocations is developed (Sections 4.3.2.5, 5.1.1); (6) it is shown that the developed compiler and the unroller do not add significant time overhead comparing to the original Porthos program, while the accuracy of the analysis increased due to the modified loop unrolling algorithm (Chapters 5, 6).

---

9 Received grants during writing the paper no

10 Publications and conference presentations on the topic of the work no

Student	<u>Artem Yushkovskiy</u>	_____
	(Full name)	(Signature)

Supervisor at Aalto University	<u>Prof. Keijo Heljanko</u>	_____
	(Full name)	(Signature)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20 \_\_\_\_.

Supervisor at ITMO University	<u>Igor I. Komarov</u>	_____
	(Full name)	(Signature)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20 \_\_\_\_.

**Министерство образования и науки Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ”**

## **АННОТАЦИЯ**

### **ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

Студент Юшковский Артем Викторович

Наименование темы ВКР: Автоматический анализ слабых моделей памяти параллельного программирования

Наименование организации, где выполнена ВКР Университет Аалто (Эспоо, Финляндия), Университет ИТМО (Санкт-Петербург, Россия)

### **ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

1 Цель исследования Разработка архитектуры и модификация статического анализатора портируемости параллельных программ Porthos с целью расширения его возможностей по анализу программ на языке программирования C, оптимизации времени работы и повышения надежности результатов анализа.

2 Задачи, решаемые в ВКР (1) изучение существующих подходов к анализу поведения параллельных программ в средах выполнения со слабыми моделями памяти; (2) расширение поддержки статическим анализатором Porthos синтаксических конструкций языка C; (3) проектирование низкоуровневого представления языка C (графа потока событий) для статического анализатора PorthosC; (4) разработка расширяемого интерпретатора программ на языке C как часть инфраструктуры неоптимизирующего компилятора входной программы в граф потока событий, используемого PorthosC;

3 Число источников, использованных при составлении обзора 53

4 Полное число источников, использованных в работе 65

5 В том числе источников по годам

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
0	0	0	27	20	18

6 Использование информационных ресурсов Internet Да, 1

7 Использование современных пакетов компьютерных программ и технологий

Пакеты компьютерных программ и технологий	Глава работы
HERD (memory model-aware symbolic checker)	2
ANTLR (parser generator)	4
Z3 (SMT solver)	4
IntelliJ IDEA (Java IDE)	4
Graphviz (graph visualiser)	4

8 Краткая характеристика полученных результатов (1) изучены основные подходы и инструменты к анализу корректности выполнения параллельных программ на процессорах, имеющих слабую модель памяти (Главы 1-3); (2) пересмотрена архитектура анализатора Porthos, выявлены необходимые компоненты и внутренние представления кода (Глава 4); (3) для поддержки инструкций безусловного перехода языка C (goto, break, continue) изменена схема кодирования потока управления программы в SMT-формулу (Раздел 3.2.1); (4) предложена новая схема развертки циклов в произвольном графе потока управления, основанная на обходе графа в глубину и генерирующая все возможные исполнения программы в пределах определяемой пользователем границы (Разделы 4.3.2.6, 5.1.2); (5) разработана полная инфраструктура неоптимизирующего компилятора C с поддержкой расширения базы знаний семантики вызываемых функций (Разделы 4.3.2.5, 5.1.1); (6) показано, что разработанный компилятор кода на C в граф потока событий и модуль развертки циклов не добавляют значительных временных издержек по сравнению с оригинальной программой Porthos, при этом, точность анализа увеличилась за счет измененного алгоритма развертки циклов (Главы 5, 6).

9 Полученные гранты, при выполнении работы нет

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы нет

Студент

Юшковский А. В.  
(ФИО)

\_\_\_\_\_  
(подпись)

Руководитель  
от Университета Аалто

Prof. Keijo Heljanko  
(ФИО)

\_\_\_\_\_  
(подпись)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ г.

Руководитель  
от Университета ИТМО

Комаров И. И.  
(ФИО)

\_\_\_\_\_  
(подпись)

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ г.