

Опубликовано в трудах II межвузовской конференции молодых учёных. – СПб.: СПбГУ ИТМО, 2004.– с. 36–40.

Методы тестирования решений задач на соревнованиях по программированию

Корнеев Г.А., Станкевич А.С.

Научный руководитель Парфенов В.Г.

В работе рассматривается классификация задач с точки зрения автоматического тестирования и предлагается новый метод автоматического тестирования программ на соревнованиях по программированию.

Введение

В настоящее время широкое распространение получили различные соревнования по программированию, например, этапы Командного студенческого чемпионата мира по программированию (АСМ ICPC), Всероссийские командные олимпиады школьников по программированию. Кроме того, указанные задачи встречаются в современных технологиях дистанционного обучения, в частности, в проекте Интернет–школы программирования (Internet Programming School, IPS), реализуемом на кафедре компьютерных технологий СПбГУ ИТМО [4].

В последнее время, все большую популярность приобретают соревнования реального времени, вытесняющие классические соревнования с отложенной проверкой. Но соревнования такого рода требуют значительно более высокого уровня организации проверки решений участников, так как время, выделяемое на тестирование решения, измеряется минутами. Напротив, в соревнованиях с отложенной проверкой время тестирования решения практически не ограничено.

В данной статье описывается новый подход к тестированию задач, основанный на игровой стратегии. Это означает, что вместо традиционного

тестирования на заранее определенном наборе тестов наша система “играет” с проверяемой программой, и по результатам данной “игры” определяет общий результат. Данный метод позволяет адекватно оценивать задачи, для которых оптимальное решение практически не достижимо в силу высоких требований к вычислительным ресурсам. Кроме того, позволяет тестировать управляющие программы, когда автоматическая тестирующая система эмулирует управляемый объект и оценивает качество управления.

Описанный в данной статье метод тестирования был успешно апробирован при проведении реальных соревнований (четверть и полуфинала Командного студенческого чемпионата мира по программированию), а так же используется в тестирующем ядре интернет-школы программирования [4] и системе автоматического проведения интернет-соревнований [3]. Более подробно об этих проектах можно узнать в работе [1]. Так же краткое описание системы автоматического тестирования, основанной на описываемых методах может быть найдено найти в работе [2].

Автоматическое тестирование

Задача процедуры тестирования программного решения состоит в определении, решает ли тестируемая программа требуемую задачу.

В последнее время широкое применение находят методы, основанные на анализе исходного кода программ. Но для любого достаточно сложного языка программирования (эквивалентного машине Тьюринга), существуют программы, для которых невозможно эффективно определить, не только решают ли они заданную задачу, но даже, закончит ли программа выполнение за конечное время.

Так как, точное определение корректности программы невозможно, то широкое применение находят различные эмпирические методы тестирования. При этом в зависимости от структуры задачи используются различные методы тестирования.

Назовем метод тестирования *косвенным*, если для определения корректности ответа, получаемый на наборе входных данных, производится проверка принадлежности его некоторому, заранее вычисленному множеству.

Косвенные методы обычно применяются при тестировании задач с единственным решением, а так же задач, для которых известны эффективные методы получения всех возможных правильных ответов.

Недостаток косвенных методов состоит в том, что для проверки правильности результата мы должны уметь решать исходную задачу, или даже более сложную (в случае, когда решение исходной задачи не единственно).

Назовем метод тестирования *прямым*, если для проверки правильности результата нет необходимости иметь решение исходной задачи.

Классификация задач по отношению к автоматическому тестированию

Рассмотрим следующую классификацию задач:

- 1) По количеству правильных ответов.
 - А) Задачи с единственным ответом.
 - В) Задачи с конечным множеством правильных ответов.
 - С) Задачи с бесконечным множеством правильных ответов.
- 2) По существованию эффективного метода решения
 - А) Для задачи известен эффективный метод получения всех ответов.
 - В) Для задачи известен эффективный метод получения некоторого ответа.
 - С) Для задачи не известны эффективные методы решения.
- 3) По существованию эффективного метода проверки решения, не включающего решение исходной задачи.
 - А) Для задачи известен эффективный метод проверки, не требующий решения исходной задачи.
 - В) Такой метод не известен.

Будем обозначать типы задач трехбуквенными сокращениями, где каждая буква является значением соответствующего параметра. На пример, задача типа ВСА — имеет конечное множество решений, для которого не известны эффективные методы нахождения хотя бы одного элемента, но существует эффективный метод проверки принадлежности данному множеству.

Очевидно, что не все возможные 18 типов задач имеют смысл. Рассмотрим примеры задач для каждого типа.

ААА. Задача извлечения арифметического квадратного корня из целого числа, являющегося полным квадратом. Для данной задачи существует метод решения линейный по количеству цифр в исходном числе. Так как ответ единственен, данным методом мы получаем все возможные решения. Посредством возведения полученного решения в квадрат, и сравнения с исходным числом производится эффективная проверка правильности результата, не основанная на решении исходной задачи.

ААВ. Задача перемножения двух чисел. Очевидно, задача имеет единственное решение, которое можно получить эффективным методом. При этом единственным разумным методом проверки результата является перемножение исходных чисел.

АВА и АВВ. Так, как правильно решение единственно, и мы можем эффективно получить некоторое правильное решение, то мы имеем эффективный метод получения всех возможных решений. Таким образом, не существует задач таких типов.

АСА. Задача вычисления дискретного логарифма, т.е. нахождение такого K , что $B = A^K \bmod N$. Если N — простое число, то результат единственен. А для его проверки нужно просто возвести число A в K -ю степень по модулю N .

АСВ. Задача разложения числа на простые множители. Одним из этапов проверки решения является определение простоты числа, которая не может быть произведена эффективно.

ВАА. Представление натурального числа N в виде суммы двух различных натуральных слагаемых. Мощность множества решений равна $\lfloor N/2 \rfloor - 1$. Проверка производится простым сложением полученных слагаемых.

ВAB. Нахождение минимального пути в графе.

ВВА. Нахождение гамильтонова пути в плотном графе.

ВВВ. Нахождение минимального пути во взвешенном графе.

ВСА. Задача разложения составного числа на пару натуральных множителей, не равных единице. Задача имеет конечное множество правильных ответов, т.е. его мощность не превосходит некоторого числа. Но эффективного метода нахождения хотя бы одного решения не известно. При этом, если ответ найден, проверить его не составляет труда — необходимо просто перемножить полученные множители и сравнить результат с исходным числом.

ВСВ. Задача о назначениях.

САА и САВ, так как все элементы бесконечного множества, не могут быть получены, то зада данных типов не существуют.

СВА. Задача нахождения Пифагоровой тройки (натуральных чисел A , B и C , таких, что $A^2 + B^2 = C^2$), все числа которой больше заданного.

ССА. Поиск простого числа, больше заданного.

ССВ. Численное решение “Задачи трех тел”

По отношению к тестированию задачи разбиваются на три класса:

типы ААА, ААВ, ВАА, ВAB, АСВ допускают прямое тестирование.

типы ААА, АСА, ВАА, ВВА, ВСА, СВА, ССА, допускающие косвенное тестирование.

типы ВВВ, ВСВ, СВВ, ССВ Задачи, не допускающие не прямых не косвенных методов тестирования. Для задач этого класса не существует эффективного метода проверки корректности ответа.

При этом, путем наложения дополнительных условий на ответ, задачи типов ВВВ и ССВ зачастую удается преобразовать в задачи с единственным решением, что переводит их в класс задач, допускающих прямое тестирование.

Тестирование на фиксированном наборе тестов

При использовании данного метода тестирования программа считается корректной, если она выдает правильный результат на некотором конечном подмножестве множества всевозможных входных данных.

Данный метод имеет как прямой, так и косвенный вариант, в зависимости, от способа проверки ответа.

В прямом варианте полученный ответ проверяется с помощью соответствующей процедуры, а в косвенном ответ сравнивается со всеми правильными ответами для данного набора входных данных (обычно применяется в случае единственности ответа).

Тестирование на наборе случайных тестов

Программа считается корректной, если она выдает правильные ответы на заданном количестве случайных наборов входных данных.

Данный метод обычно используется как косвенный, в случаях, когда известен эффективный метод проверки правильности решения, но не нахождения решения как такового и может быть применен.

Тестирование посредством эмуляции

Программе на вход подается некоторая исходная ситуация, после чего она изменяется, в соответствии с полученным ответом и вновь подается на вход тестируемой программе. Если после ряда запусков программе удастся добиться требуемого результата, она считается корректной.

Обычно используется для программ управления или программ для игр (в математическом смысле этого слова), в которых не известна оптимальная стратегия. В первом случае, тестирующая программа эмулирует управляемый объект, во втором, “играет против” тестируемой программы.

Все приведенные методы тестирования могут быть реализованы в виде “диалога” тестирующей и тестируемой программы, что позволяет реализовать единую тестирующую систему, включающую их. Данный подход и был заложен, как базовый в автоматическую тестирующую систему PCMS2.

Игровая стратегия тестирования

На данный момент подавляющее большинство систем автоматического тестирования использует по тестовый подход. В рамках данного подхода тестируемая программа запускается на заранее определенном наборе тестов, после чего выданный ею результат проверяется. Существенным недостатком данного подхода является невозможность адаптации системы тестов под конкретное решение, для определения его сильных и слабых сторон.

При использовании данной стратегии, проверка решения рассматривается как игра двух противников: тестирующей системы и тестируемой программы. Первый ход осуществляет тестирующая система (проверяющая программа). Она подготавливает входные данные, для запуска тестируемой программы, а так же устанавливает ограничения на количество ресурсов, которые будут доступны ей. Далее ход переходит к тестируемой программе. Она запускается на входных данных, подготовленных тестирующей системой, и выдает некоторый результат. При этом осуществляется контроль за объемом использованных ресурсов. Ход вновь получает тестирующая система. Она анализирует результат, полученный тестируемой программой. На основе результатов анализа принимается решение о продолжении тестирования. Если тестирование продолжается, то проверяющая система подготавливает новый набор входных данных, при этом может быть использована информация, полученная от тестируемой программы, на предыдущих шагах. После чего ход снова передается тестируемой программе, и так далее.

Игровая стратегия тестирования позволяет повысить гибкость проверки решений и существенно расширяет круг задач, доступных для тестирования.

Очевидно, по тестовый подход является подмножеством предложенной стратегии тестирования. При этом, информация, полученная при запуске программы участника на одном тесте, ни как не используется для проверки других.

То, что, в свой ход, автоматическая система тестирования может производить произвольные действия (в том числе, и запуск внешних программ), позволяет без дополнительных механизмов осуществлять вероятностное тестирование. При этом, тестирующая система, создает случайный набор входных данных, удовлетворяющих условию задачи и передает ход тестируемой программе. Для проверки полученных результатов возможны следующие подходы:

- Тестирующая программа непосредственно проверяет правильность полученного ответа. Применяется при тестировании задач классов $??A$.
- Запускается правильное решение (решение жюри). Результаты, выданные тестируемой программой и правильным решением сравниваются. Применяется для тестирования задач с единственным правильным решением.
- Запускается генератор множества всех правильных решений. Результат, выданный тестируемой программой, проверяется на принадлежность сгенерированному множеству. Применяется для тестирования задач класса $?A?$, при не большой верхней границе мощности множества правильных ответов.
- Запускается решение жюри и полученный результат, сравнивается по оптимальности с результатом, выданным тестируемой программой. Данный подход используется для тестирования оптимизационных задач, для которых точный ответ получить не представляется возможным (на пример, задачи Коммивояжера для плотных графов с большим числом вершин).

Использование последнего из предложенных подходов, для тестирования на заранее заданном наборе тестов, позволяет осуществлять

детерминированное тестирование, задач, точное решение для которых не известно.

Дополнительной возможностью игровой стратегии тестирования является сравнительное тестирование игровых задач. При этом проверяющая программа действительно играет с тестируемой. В начале, задается исходная позиция в игре, после чего, одна из сторон делает первый ход и передает ход другой стороне, которая в свою очередь делает свой ход и т.д. При этом проверяющая программа проверяет корректность производимых ходов обеих сторон, и оценивает результаты игры.

Так же игровая стратегия тестирования позволяет осуществлять проверку программ управления. При этом, проверяющая система, в свой ход, проверяет корректность управляющих воздействий, предложенных тестируемой программой и рассчитывает отклик управляемой системы на произведенные воздействия, после чего подает полученный результат на вход тестируемой программе и т.д. Тестируемая программа, признается корректной, при достижении требуемого состояния управляемой системы.

Таким образом, в рамках игрового подхода к тестированию, может быть осуществлен двухуровневый подход к тестированию. На верхнем уровне используется заранее заданный набор тестов. На нижнем уровне производится несколько запусков тестируемой программы, с использованием схемы сравнительного тестирования или тестирования программ управления. На верхнем же уровне, оценивается результат произведенной последовательности запусков в целом (результат игры, состояние управляемой системы, и т.п.).

Описанные в данном разделе новые методы тестирования не могут быть реализованы в рамках классической по тестовой стратегии.

Список литературы

1. Казаков М. А. Разработка и внедрение системы поддерживающей новые технологии обучения программированию // Телекоммуникации и информатизация образования. — 2002 — №6, — с. 81-100.

2. Корнеев Г. А., Елизаров Р. А. Автоматическое тестирование решений на соревнованиях по программированию // Телекоммуникации и информатизация образования. — 2003. — №1. — с. 61-73.
3. Архив соревнований с системой автоматического проведения online-соревнований // <http://neerc.ifmo.ru/online>.
4. Интернет-школа Программирования // <http://ips.ifmo.ru>