Верификация программ

Андрей Михайлович Миронов

Кафедра Математической Теории Интеллектуальных Систем
http://intsys.msu.ru/staff/mironov/

amironov66@gmail.com

Проблема создания высоконадежного программного обеспечения

1. Во время запуска ракеты Ariane-5 в июне 1996 года произошел взрыв ракеты спустя 40 сек. после старта (на разработку ракеты было затрачено около \$7 млрд.).
   Причина – неправильная программная реализация численных алгоритмов в системе управления.
2. В феврале 1991 года зенитный комплекс Patriot промахнулся мимо сбиваемой ракеты Scud, ущерб - 28 убитых, более 100 раненых.
   Причина – ошибки в программном обеспечении системы управления противовоздушной обороной.

Проблема создания высоконадежного программного обеспечения (продолжение)

Другие примеры:

1. ошибки в программе обработки заявлений доноров органов в Великобритании (у 25 человек взяли не те органы)
2. 88 критических ошибок в операционной системе Android Froyo (используемой в мобильных устройствах), приводящих к неавторизованному доступу к личным данным пользователей (в том числе - нарушению конфиденциальности электронной переписки, краже денег с банковских счетов, и т.д.).
3. Недавние авиакатастрофы Boeing 737 (сотни погибших).

Это лишь малая часть проблем, связанных с нарушением безопасности компьютерных систем.

Теория верификации программ

Гарантированное обоснование надежности программного обеспечения достигается только путем построения математических моделей программ и применения методов **верификации** программ.

Теория верификации программ - центральный раздел направления **компьютерная безопасность**.
Математические основы формальной верификации: методы, основанные на логиках высших порядков (HOL4, Isabelle/HOL, Coq), model checking, исчисления процессов, и др.

"Cryptography is only a tiny piece of the security puzzles, most systems break elsewhere

- incorrect requirements or specifications
- implementation errors"

Bart Preneel,
President of the International Association for Cryptologic Research (IACR)

Задача верификации программ

Верификация программы – математическое доказательство утверждения о том, что программа соответствует (или не соответствует) своей спецификации.

Верификация распределенных программ – требуется разработать методы верификации (т.е. доказательства корректности) программ, состоящих из нескольких взаимодействующих компонентов, такими программами могут быть

- программы, выполняющиеся на многопроцессорных и распределенных вычислительных системах,
- криптографические протоколы,
- смарт-контракты в блокчейновых системах,
- системы управления автономными транспортными средствами, коллективами роботов, беспилотными летающими аппаратами.

Примеры успешного применения верификации

В западно-европейской индустрии высоконадежного ПО, эти методы применяются весьма широко с 1990-х гг.:

- система управления в парижском метро (Meteor)
- системы управления во французской ядерной энергетике
- система управления Airbus A380, и т.д.

Российская формально верифицированная Система Распределенного Реестра InnoChain (2020-2021, разрабатывается участниками моей команды):

- Формально-верифицированная компиляция смарт-контрактов в машинный код (исключаются ошибки компилятора или виртуальной машины)
- Формально-верифицированные алгоритмы узла CPP исключаются ошибки run-time (например, переполнения памяти)
- Формально-верифицированный протокол консенсуса.

Верификация в финансовой математике

- Задачи формальной верификации для высокочастотных стратегий алгоритмической торговли на финансовых рынках.
- Интеграция методов формальной верификации
  - с методами стохастической финансовой математики, и
  - методами машинного обучения

  для управления портфелями финансовых инструментов и финансовыми рисками.

Математические проблемы, связанные с криптографическими протоколами

Разработка новых криптографических протоколов для

- аутентификации участников протоколов, электронной подписи, совместного подписания документов,
- анонимных коммуникаций,
- распределения криптографических ключей,
- передачи информации по квантовому каналу,
- разделения секрета,
- электронного голосования,
- электронной коммерции,
- анонимных аукционов,
- конфиденциальных вычислений в облачных вычислительных средах.

Спецификация и верификация смарт-контрактов

Смарт-контракты – это протоколы взаимодействия агентов в блокчейновых системах.

- Создание новых математических моделей смарт-контрактов и формальных языков для описания их свойств (корректности, безопасности, устойчивости, оптимальности и т.п.).
- Разработка новых математических моделей и методов верификации смарт-контрактов с потенциально неограниченным количеством участников

Большие данные и машинное обучение

- Автоматное обучение и процессное обучение, в классах
  - вероятностных автоматов,
  - нечетких автоматов,
  - распределенных алгоритмов

  Требуется построить алгоритм синтеза оптимальных автоматов (детерминированных, вероятностных, нечетких, автоматов над термами и т.п.) по частичной информации об их поведении.
- Анализ естественно-языковых текстов, кластеризация, вероятностно-тематическое моделирование.
- Построение агрегирующих алгоритмов в математической теории прогнозирования. Задача агрегирующего алгоритма — выработка предсказаний с учетом мнения экспертов. Требуется построить такие агрегирующие алгоритмы, качество предсказания которых отличается на небольшую величину (называемую регретом) от качества предсказания наилучшего эксперта.