

На правах рукописи



Удалова Юлия Васильевна

**ОТЛАДКА И ВЕРИФИКАЦИЯ ФУНКЦИОНАЛЬНО-ПОТОКОВЫХ  
ПАРАЛЛЕЛЬНЫХ ПРОГРАММ**

05.13.11 – «Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных  
сетей»

**АВТОРЕФЕРАТ**  
диссертации на соискание ученой степени  
кандидата технических наук

Красноярск – 2015

Работа выполнена в Федеральном государственном автономном образовательном учреждении высшего профессионального образования «Сибирский федеральный университет»

- Научный руководитель: доктор технических наук, профессор  
Легалов Александр Иванович
- Официальные оппоненты: Зюбин Владимир Евгеньевич,  
доктор технических наук, доцент, Федеральное государственное бюджетное учреждение науки  
Институт автоматизации и электрометрии  
Сибирского отделения Российской академии наук, ведущий научный сотрудник;
- Мурзин Федор Александрович,  
кандидат физико-математических наук,  
Федеральное государственное бюджетное учреждение науки Институт систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук, заместитель директора по научной работе;
- Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт динамики систем и теории управления Сибирского отделения Российской академии наук

Защита состоится «21» мая 2015 г. в 16<sup>00</sup> часов на заседании диссертационного совета Д.212.173.06 при Федеральном государственном бюджетном образовательном учреждении высшего профессионального образования Новосибирский государственный технический университет, по адресу: 630073, г. Новосибирск, пр. К.Маркса, 20.

С диссертацией можно ознакомиться в библиотеке Новосибирского государственного технического университета и на сайте <http://www.nstu.ru>.

Автореферат разослан «\_\_» апреля 2015 г.

Ученый секретарь  
диссертационного совета



Фаддеенков Андрей Владимирович

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность работы.** В теории и практике параллельного программирования задачи повышения надежности, отладки и верификации параллельных программ занимают одно из доминирующих положений из-за значительного числа вариантов взаимодействий параллельных участков программы, в которых легко скрываются логические ошибки, риски появления ошибок, связанных с механизмами синхронизации и передачи сообщений, риски тупиковых ситуаций и ресурсных конфликтов.

Вопросам повышения надежности параллельных программ посвящены работы Непомнящего В.А., Карпова Ю.Г. и др., из зарубежных ученых в первую очередь следует выделить работы Хоара Ч., Вудкока Дж., Кларка Э.М. Ведущие зарубежные производители – Microsoft, Intel, Bell Labs, IBM, научные группы ведущих зарубежных и отечественных университетов - Кембриджского университета, Иллинойского университета, ИЦМС ИПС РАН, предлагают отладчики и верификаторы императивных параллельных программ, предоставляющие обширный инструментарий, например Intel Parallel Composer, TotalView, TDB, mpC Workshop, Panorama, SFP, Spin, Isabelle, Bogor, VCC, RRD и др. Ведутся работы в направлении верифицирующих компиляторов.

Современное параллельное программирование в основном опирается на императивную парадигму, позволяющую разрабатывать программы, эффективно использующие предоставляемые вычислительные ресурсы. Вместе с тем, развивается и архитектурно-независимое параллельное программирование, основной задачей которого является создание параллельных программ, не связанных с конкретными параллельными вычислительными системами. Один из таких подходов опирается на функционально-потокową парадигму параллельного программирования, предложенную в работах А.И. Легалова.

В настоящее время инструментальные средства разработки функционально-потокowych параллельных программ обеспечивают только

выполнение программы и простейшую отладку. Средства автоматической верификации функционально-поточковых параллельных программ на сегодняшний день отсутствуют. В связи с этим является актуальной задача создания средств, обеспечивающих поддержку отладки и верификации.

**Цель и задачи исследования.** Целью работы является исследование и разработка методов и инструментальных средств, обеспечивающих отладку и верификацию функционально-поточковых параллельных (ФПП) программ.

Для достижения указанной цели в работе решаются следующие задачи:

- анализ существующих методов отладки и верификации параллельных программ, выбор среди них методов, поддерживающих эффективную отладку и верификацию функционально-поточковых параллельных программ;
- разработка новых методов и алгоритмов, обеспечивающих отладку и верификацию ФПП программ;
- создание инструментальных средств, обеспечивающих поддержку предложенных методов отладки и верификации.

**Предмет и объект исследования.** Объектом исследования является функционально-поточковая модель параллельных вычислений. Предмет исследования – методы отладки и верификации ФПП программ.

**Методы исследования.** Поставленные задачи решались с использованием методов доказательства теорем, формального доказательства правильности программ, методов проверки моделей, элементов теории графов, теории языков программирования, математической логики и интервального анализа.

При создании программных инструментов для отладки и верификации использовались структурное и объектно-ориентированное программирование.

**Научные результаты, выносимые на защиту.**

1. Методы и алгоритмы обхода траекторий отладки функционально-поточковых параллельных программ.

2. Метод верификации функционально-поточковых параллельных программ, использующий информационные графы программы, спецификацию пользователя и интервальные константы.

3. Метод верификации функционально-поточковых параллельных программ с асинхронными списками.

4. Архитектура программной среды, обеспечивающей поддержку разработанных методов отладки и верификации функционально-поточковых параллельных программ.

### **Научная новизна результатов исследования.**

1. Предложен метод отладки функционально-поточковых параллельных программ, базирующийся на различных принципах обхода траекторий отладки, позволяющий в сочетании с методами визуализации отладочной информации повысить эффективность процесса поиска и анализа ошибок, как в текстовом, так и графическом режимах представления функционально-поточковых параллельных программ. Некоторые принципы обхода траекторий отладки и элементы применения графов встречаются в отладчиках ФПП программ, но представленный комплекс режимов отладки уникален.

2. Разработан метод верификации функционально-поточковых параллельных программ, основанный на принципе обхода информационного графа с использованием интервальных констант, позволяющий осуществлять автоматизированную проверку спецификации пользователя и получить интервальные оценки результатов работы операторов. Метод опирается на известные методы доказательства теорем, изначально сформулированные для императивных последовательных программ. Применение методов доказательства теорем к ФПП программе в совокупности со спецификацией данных ФПП программы на графе с помощью интервальных констант оригинальны.

3. Разработан метод верификации функционально-поточковых параллельных программ с асинхронными списками, базирующийся на

принципе комбинаторного анализа результатов выполнения программных операторов, позволяющий формировать оценочный пакет результатов выполнения программ с асинхронными списками. Базой метода являются общие принципы известного метода проверки модели, применение его к верификации ФПП программ с асинхронными списками оригинально.

Методы отладки и верификации разработаны с учетом применения к функционально-потокowому параллельному языку Пифагор, для которого до получения результатов диссертационной работы был реализован только простейший отладчик, а способов автоматической верификации не существовало.

### **Практическая значимость работы.**

1. На основе предложенных методов отладки и верификации разработаны инструментальные средства, интегрированные в существующую систему функционально-потокowого параллельного программирования Пифагор.

2. Результаты работы использовались при выполнении научно-методического проекта «Среда разработки для языка параллельного программирования Пифагор» в рамках «Программы развития СФУ на 2007–2010 годы».

3. Результаты работы использовались при выполнении научно-методического проекта «Инструментальная поддержка архитектурно-независимой разработки параллельных программ на основе функционально-потокowой парадигмы параллельного программирования» в рамках федеральной целевой программы «Научные и научно-педагогические кадры инновационной России», выполнявшегося в 2012-2013 гг.

4. Полученные научные результаты использованы в учебном процессе по дисциплине «Технология программирования» в виде лекций по этапам разработки программного обеспечения для подготовки специалистов по специальности 230101.65 «Вычислительные машины, комплексы, системы и сети» в ФГАОУ ВПО «Сибирский федеральный университет».

Все результаты практического применения диссертационного исследования подтверждены соответствующими актами и свидетельствами о регистрации программного обеспечения.

**Достоверность полученных результатов** подтверждается корректным использованием математического и программного аппарата, работоспособностью разработанного программного обеспечения (ПО), применением разработанного ПО в научной деятельности по развитию представленных методов отладки и верификации и в учебном процессе.

**Соответствие диссертации паспорту специальности.**

Содержание работы соответствует п. 1 (Модели, методы и алгоритмы проектирования и анализа программ и программных систем, их эквивалентных преобразований, верификации и тестирования.) паспорта специальности 05.13.11 (математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей).

**Апробация результатов диссертации.** Основные положения диссертации докладывались и обсуждались на следующих конференциях и семинарах: четвертой Российско-германской школе по параллельным вычислениям, г. Новосибирск, 2007г.; Всероссийских научно-технических конференциях «Молодежь и наука –XXI век», г. Красноярск, 2007-2009гг.; четвертой Сибирской школе-семинаре по параллельным вычислениям, г. Томск, 2007г.; Международной научной конференции «Параллельные вычислительные технологии (ПаВТ'2009)», г. Красноярск, 2009г.; XI Всероссийской научно-практической конференции «Проблемы информатизации региона (ПИР-2009)», г. Красноярск, 2009г.; Международной Ершовской конференции по Информатике, рабочий семинар «Наукоемкое программное обеспечение», г. Новосибирск, 2011г.; Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма», Новороссийск, 2013г.

**Личный вклад автора** в научные работы, опубликованные в соавторстве с научным руководителем, заключается в следующем: анализе

существующих методов отладки и верификации на предмет применения к ФПП языку; разработке методов и алгоритмов отладки и верификации ФПП программ; представлении архитектуры инструментальной среды для разработки ФПП программ. Совместно с научным руководителем автор осуществлял постановку целей и задач, выбор методов для реализации, анализ полученных результатов. В совместные публикации автора с Сиротининой Н.Ю. и Кропачевой М.С. включены разработанные ими теоретические идеи о верификации ФПП программ на языке Пифагор, являющиеся параллельной научной разработкой и не задействованные в диссертационной работе автора. В совместных публикациях автора с Матковским И.В. и Васильевым В.М., научные результаты автора являются прикладным дополнением к концепции разрабатываемого ими транслятора ФПП программ.

**Публикации.** По теме диссертации опубликовано 20 научных работ, из которых 3 статьи в изданиях, рекомендуемых ВАК и 2 свидетельства о государственной регистрации программ для ЭВМ.

**Структура и объем диссертации.** Диссертация состоит из введения, 5 глав, заключения и трех приложений. Работа содержит 140 страниц основного текста, 49 рисунков и 2 таблицы. Список литературы содержит 134 наименования.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** приводится общая характеристика работы, показывается актуальность работы и ее практическая значимость, формулируются цели работы и положения, выносимые на защиту, дается краткий обзор содержания диссертации.

**В первой главе** проведен анализ методов отладки параллельных программ и предложена классификация методов отладки. Выбраны следующие опции, обеспечивающие эффективную отладку ФПП программ.

1 Способ обхода операторов параллельной программы подразумевает метод выбора подмножества параллельных примитивов для



выполнения шага отладки. Наибольшую эффективность для ФПП программ обеспечивает использование автоматического выбора пути выполнения программы, при котором из множества готовых к выполнению операторов выбирается один или несколько.

2 Распределение шага отладки определяет набор вычисляемых операторов до передачи управления пользователю. Выбраны режимы: пошаговый (в отдельном примитиве); поярусный. Предложен дополнительный режим распределения шага отладки, при котором выделяются операторы, образующие ветвь графа.

3 Типы контрольных точек. Точки останова в ФПП программе могут быть связаны с операторами или прикрепленными к ним пользовательскими условиями, позволяющими повысить эффективность локализации ошибок и задать спецификацию в формальном виде.

4 Отладочные значения являются дополнительными идентификаторами или формулами, введенными пользователем перед началом или во время отладки и не изменяющими саму программу. Их использование позволяет контролировать вычисления без изменения кода. Пользовательские условия для контрольных точек являются дополнительными отладочными значениями.

5 Направление хода отладки в ФПП программе может быть прямым и обратным при любом режиме распределения шага отладки.

6 Возможности визуализации ФПП программы связаны с отображением ациклических графов ее отдельных функций:

- графическое представление отображает операторы (вершины графа) и связи между ними (данные), точки синхронизации программы, уровень ее параллелизма, вершины графа представляют операторы и хранят программную запись оператора и его вычисленное значение;

- визуализация ошибок задается цветом операторов-вершин, выдавших неверные значения.

Во второй главе проведен анализ методов верификации и возможностей их использования при ФПП программировании. Выделены методы, применимые для верификации ФПП программ. Задача верификации сформулирована как верификация на информационном графе программы, что позволяет уменьшать набор рассматриваемых утверждений.

Метод индуктивных утверждений – базовый метод доказательства теорем для верификации как последовательных, так и параллельных программ. Для его применения необходимо знать порядок обхода операторов программы, для обхода ФПП программы предлагается использовать ее информационный граф (рисунок 1).

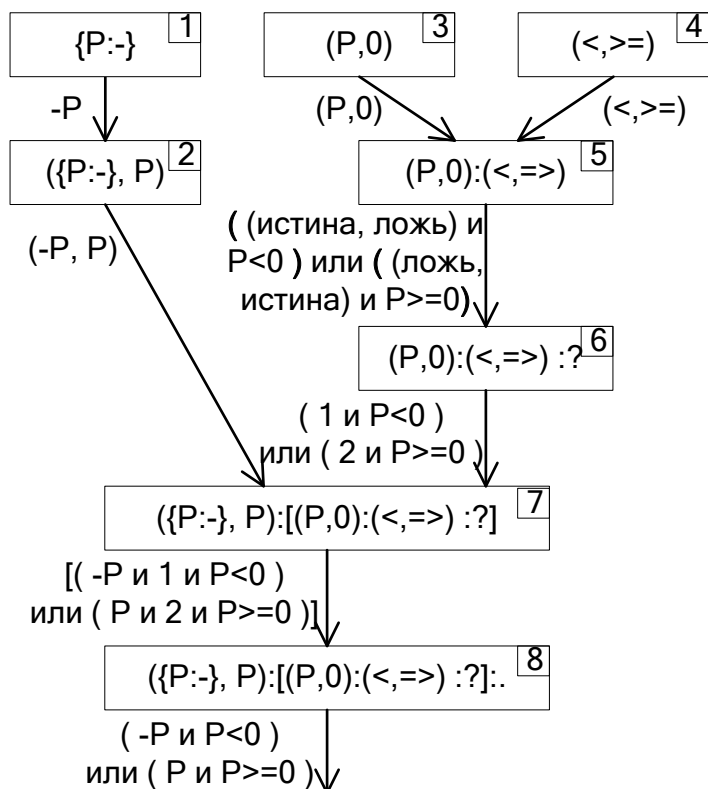


Рисунок 1 – Граф ФПП программы вычисления модуля с выводом утверждений

Верификация на графе обладает таким достоинством, как сужение множества утверждений, рассматриваемых для построения или проверки следующих утверждений. Это связано с тем, что утверждения, получаемые при рассмотрении конкретной вершины-оператора, опираются на предыдущие утверждения не от каждого ранее рассмотренного оператора

программы, а от множества вершин-операторов, связанных с рассматриваемой.

Метод индуктивных утверждений недостаточен для верификации циклов и рекурсий. Повторные вычисления в ФПП программах задаются рекурсиями, для проверки корректности которых выбран метод индукции. Вывод утверждений на итерациях метода индукции осуществляется методом индуктивных утверждений с использованием графа ФПП программы. Метод индукции результативен для ФПП программ и применяется к ним почти также как и для последовательных рекурсивных программ, за исключением необходимости учета задержанных вычислений.

Особенности функционально-поточковой модели параллельных вычислений, такие как независимость вычисленных значений от порядка обхода операторов, отсутствие механизмов синхронизации и передачи сообщений, ресурсных конфликтов, снижают эффективность применения методов проверки моделей, базирующихся на комбинаторном переборе состояний модели программы и подстановке конкретных значений при вычислении таких состояний. Из-за приведенных особенностей ФПП модели верификация ФПП программы методами проверки моделей будет аналогична по полученным результатам отладке ФПП программы с пользовательскими условиями.

Метод проверки модели выбран для верификации программы с асинхронными списками, которые характеризуются недетерминированным поступлением данных. Для каждого оператора, обращающегося к асинхронному списку, выполняется перебор всех возможных вариантов его срабатывания, при обнаружении следующего такого оператора, необходимо также перебрать все возможные его значения, но уже с учетом всех возможных значений, полученных на предыдущем операторе и так далее. Цель применения метода – получение ответа на вопрос: влияет ли использование асинхронного списка на результат.

Для спецификации ФПП программ выбрана форма, определяемая методом индуктивных утверждений, где обязательны входное и выходное утверждение, а промежуточные утверждения необязательны. Последние предлагается приписывать к операторам-вершинам графа. Выбор такой формы спецификации обусловлен тем, что основной модуль верификации будет основан на методе индуктивных утверждений, а также возможностями визуализации ошибочных вычислений на графе, повышающими эффективность локализации некорректных вычислений.

**В третьей главе** предложены методы отладки ФПП программ. На основе проведенного в первом разделе анализа выделяются четыре режима отладки: пошаговая отладка, отладка слоев, отладка ветвей, проверка формул. Представлены различные способы выполнения и отображения процесса отладки в различных режимах визуализации: режим операторов, список функций, свертка функций, отображение на графе программы.

Режим пошаговой отладки подобен аналогичному режиму для последовательных программ. За один шаг выполняется один оператор. Возможность использования пошаговой отладки для ФПП программы обусловлена спецификой языка программирования.

Режим отладки слоев (ярусов) предполагает, что на первом шаге отладки выполняется первый слой, т.е. все те операторы, которым требуются только входные параметры и константы (рисунок 2). После того, как эти операторы окажутся вычисленными, соответствующие им вершины помечаются как выполненные, и на следующем шаге для вычисления выбираются все те операторы-вершины, которые имеют на своем входе вычисленные данные, таким образом формируется следующий слой.

Отладка по слоям предоставляет пользователю возможность наглядно увидеть количество ярусов программы и количество операторов в каждом ярусе, то есть фактически оценить уровень параллелизма написанной программы. Такая схема выполнения наиболее точно показывает работу ФПП

модели вычислений. Также отладка слоев уменьшает число шагов, требуемых для анализа программы.

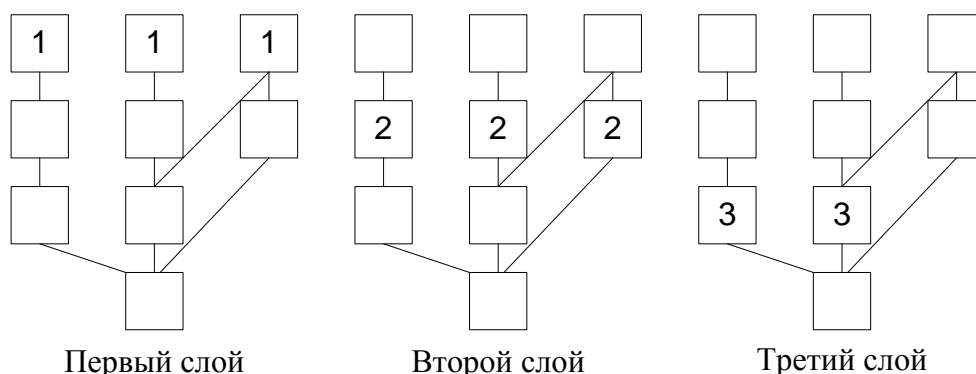


Рисунок 2 - Пример послойного выполнения операторов функционально-поточковой параллельной программы

Режим отладки ветвей (рисунок 3) предполагает, что на каждом шаге отладки для выполнения выбираются все операторы независимой ветви информационного графа. Такие операторы всегда выполняются в строгой последовательности, заданной графом программы. Это позволяет выделять в параллельной программе операторы, которые могут быть выполнены только один за другим, но при этом не нуждаются в получении каких либо других входных значений из других вершин. Данный режим выполняет односвязную последовательность команд за один шаг, что в определенных случаях может сократить время проверки. Выполнение логически связанных операторов облегчает восприятие отладочной информации.

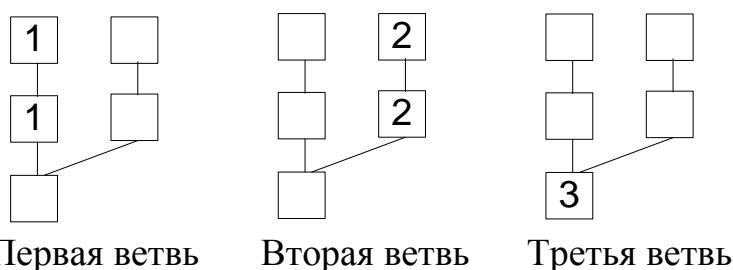


Рисунок 3 - Использование ветвей при отладке операторов функционально-поточковой параллельной программы

Режим проверки формул (рисунок 4) позволяет пользователю закрепить за произвольными узлами-операторами собственные утверждения (являющиеся выражениями на языке программирования), использующие в качестве значений аргумент функции или значения, вычисленные любым

узлом-оператором графа. Если отлаживаемая функция не является рекурсивной, отладка выполняется за один шаг, на котором вычисляются все узлы графа и все дополнительные утверждения.

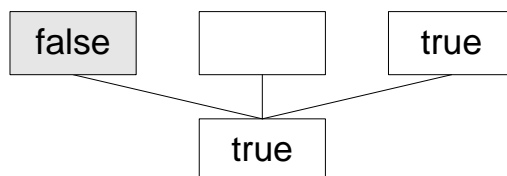


Рисунок 4 - Разметка вершин при отладке в режиме проверки формул

Вершины графа, содержащие утверждения, помечаются как истинные, если выражения, введенные пользователем, возвращают истину, или как ложные, если хотя бы одно из выражений не равно истине. Для каждой такой вершины вместе со значением соответствующего оператора программы можно увидеть вычисленные значения утверждений. Если отлаживаемая функция является рекурсивной, то число шагов отладки совпадает с числом итераций рекурсивной функции, а пользовательские формулы повторно проверяются на каждой итерации.

Выражения, записанные пользователем, задают требования к различным частям программы. Режим проверки формул позволяет специфицировать программу, определить, соответствует ли выполнение программы требованиям пользователя при конкретных начальных данных, или вычислить интересные для пользователя выражения, не внося изменений в саму программу.

**В четвертой главе** предложены методы верификации ФПП программ:

- метод верификации при наличии асинхронных списков, основанный на переборе возможных путей выполнения программы;
- метод формальной верификации, основанный на спецификации входных данных программы с помощью конкретных значений и дополнительных предложенных формул и добавлении условий к вычислениям программы на ее информационном графе.

Проверка ФПП программ с асинхронными списками основана на вычислении операторов программы и подстановке в них конкретных

значений. Тогда как формальная верификация со спецификацией пользователя на графе позволяет делать заключения о вычислениях программы над данными, заданными в обобщенном виде.

Для верификации применяется спецификация входных данных, описывающая значения посредством стандартных данных (таких как списки, строки и числа), а также специальных утверждений и констант. Множество утверждений спецификации выглядит следующим образом:

~unknownnumber – неизвестное число.

~unknownbool – неизвестное логическое значение (ложь или истина).

~unknown – неизвестное значение, может быть числом, текстом, логическим значением, списком, строкой или другим данным программы.

~gt A – число большее чем указанное число A.

~lt A – число меньшее чем указанное число A.

~ge A – число больше либо равно указанному числу A.

~le A – число меньше либо равно указанному числу A.

~A interval B – число, лежащее в указанном интервале [A, B].

Разработанный метод верификации применяет формулы интервального анализа и набор правил, прописанный для каждого оператора ФПП языка, позволяющий для известных входных данных оператора, среди которых есть хотя бы одно, заданное утверждением спецификации, поставить в соответствие результат работы оператора. Выходное значение оператора может быть известным данным, конкретным числом, строкой, списком или также являться утверждением спецификации, может являться комбинацией известных значений и утверждений спецификации. Набор правил не охватывает все возможные варианты, если во время верификации при рассмотрении оператора отсутствует правило для вывода его результата работы, результатом по умолчанию является константа спецификации ~unknown. При возникновении такой ситуации в ходе верификации, управление передается человеку. Пользователю предлагается самостоятельно ввести результат выполнения оператора над входными данными. Отказ от

ввода означает, что результатом становится ~unknown, если же пользователь ввел результат, он принимается истинным и используется в дальнейших вычислениях.

В разработанной среде для создания, отладки и верификации программ, интерфейс верификации почти совпадает с интерфейсом отладки. Результаты верификации предоставляются пользователю в том же виде, что и результаты отладки, верификация происходит, как и отладка, пошагово.

Если для верификации был выбран режим «Проверка формул», то появляется возможность, кроме спецификации входных данных функции добавить произвольное число условий к произвольным операторам функции. Условия записываются как выражения на языке ФПП программирования Пифагор, в них можно использовать формулы спецификации, а также специальные константы, доступные только в этом режиме: ARG – аргумент функции, NODE – значение той вершины-оператора графа функции, к которой добавлено пользовательское условие, NODE <натуральное число> - значение оператора с указанным номером.

При определении результата пользовательских выражений (условий) используется аналогичный подход, что и для вычисления самих операторов. То есть, если формулы спецификации отсутствуют в выражении, результат вычисляется интерпретатором, иначе происходит поиск в наборе правил. Если поиск не дает результата, то происходит запрос к пользователю.

Режим формальной верификации дает возможность проследить выполнение программы над обобщенными входными данными, заданными посредством формул спецификации, получить результаты работы каждого оператора, если они есть в наборе правил, а также результат, вычисляемый программой. Анализ процесса и результата выполнения программы над обобщенными начальными данными производится самим пользователем или отслеживается с помощью дополнительных пользовательских условий, приписанных к произвольным операторам программы.

Разработанные методы верификации позволяют:



- определить, способно ли наличие асинхронных списков привести к вычислению различных результатов при различных запусках программы;
- получить результаты работы операторов и предупреждения о возможных ошибках над обобщенными данными;
- подтвердить или опровергнуть соответствие вычислений программы спецификации пользователя для описанного множества входных данных, то есть установить корректность программы;
- получить интервальные оценки результатов работы операторов и выявить возможности выхода вычислений за границы типов данных.

**В пятой главе** представлена архитектура среды разработки, отладки и верификации ФПП программ. Описаны принципы работы транслятора, отладчика и верификатора, а также словесное описание их базовых алгоритмов, представлен интерфейс разработанной среды.

Среда разработки является единым приложением с графическим интерфейсом, способным порождать процессы трансляции, интерпретации и обрабатывать результаты их выполнения. Она включает следующие программные блоки.

1 Транслятор текста программы во внутренние представления, используемые отладчиком.

2 Отладчик ФПП программ получает внутреннее представление программы от транслятора и набор опций, заданный пользователем через графический интерфейс, регулирующие включение различных режимов отладки. Отладчик выполняет операторы программы с помощью запросов к интерпретатору или верификатору. Порядок обхода операторов и форма взаимодействия с пользователем определяются опциями отладки.

3 Верификатор ФПП программ получает от отладчика оператор и его входные значения, содержащие формулы спецификации (возможно наряду с точными значениями), осуществляет поиск результата работы оператора в наборе правил и возвращает найденное значение или константу спецификации ~unknown, если поиск не дал результата. Запуск верификатора

инициируется отладчиком при появлении формул спецификации во входных данных рассматриваемых операторов.

4 Интерпретатор получает от отладчика оператор и его аргументы как точные значения и выдает результат работы оператора.

Разработанная инструментальная среда позволила автоматизировать предложенные в разделах 3 и 4 методы отладки и верификации ФПП программ.

**В заключении** формулируются основные выводы по результатам исследования.

## ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1 Предложены и реализованы методы отладки ФПП программ, использующие различные формы обхода информационного графа программы с возможностью задания дополнительных условных выражений для проверки данных. Методы позволяют проводить детальный анализ процесса выполнения программы, уменьшать число шагов отладки, увеличивать эффективность локализации логических ошибок, отображать уровень параллелизма функций и синхронизацию ее операторов.

2 Разработаны способы применения методов индуктивных утверждений и индукции для функционально-поточковой модели параллельных вычислений с использованием информационного графа программы. На основе этого предложен метод формальной верификации функционально-поточковых параллельных программ, основанный на спецификации входных данных с использованием как конкретных значений, так и дополнительных условий, накладывающих ограничения на аргументы и промежуточные результаты. Это обеспечивает автоматизированную проверку программы для обобщенного множества входных данных и позволяет установить соответствие вычислений, выполняемых программой, спецификациям разработчика. Показано, что в ряде случаев возможно автоматическое определение свойства корректности программы или получение интервальной оценки вычисляемых числовых значений.

3 Предложен метод верификации функционально-поточковых параллельных программ, позволяющий установить, приводит ли произвольное появление данных внутри асинхронных списков к различным результатам вычислений.

4 Реализована среда разработки, обеспечивающая отладку и верификацию функционально-поточковых параллельных программ с использованием предложенных методов.

### **Основные публикации автора по теме диссертации**

**Журналы из Перечня ВАК ведущих рецензируемых научных изданий для опубликования основных результатов диссертаций на соискание учёной степени доктора и кандидата наук:**

1. Удалова, Ю.В. Методы отладки и верификации функционально-поточковых параллельных программ / Ю.В. Удалова, А.И. Легалов, Н.Ю. Сиротинина // Журнал Сибирского федерального университета. Техника и технологии. - 2011. - N 2. - С. 213-224.

2. Удалова, Ю.В. Отладка программ на функционально-поточковом параллельном языке Пифагор с подстановкой интервальных значений / Ю.В. Удалова, А.И. Легалов, Н.Ю. Сиротинина // Алтайский государственный технический университет им. И.И. Ползунова, Ползуновский вестник. - 2013. - N 2. - С. 46-48.

3. Удалова, Ю.В. Верификация функционально-поточковых параллельных программ методом индуктивных утверждений / Ю.В. Удалова, А.И. Легалов // Доклады Академии наук высшей школы Российской Федерации. - 2014. - N 2-3(23-24). - С. 125-132.

### **Публикации в рецензируемых изданиях:**

4. Удалова, Ю.В. Средства отладки функционально-поточковых параллельных программ / Ю.В. Удалова, А.И. Легалов, Н.Ю. Сиротинина // Доклады Академии наук высшей школы Российской Федерации. - 2008. - N 1(10). - С. 96-106.

### **Материалы конференций:**

5. Удалова, Ю.В. О верификации функционально-поточковых параллельных программ на примере задачи разделения множеств / Ю.В. Удалова // Распределенные и кластерные вычисления. - Красноярск, 2006. - С. 168-182.

6. Удалова, Ю.В. Отладка функционально-поточковых параллельных программ / Ю.В. Удалова // Информатика и информационные технологии. - Красноярск, 2007. - С. 43-46.

7. Удалова, Ю.В. Режимы отладки функционально-поточковых параллельных программ / Ю.В. Удалова // Четвертая Российско-германская школа по параллельным вычислениям, Научная сессия, Тезисы докладов. - Новосибирск, 2007. - С. 31-33.

8. Удалова, Ю.В. Реализация отладки функционально-поточковых параллельных программ / Ю.В. Удалова, Н.Ю. Сиротинина // Молодежь и наука – третье тысячелетие: Сб. материалов Всероссийской научной конференции студентов, аспирантов и молодых ученых. - Красноярск, 2007. - Часть 1. - С. 51-55.

9. Удалова, Ю.В. Методы отладки функционально-поточковых программ / Ю.В. Удалова // Четвертая Сибирская школа-семинар по параллельным и высокопроизводительным вычислениям. - Томск: Дельтаплан, 2008. - С. 176-184.

10. Удалова, Ю.В. Отладка и верификация функционально-поточковых параллельных программ / Ю.В. Удалова // Молодежь и наука – третье тысячелетие: Сб. материалов Всероссийской научной конференции студентов, аспирантов и молодых ученых. - Красноярск, 2008. - Часть 2. - С. 8-12.

11. Удалова, Ю.В. Об отладке и верификации функционально-поточковых параллельных программ [Электронный ресурс] / Ю.В. Удалова, А.И. Легалов, Н.Ю. Сиротинина, М.С. Кропачева // Параллельные вычислительные технологии (ПаВТ'2009): Труды международной научной

конференции (Нижний Новгород, 30 марта - 3 апреля 2009 г.). - Челябинск: Изд. ЮУрГУ, 2009. - 1 электрон, опт. диск (CD-ROM). - С. 757-764.

12. Удалова, Ю.В. Об отладке и верификации программ на функционально-поточковом параллельном языке Пифагор / Ю.В. Удалова, М.С. Кропачева // Молодежь и наука – третье тысячелетие: Сб. материалов Всероссийской научной конференции студентов, аспирантов и молодых ученых. - Красноярск, 2009. - Часть 2. - С. 40-44.

13. Удалова, Ю.В. Отладка функционально-поточковых параллельных программ / Ю.В. Удалова, Н.Ю. Сиротина, М.С. Кропачева // Проблемы информатизации региона: Материалы XI Всероссийской научно-практической конференции. - Красноярск, 2009. - С. 315-318.

14. Удалова, Ю.В. Верификация и спецификация программ на функционально-поточковом параллельном языке Пифагор / Ю.В. Удалова // Международная Ершовская конференция по информатике, рабочий семинар «Наукоемкое программное обеспечение». - Новосибирск, 2011. - С. 259-264.

15. Удалова, Ю.В. Отладка функционально-поточковых параллельных программ с подстановкой интервальных значений [Электронный ресурс] / Ю.В. Удалова // Молодежь и наука: сборник материалов VIII Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых, посвященной 155-летию со дня рождения К.Э.Циолковского. - Красноярск, 2012. - Режим доступа: <http://conf.sfu-kras.ru/sites/mn2012/thesis/s012/s012-014.pdf>

16. Удалова, Ю.В. Отладка программ на функционально-поточковом параллельном языке с подстановкой интервальных значений и пользовательских формул [Электронный ресурс] / Ю.В. Удалова // Молодежь и наука: сборник материалов IX Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых с международным участием, посвященной 385-летию со дня основания г. Красноярска. - Красноярск, 2013. - Режим доступа: <http://conf.sfu-kras.ru/sites/mn2013/thesis/s044/s044-053.pdf>

17. Удалова, Ю.В. Отладка функционально-поточковых параллельных программ с подстановкой интервальных значений и пользовательских формул / Ю.В. Удалова, А.И. Легалов, Н.Ю. Сиротинина // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов. Сборник статей всероссийской научно-практической конференции. - Барнаул, 2013. - С. 157-161.

18. Легалов, А.И. Технологические аспекты создания, преобразования и выполнения функционально-поточковых параллельных программ / А.И. Легалов, И.В. Матковский, М.С. Кропачева, Ю.В. Удалова, В.М. Васильев // Научный сервис в сети Интернет: все грани параллелизма: Труды Международной суперкомпьютерной конференции (23-28 сентября 2013 г., Новороссийск). - М.: Изд-во МГУ, 2013. - С. 443-447.

**Свидетельства о государственной регистрации программы для ЭВМ:**

19. Графический отладчик функционально-поточковых параллельных программ на языке Пифагор под ОС Linux [Текст] : пат. 2012612190 Рос. Федерация / А.И. Легалов, О.В. Непомнящий, Ю.В. Удалова, В.А. Хабаров ; заявитель и патентообладатель Федеральное государственное учреждение высшего профессионального образования "Сибирский федеральный университет" (СФУ) ; заявл. 28.12.2011 ; опубл. 28.02.2012

20. Верификатор функционально-поточковых параллельных программ на языке Пифагор под ОС Linux [Текст] : пат. 2013660557 Рос. Федерация / А.И. Легалов, Ю.В. Удалова ; заявитель и патентообладатель Федеральное государственное учреждение высшего профессионального образования "Сибирский федеральный университет" (СФУ) ; заявл. 18.11.2013 ; опубл. 15.01.2014