

возможности ограничены. В связи с этим автором ведутся исследования по адаптации и применению специализированных языков (например, языка TreeDl [3]) для построения генераторов тестов. Ожидаемый результат от внедрения этого подхода заключается в появлении средства создания тестов, обладающего следующими свойствами:

- максимально возможная независимость генератора тестов от целевого ПО;
- минимальное время разработки тестов для нового целевого ПО;
- обеспечение повышения объема покрытия кода целевого ПО.

ЛИТЕРАТУРА

1. Макаров А. Н. Метод автоматизированного поиска программных ошибок в алгоритмах обработки сложноструктурированных данных // Прикладная дискретная математика. 2009. №3(5). С. 117–127.
2. Miller C., Peterson Z. N. J. Analysis of Mutation and Generation-Based Fuzzing. www.securityevaluators.com. 2007.
3. treedl.org — сайт проекта TreeDl.

УДК 519.7

ПРОГРАММНАЯ ТРАНСЛЯЦИЯ АЛГОРИТМОВ В ПРОПОЗИЦИОНАЛЬНУЮ ЛОГИКУ ПРИМЕНИТЕЛЬНО К КОМБИНАТОРНЫМ ЗАДАЧАМ

И. В. Отпущенников, А. А. Семёнов

Интенсивный рост производительности современных вычислительных архитектур сделал возможным решение комбинаторных задач таких размерностей, которые казались непреодолимыми еще 15–20 лет назад. В связи с этим возникли новые направления в компьютерной алгебре и вычислительных отраслях дискретной математики и математической логики. Одной из наиболее актуальных в этом смысле областей является решение логических (булевых) уравнений. В последние 10 лет наблюдается существенный прогресс в разработке алгоритмов, эффективных на практически важных классах логических уравнений.

К логическим уравнениям эффективно сводятся многочисленные комбинаторные задачи. Однако при практическом осуществлении соответствующих сведений приходится сталкиваться с различными препятствиями (разнородность структур данных; отсутствие идеологии, применимой к широкому классу проблем, и др.). Сказанное означает актуальность проблемы разработки многофункционального транслятора, осуществляющего сведение различных комбинаторных задач к задачам поиска решений логических уравнений. Следует отметить, что во многих имеющихся разработках данная проблема так или иначе решалась (см. [1–4]). Однако во всех этих случаях осуществлялась трансляция некоторого «узкого» класса проблем, ограниченного конкретной предметной областью. Комплекс Verilog [3] используется при описании логических схем, и все его базовые конструкции ориентированы на специалистов в схемотехнике. Комплекс ABC [4] предназначен для решения задач синтеза и верификации логических схем; схемы могут быть описаны в различных форматах, однако предполагается, что пользователь эти описания каким-либо образом получил (например, используя Verilog).

В работе излагаются общие принципы построения нового многофункционального транслятора Transalg. Данный программный комплекс позволяет осуществлять пропозициональное кодирование алгоритмов, не предполагая узкоспециальных знаний у

пользователя. Язык Transalg — это фактически самостоятельный язык с С-подобным синтаксисом. Семантика языка определяет правила построения логических уравнений по соответствующему описанию алгоритма. Ниже приведены основные этапы трансляции алгоритмов в комплексе Transalg.

- 1) Анализ текста программы и построение дерева синтаксического разбора.
- 2) Обход построенного дерева, выполнение семантических действий: «пропозициональная свертка» условных переходов (см. [5]) за счет преобразований Цейтина, построение систем логических уравнений.
- 3) Приведение систем логических уравнений к нормальным формам; преобразование в формат И-НЕ графов.

Пропозициональные кодировки многих алгоритмов, реализованные в трансляторе Transalg, по объему существенно меньше известных аналогов (например, пропозициональная кодировка алгоритма DES компактнее кодировки, представленной в [1], более чем в два раза).

Дополнительной возможностью комплекса Transalg является сведение к SAT (или к системе логических уравнений в любой нормальной форме) задач 0-1-целочисленного линейного программирования (0-1-ЦЛП). Получаемые кодировки по компактности также превосходят известные аналоги [6].

ЛИТЕРАТУРА

1. *Massacci F., Marraro L.* Logical Cryptanalysis as a SAT Problem // J. Automated Reasoning. 2000. V. 24. No. 1–2. P. 165–203.
2. *Буранов Е. В.* Программная трансляция процедур логического криптоанализа симметричных шифров // Вестник Томского госуниверситета. Приложение. 2004. № 9(1). С. 60–65.
3. <http://www.veripool.org/>
4. Berkeley Logic Synthesis and Verification Group, ABC: A System for Sequential Synthesis and Verification. <http://www.eecs.berkeley.edu/~alanmi/abc/>.
5. *Семёнов А. А.* Трансляция алгоритмов вычисления дискретных функций в выражения пропозициональной логики // Прикладные алгоритмы в дискретном анализе. Сер. Дискретный анализ и информатика. Иркутск: Изд-во ИГУ, 2008. Вып. 2. С. 70–98.
6. *Een N., Sorensson N.* Translating Pseudo-Boolean Constraints into SAT // J. Satisfiability, Boolean Model. Computat. 2006. No. 2. P. 1–25.

УДК 519.682

О ПРЕДСТАВЛЕНИИ КОНТЕКСТНО-СВОБОДНЫХ ЯЗЫКОВ ДИАГОНАЛЯМИ ЛИНЕЙНЫХ ЯЗЫКОВ

К. В. Сафонов, Д. А. Калугин-Балашов

В теории контекстно-свободных языков (кс-языков) и грамматик словарь языка $X = \{x_1, \dots, x_n\}$ обычно называют терминальным множеством, тогда как нетерминальным называют конечное множество $Z = \{z_1, \dots, z_m\}$ вспомогательных символов, необходимых для задания грамматических правил (грамматики) данного языка. Для элементов этих множеств определены операции конкатенации и формальной суммы, приводящие к мономам и многочленам.

Если последние построены в соответствии с грамматическими правилами данного кс-языка, то эти мономы и многочлены интерпретируются как его предложения и совокупности предложений. Рассмотрение совокупности всех грамматически правиль-