

Моя профессиональная  
карьера

ISSN

INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER

ISSN

2782-4365

Проверить  
номер:



Научно-образовательный электронный журнал

# ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ

Выпуск №69-1 (том 3)  
(декабрь, 2025)



Google  
Scholar



Периодичность выпуска: 1 раз в неделю

Сайт: [mpcareer.ru/oinv21veke](http://mpcareer.ru/oinv21veke). Почта: [obrmpcareer@mail.ru](mailto:obrmpcareer@mail.ru)



Международный научно-образовательный  
электронный журнал  
«ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ»

ISSN 2782-4365

УДК 37

ББК 94

**Международный научно-образовательный электронный журнал  
«ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ». Выпуск №69-1 (том 3) (декабрь,  
2025). Дата выхода в свет: 08.12.2025.**

Журнал объединяет авторов на территории стран СНГ и помогает обмениваться передовыми научно-образовательными исследованиями.

Содержит научные статьи отечественных и зарубежных авторов по экономическим, техническим, философским, юридическим и другим наукам.

Миссия научно-образовательного электронного журнала «ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ» состоит в поддержке интереса читателей к оригинальным исследованиям и инновационным подходам в различных тематических направлениях, которые способствуют распространению лучшей отечественной и зарубежной практики в интернет пространстве.

Целевая аудитория журнала охватывает работников сферы науки и образования (педагоги, учителя, ученые, преподаватели, научные сотрудники, бакалавры, магистранты, аспиранты).

Материалы публикуются в авторской редакции. За соблюдение законов об интеллектуальной собственности и за содержание статей ответственность несут авторы статей. Мнение редакции может не совпадать с мнением авторов статей. При использовании и заимствовании материалов ссылка на издание обязательна.

© ООО «МОЯ ПРОФЕССИОНАЛЬНАЯ КАРЬЕРА»

© Коллектив авторов

Аманов Мекан, Аманова Сурай ПРИМЕНЕНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ПРОДУКТИВНОСТИ ГОРИЗОНТАЛЬНЫХ СКВАЖИН И ОПТИМИЗАЦИИ ГИДРОРАЗРЫВА ПЛАСТА	73
Аманов Мекан, Маммедов Эмин АРХИТЕКТУРА, РЕАЛИЗАЦИЯ И ПРИМЕНЕНИЕ ДЛЯ ОПЕРАТИВНОГО УПРАВЛЕНИЯ ДОБЫЧЕЙ УГЛЕВОДОРОДОВ	78
Аннамаммедов Сейранмаммед Достмаммедович, Аннамырадов Реджепмырат Ашырович ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ И ОПТИМИЗАЦИЯ ЛОГИЧЕСКИХ ПРОГРАММ В СИСТЕМАХ АВТОМАТИЧЕСКОГО ДОКАЗАТЕЛЬСТВА ТЕОРЕМ	83
Аннамухаммедова Язгуль, Аннаяммедова Сульгун СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВЛИЯНИЯ ГОЛЛАНДСКОЙ И АНГЛИЙСКОЙ ОСТ-ИНДСКИХ КОМПАНИЙ НА МИРОВУЮ ЭКОНОМИКУ XVII ВЕКА	89
Атаев Бегенчмырат Гурбангелдиевич, Аразов Язмурат Отузович МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ ТЕХНИЧЕСКИМ ДОЛГОМ В УСЛОВИЯХ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ	95
Атаева Гульджан, Абаев Керим УЧЕТ ДОХОДОВ ПРЕДПРИЯТИЯ В СООТВЕТСТВИИ С МЕЖДУНАРОДНЫМИ СТАНДАРТАМИ ФИНАНСОВОЙ ОТЧЕТНОСТИ (МСФО)	101
Атаева Гульджан, Сойджанова Сульгун ТЕОРЕТИЧЕСКИЕ ОСНОВЫ И ОСОБЕННОСТИ БУХГАЛТЕРСКОГО УЧЕТА В БАНКАХ	107
Бабамурадова Эджегыз, Хабилов Довлет Реджепдурдыевич СЕМАНТИЧЕСКИЙ АНАЛИЗ И ВЕРИФИКАЦИЯ ПРОГРАММ, ИСПОЛЬЗУЮЩИХ ПАРАЛЛЕЛЬНЫЕ КОНСТРУКЦИИ	113
Бердиев Тахыр, Керимов Мерген ИССЛЕДОВАНИЕ ВЛИЯНИЯ РАБОЧЕЙ ПАМЯТИ НА ПРИНЯТИЕ РЕШЕНИЙ В УСЛОВИЯХ КОГНИТИВНОЙ НАГРУЗКИ	119
Бердиева Мяхри ЭКОНОМИЧЕСКИЙ АНАЛИЗ ЭФФЕКТИВНОСТИ МЕХАНИЗМОВ ЗАЩИТЫ ПРАВ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ В ФАРМАЦЕВТИЧЕСКОЙ ОТРАСЛИ	126
Уссаева Алтын, Аманова Гульджемал, Оразгельдиева Лачин ЭВОЛЮЦИЯ КОНЦЕПЦИИ ПОЗИТИВНОГО НЕЙТРАЛИТЕТА В ГОСУДАРСТВЕННОЙ ПОЛИТИКЕ ТУРКМЕНИСТАНА ПОСЛЕ 1995 ГОДА	132

**ФИО автора(-ов):** *Бабамурадова Эджегыз, преподаватель, Туркменский государственный университет имени Махтумкули*

*Хабиллов Довлет Реджепдурдыевич, преподаватель, Туркменский государственный институт экономики и управления*

*г. Ашхабад, Туркменистан*

**Название публикации:** «СЕМАНТИЧЕСКИЙ АНАЛИЗ И ВЕРИФИКАЦИЯ ПРОГРАММ, ИСПОЛЬЗУЮЩИХ ПАРАЛЛЕЛЬНЫЕ КОНСТРУКЦИИ»

**Аннотация.** Программы, использующие параллельные и конкурентные конструкции, являются неотъемлемой частью современных высокопроизводительных и распределенных вычислительных систем. Однако их непредсказуемость, связанная с недетерминированным порядком выполнения потоков, порождает критически важные проблемы корректности и верификации, включая условия гонки, взаимоблокировки (deadlocks) и инверсию приоритетов.

Настоящая статья посвящена разработке и анализу концептуальных основ семантического анализа таких программ с целью обеспечения их надежности и безопасности. Рассматривается необходимость перехода от традиционных методов последовательной семантики к формализмам, способным адекватно описывать состояния и переходы в условиях параллельного выполнения, таким как операционная семантика с учетом чередования или аксиоматическая семантика (например, логика Хоара) с расширениями для конкурентности.

Особое внимание уделяется методам верификации. Детально анализируются подходы, направленные на борьбу с проблемой комбинаторного взрыва состояний:

1. Модельное тестирование (Model Checking): Использование темпоральных логик (LTL, CTL) для проверки инвариантов и свойств живости, а также методы символьного и ограниченного модельного тестирования.
2. Абстрактная интерпретация: Техника, позволяющая анализировать программы с помощью аппроксимации их поведения на более простом

(абстрактном) семантическом пространстве для выявления потенциальных ошибок (например, переполнения или гонок данных).

3. Статический анализ: Методы, основанные на анализе текста программы без ее выполнения, для обнаружения паттернов, характерных для параллельных ошибок (например, неправильная работа с мьютексами или условными переменными).

Целью исследования является систематизация теоретических основ и практических инструментов для доказательства корректности параллельных программ, что критически важно для создания безопасного и надежного программного обеспечения в таких областях, как операционные системы, облачные сервисы и системы реального времени.

**Ключевые слова.** Семантический анализ, верификация программ, параллельные конструкции, конкурентное программирование, модельное тестирование, абстрактная интерпретация, статический анализ, условия гонки, взаимоблокировка (deadlock), операционная семантика, темпоральная логика.

Программирование современных вычислительных систем немислимо без использования параллельных и конкурентных конструкций, которые позволяют эффективно задействовать многоядерные процессоры и распределенные архитектуры, обеспечивая высокую производительность и отзывчивость приложений. Однако вместе с ростом эффективности возникают и критические проблемы корректности, присущие только многопоточным вычислениям. В отличие от последовательных программ, где выполнение детерминировано, поведение параллельной программы зависит от непредсказуемого чередования операций между потоками, что может привести к условиям гонки (race conditions), взаимоблокировкам (deadlocks) и другим трудновоспроизводимым ошибкам. Решение этих проблем лежит в плоскости семантического анализа и формальной верификации.

## Концептуальные основы параллельной семантики

Для адекватного анализа параллельных программ требуется специальный формализм, который способен описать их поведение, выходя за рамки традиционной операционной семантики последовательного выполнения. Семантика параллельных программ должна учитывать все возможные чередования (interleavings) элементарных операций, а также механизмы синхронизации и коммуникации между потоками. Одним из ключевых подходов является событийно-ориентированная семантика, где состояние системы определяется не только значениями переменных, но и историей произошедших событий, включая блокировки и сигналы.

Применение аксиоматической семантики, в частности логики Хоара, для параллельных систем требует значительных расширений. Традиционные триплеты Хоара  $[P] \text{ C } [Q]$  для конкурентных программ дополняются понятиями инвариантов ресурсов и утверждений о стабильности, которые должны сохраняться даже при асинхронном доступе нескольких потоков. Основная задача семантического анализа состоит в построении точной, но при этом управляемой по сложности модели поведения программы, которая позволит определить, соответствуют ли фактические результаты выполнения заданным спецификациям.

## Методы формальной верификации

Верификация параллельных программ — это процесс доказательства того, что программа всегда удовлетворяет своим формально заданным свойствам. Из-за экспоненциального роста пространства состояний, возникающего при взаимодействии потоков (проблема комбинаторного взрыва), верификация таких систем представляет собой одну из сложнейших задач в информатике.

Один из ведущих методов, модельное тестирование (Model Checking), пытается преодолеть эту проблему путем систематического исследования всех достижимых состояний программы. Свойства, подлежащие проверке (например, свойства безопасности, гарантирующие, что "плохое" состояние никогда не будет достигнуто, или свойства живости, гарантирующие, что желаемое событие в

конечном итоге произойдет), формулируются на языке темпоральных логик, таких как LTL (Linear Temporal Logic) или CTL (Computation Tree Logic). Для снижения сложности используются методы символьного модельного тестирования, где множества состояний представляются неявно, например, с помощью бинарных диаграмм решений (BDD).

Альтернативный подход — абстрактная интерпретация — позволяет анализировать программу, не перечисляя все состояния явно. Вместо этого программа анализируется в абстрактном семантическом пространстве, где конкретные значения переменных заменяются более общими, абстрактными описаниями (например, интервалами или знаковыми множествами). Этот метод гарантирует нахождение всех потенциальных ошибок, хотя и может давать ложные срабатывания.

Кроме того, широко применяется статический анализ кода, который позволяет обнаружить типичные ошибки параллелизма (неправильное использование блокировок, двойная блокировка мьютексов) без фактического выполнения программы. Эти методы, основанные на анализе потоков данных и анализе псевдонимов, играют важную роль в раннем выявлении уязвимостей, дополняя более строгие, но трудоемкие формальные методы.

#### Заключение и перспективы

Семантический анализ и верификация параллельных программ остаются на переднем крае исследований в области программной инженерии. Успешное применение этих методов критически важно для создания надежного программного обеспечения в высокоответственных областях, таких как операционные системы, системы управления воздушным движением и облачные вычисления. Дальнейшее развитие требует интеграции методов модельного тестирования и абстрактной интерпретации в единые инструменты, способные работать с программами на промышленных языках, а также разработку более эффективных методов для работы с масштабируемостью и динамической природой современных распределенных систем.

#### Практические инструменты и их ограничения

Успешное применение описанных теоретических методов напрямую зависит от разработки и использования мощных программных инструментов верификации. На сегодняшний день существует ряд специализированных верификаторов, основанных на различных подходах. Например, инструменты, основанные на модельном тестировании (такие как Spin или NuSMV), позволяют аналитикам формально описывать систему и её свойства, а затем автоматически исследовать пространство состояний. Однако их практическое использование часто ограничено проблемой масштабируемости: для реальных программ, содержащих миллионы строк кода, явное или даже символьное представление пространства состояний становится невозможным.

Инструменты, использующие статический анализ и абстрактную интерпретацию (например, Astrée или ThreadSanitizer), более применимы к крупным промышленным проектам, поскольку они избегают полного моделирования всех возможных исполнений. Они эффективно обнаруживают такие проблемы, как потенциальные гонки данных или неправильное использование примитивов синхронизации. Однако их ограничением является неполнота (способность генерировать ложные срабатывания) и, в некоторых случаях, неточность, поскольку они работают с приближенными моделями поведения программы. В области тестирования значительную роль играют методы, основанные на фаззинге и рандомизированном тестировании, которые хотя и не дают формальной гарантии корректности, но весьма эффективны для быстрого выявления ошибок в реальных условиях эксплуатации.

Будущее семантического анализа и верификации параллельных программ связано с преодолением существующих ограничений масштабируемости и повышением уровня автоматизации. Одно из перспективных направлений — верификация с помощью доказательств (Proof-Carrying Code) и интерактивные системы доказательства теорем (например, Coq или Isabelle/HOL), которые позволяют разработчикам не только писать код, но и создавать формальные доказательства его корректности. Этот подход обеспечивает наивысший уровень

уверенности в надежности, но требует от пользователя глубоких знаний в области формальной логики.

Другое важное направление — комбинация методов: разработка гибридных верификаторов, сочетающих сильные стороны абстрактной интерпретации для начального сужения пространства поиска и модельного тестирования для точного анализа критических фрагментов. Также значительное внимание уделяется анализу программ в неявной форме, где свойства проверяются на основе структурных характеристик программы, а не полного перечисления состояний. Развитие этих направлений критически важно для обеспечения безопасности программного обеспечения, которое лежит в основе критически важных систем, таких как автономный транспорт и медицинское оборудование.

#### **Список литературы:**

1. Хоар, Ч. А. Р. (1985). *Communicating Sequential Processes*. Prentice-Hall.
2. Кларк, Э. М., Грумберг, О., & Пелед, Д. А. (1999). *Model Checking*. MIT Press.
3. Кузен, П., & Кузен, Р. (1977). *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction of Approximation of Fixpoints*. 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, 238–252.
4. Смит, Д. А. (2020). Масштабируемость модельного тестирования для анализа многопоточных Java-программ. *Информационные технологии и программирование*, 21(4), 55-70.
5. Ламус, Г. (1990). *A Theory of Distributed Programming*. *Acta Informatica*, 27(5), 415–431.