МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи

Ушакова Мария Сергеевна

Формальная верификация функционально-потоковых параллельных программ

05.13.11 — математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Аннотация научно-квалификационной работы

Научный руководитель
доктор техн. наук, профессор
Легалов А.И.

Красноярск 2018

В связи со всё возрастающими требованиями к надежности программного обеспечения, наряду с традиционными методами тестирования, всё чаще стали использоваться методы формальной верификации программ. Формальная верификация — это доказательство корректности программы, которое заключается в установлении соответствия между программой и ее спецификацией, описывающей цель разработки. Методы формальной верификации позволяют доказать отсутствие ошибок в программе, в то время как тестирование лишь выявляет ошибки, но не даёт гарантии их отсутствия.

Существуют различные подходы к формальной верификации программ. Основными являются метод проверки моделей и дедуктивный анализ. Метод проверки моделей (model checking) позволяет осуществить автоматизируемый перебор всех возможных вариантов выполнения программы, но применим только в случае, когда программа принимает конечное число состояний. Метод проверки моделей достаточно эффективно используется при анализе взаимодействия параллельно функционирующих процессов. Однако, существуют и другие подходы к формальной верификации программ, направленные на анализ логики преобразования данных, определяемой решением поставленной задачи. Обычно такие задачи имеют бесконечное число состояний, и их корректность проверяется с помощью дедуктивного анализа. В настоящее время достигнуты определенные успехи в практическом применении дедуктивного анализа для верификации последовательных программ. Для поддержки этого процесса разработан ряд систем. В качестве примера можно привести верификатор программ на языке C Boogie и систему СПЕКТР, а также системы для верификации объектно-ориентированных программ на Java: LOOP и KeY.

Распараллеливание программ позволяет существенно увеличить их производительность на современных вычислительных системах. Однако параллелизм приводит к значительному усложнению разработки и, особенно, отладки. В основном программы пишутся на императивных языках программирования. По сравнению с последовательными, параллельные императивные программы могут содержать новые виды ошибок, которые трудно выявить при тестировании. Также резко увеличивается сложность формальной верификации парал-

лельных императивных программ.

Вместе с тем, всё большую популярность приобретает функциональное программирование, которое ориентируется на отношение между данными. В рамках этого направления формальные методы верификации развиваются достаточно интенсивно. Основополагающей работой в данной области является работа Бойера и Мура. В системе NQTHM они реализовали метод автоматизированного доказательства функций, написанных на языке LISP. Другой пример — доказательство утверждений для программ на Haskell.

Одним из функциональных языков является язык функционально-потокового параллельного программирования Пифагор.

Модель вычислений, лежащая в основе языка задает вычисления в автоматически выделяемых бесконечных ресурсах. Это позволяет не учитывать возможные ресурсные конфликты, что облегчает процесс написания функционально-потоковой параллельной программы. Каждая программа — это функция, поэтому в языке отсутствуют переменные и циклы, а операции выполняются по готовности данных. В результате, сложность формальной верификации ФПП программ сравнима со сложностью верификации последовательных программ. Другая важная особенность языка — возможность достичь максимального параллелизма программы за счет того, что параллелизм реализуется на уровне операций. После доказательства корректности такой программы, она может быть перенесена на конкретную архитектуру с конечными ресурсами, при необходимости, с ограничением ее параллелизма.

В настоящее время существуют работы, связанные с организацией отладки и верификацией ФПП программ, однако вопросы формальной верификации на базе дедуктивного анализа на базе исчисления Хоара не проработаны.

Поэтому актуальной является разработка формальных методов и средств верификации для функционально-потокового языка параллельного программирования, обеспечивающих проверку логики функционирования программ.

Цель работы. Исследование и разработка формальных методов верификации функционально-потоковых параллельных программ, разработка системы, поддерживающей процесс формальной верификации. Для достижения

указанной цели в решаются следующие задачи:

1. Обзор и исследование существующих методов и средств верификации программ.
2. Разработка метода формальной верификации функционально-потокового параллельного языка программирования Пифагор.
3. Разработка архитектуры и создание инструментального средства для поддержки формальной верификации программ на языке Пифагор.

В рамках работы получены следующие научные и практические результаты.

1. Исследованы существующие методы верификации программ, рассмотрена возможность их применения к функционально потоковому языку параллельного программирования Пифагор.
2. Формализовано описание семантики языка Пифагор, описан язык спецификации свойств программ, создана аксиоматическая теория на базе исчисления Хоара, в рамках которой можно доказывать корректность функционально-потоковых параллельных программ.
3. Проведён обзор методов доказательства завершения программ, на основе которого предложен метод доказательства завершения программ на языке Пифагор. Предложен метод устранения взаимной рекурсии нескольких функций.
4. Проведён обзор инструментальных средств поддержки доказательства на основе дедуктивного анализа. Разработана архитектура инструментального средства поддержки доказательства корректности функционально-потоковых параллельных программ. Предложена реализация системы, позволяющая доказывать корректность программ на языке Пифагор. Рассмотрена возможность использования SMT-решателей для проверки истинности и выполнимости генерируемых системой условий корректности.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи

Ушакова Мария Сергеевна

Формальная верификация функционально-потоковых параллельных программ

05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Аннотация научно-квалификационной работы

Научный руководитель
доктор техн. наук, профессор
Легалов А.И.

Красноярск 2018