

Обнаружение дефектов в процессе проектирования и эксплуатации программ

Соколова Э.С., Дмитриев Д.В.

Нижегородский государственный технический университет, Н. Новгород

В связи с применением средств вычислительной техники во всех областях человеческой деятельности актуальна проблема надежности программного обеспечения (ПО). Использование современных технологий в программировании направлено, в основном, на создание высококачественного программного обеспечения, характеризуемого временем исполнения, требуемым объемом оперативной памяти и т. д. Однако внедрение новых эффективных организационных и методологических правил проектирования программных комплексов не предотвращает появления ошибок при эксплуатации программ. Кроме того, очень часто современная практика создания программ базируется лишь на квалификации и интуиции специалистов, что приводит к наличию большого количества не выявленных при тестировании ошибок и низкому качеству программного продукта. В настоящее время достаточно внимания уделяется решению вопросов организации эффективного тестирования программ, появились работы в области оценки надежности и качества программных средств, разрабатываются теоретические основы диагностирования программных модулей [1-3]. Однако формализованный подход к решению задач диагностирования ограничивается, как правило, обнаружением факта существования дефектов программ, без локализации места их возникновения. Под термином дефект будем понимать программную ошибку, приводящую к несоответствию ожидаемых результатов выполнения программного продукта и фактически полученных результатов. После обнаружения дефекта необходимо подвергнуть его анализу с целью определения его природы – представляет ли он неисправность теста, ошибку в программном коде или является дефектом в проектных спецификациях.

Данная работа посвящена вопросам обнаружения и устранения ошибок в программных кодах. Предлагается формализованный подход к проблеме диагностирования структуры программных модулей, заключающийся в локализации кратных ошибок кода программного модуля, обнаруженных во время тестирования.

После принятия решения об устранении обнаруженных проблем, связанных с дефектами программного кода, разработчик вносит соответствующие изменения в программный код и проводит контрольное тестирование с целью подтверждения правильности работы исправленного кода. При этом довольно часто принимается априори решение о том, что дефект программы вызван существованием единственной ошибки, что на практике встречается довольно редко. Такой подход приводит к неверным результатам корректировки программ и значительно затрудняет поиск и устранение ошибок кода. Это связано с тем, что проявленный дефект работы программы может быть следствием не одной, а сразу нескольких ошибок. Кроме того, часто исправление одной из ошибок приводит к тому, что программа начинает работать зна-

чительно хуже, что связано с эффектом компенсации ошибок друг другом. Поэтому задача поиска дефекта кода должна рассматриваться как задача поиска всех возможных кратных ошибок на данном наборе тестов [4]. Отметим, что проблема построения тестов в данной работе не рассматривается.

Постановка задачи. Одним из признаков высокого качества технического проектирования является его модульность. В условиях модульного проектирования система разбивается на отдельные компоненты, при этом каждый компонент имеет свое назначение и четко определенные входы и выходы. Структуру программного модуля представим графом по управлению [1]. Каждая вершина такого графа соответствует линейному участку программы, который может заканчиваться оператором ветвления, а дуги отражают связи по управлению между линейными участками. Линейный участок характеризуется единственным входом и единственным выходом и сосредотачивает все операторы, исполняемые без разветвлений. Предполагается, что исходный граф программы ациклический и корректен, т.е. не содержит тупиковых и висячих вершин.

Ввод контрольных точек в исполняемые маршруты позволяет реализовать принятую стратегию тестирования. Под контрольной точкой понимается место в программе, где она останавливается для отладки. Программисту становится доступной дополнительная информация, помогающая локализовать ошибки – значения переменных в этой точке, распределение памяти и регистры.

Требуется выделить реализуемые маршруты тестирования программы в соответствии с заданным критерием и назначить минимальное множество точек контроля на выделенных реализуемых маршрутах для определения ошибок произвольной кратности.

Алгоритм построения графа маршрутов тестирования. В графе по управлению определим отношение порядка таким образом, чтобы каждая направленная дуга выходила из вершины с меньшим номером и входила в вершину с большим номером. Чтобы автоматизировать процесс тестирования, необходимо построить математическую модель графа маршрутов тестирования.

Маршрутом программы назовем последовательность вида $v_1, (v_1, v_2), v_2, (v_2, v_3), \dots, (v_{n-1}, v_n), v_n$, где v_i – исполняемый блок программы, (v_i, v_{i+1}) – передача управления.

Графом маршрутов назовем граф G , в котором множество вершин является объединением множеств вершин выделенных маршрутов тестирования, и i -я и j -я вершины соединяются дугой, если существует соответствующая дуга хотя бы в одном выделенном маршруте, при этом направление дуги не меняется.

Существуют различные критерии выделения маршрутов для тестирования структуры программного модуля: α_1 – выделение минимального множества тестируемых маршрутов программ, охватывающего все направления передач управления; α_2 – выбор маршрутов тестирования на основе анализа базовых маршрутов в программе, формируемых и оцениваемых на основе определения цикломатического числа

исходного графа программы; α_3 - выделений полного состава базовых структур графа программы - нахождение всех существующих ациклических маршрутов, представляющих собой полное сочетание всех дуг в исходном графе программы. Очевидно, что тестирование программного модуля по критерию α_3 позволяет обеспечить его максимальную корректность. Однако данный метод тестирования не получил практического развития из-за следующих причин: на практике существует сложность реализации всех тестовых воздействий; некоторые маршруты могут оказаться нереализуемыми из-за несовместимости условий, которые последовательно обрабатывались в разных вершинах. Это приводит к необходимости предварительного тщательного анализа совместимости условий в программах и отсеивания таких маршрутов перед построением программ тестирования.

В данной работе предложен автоматизированный метод тестирования программных модулей по критерию α_3 и показаны преимущества его использования, причем нереализуемые в программных модулях маршруты определяются в процессе тестирования, что в некоторой степени упрощает процедуру их выявления.

В качестве математического эквивалента граф-модели используется матрица смежности его вершин. Алгоритм выделения тестируемых маршрутов по критерию α_3 заключается в следующем.

На первом этапе определяется полное множество маршрутов графа программы без анализа совместимости условий в вершинах ветвления. Этот процесс полностью автоматизирован и включает следующие шаги:

1. Исходный ациклический мультиграф программы представляется его математической моделью - матрицей смежности $C = \|c_{ij}\|$, где c_{ij} ($i, j = 1, 2, \dots, n$) равно количеству дуг, инцидентных вершинам i и j .

2. Строится приведенная матрица смежности C^* мультиграфа программы. При этом количество возможных маршрутов в графе равно сумме элементов первой строки матрицы C^* (для многовходовых граф-моделей - сумме элементов первых строк, соответствующих входам).

3. По матрице смежности C исходного графа программы строится матрица смежности \bar{C} графа всех возможных маршрутов программы. Данный алгоритм формализован и автоматизирован. Построенная матрица \bar{C} является математическим эквивалентом графа маршрутов программы, выделенных по критерию α_3 . По матрице \bar{C} однозначно строится граф маршрутов программы \bar{G} на n вершинах. Количество маршрутов в программе равно сумме элементов строк матрицы \bar{C}^* , являющимися входами графа маршрутов (номера входных вершин - номера нулевых столбцов матрицы \bar{C}).

4. Строится матрица маршрутов $B = \|b_{ij}\|$ размерностью $(m \times n)$, m строк которой соответствуют проверяемым маршрутам, n столбцов - вершинам графа маршрутов \bar{G} . Алгоритм ее построения полностью формализован. Если построенная граф-модель тестируемых маршрутов имеет несколько входов, она

приводится к виду, имеющему один вход и один выход добавлением фиктивных вершин.

Предполагается, что на вход при тестировании подается тестовое воздействие, с выхода снимаются результаты работы программы.

На втором этапе проводится тщательный анализ совместимости условий программы, входящих в каждый маршрут матрицы B . В результате выявляются нереализуемые маршруты, которые удаляются из матрицы B .

Алгоритм назначения минимального множества точек контроля. В качестве контрольных точек будем использовать выход программного модуля и промежуточные результаты работы программы на выходах ее линейных блоков, соответствующих выходам вершин графа.

Проведем тестирование по всем маршрутам матрицы B . Маршруты, для которых имеем неверные результаты на выходе, перепишем в новую матрицу B^{er} - матрицу маршрутов, содержащих некорректные блоки.

Процедура поиска кратных ошибок $\Pi = \{\bar{b}_{i_1}, \bar{b}_{i_2}, \dots, \bar{b}_{i_k}\}$ будет заключаться в последовательной подаче тестовых воздействий для исполнения маршрутов $\bar{b}_{i_1}, \bar{b}_{i_2}, \dots, \bar{b}_{i_k}$. Блок, соответствующий структурной компоненте программы, содержит ошибку, если результаты на его выходе не соответствуют заданным эталонным значениям. При тестировании блока проверяются все направления передач управления. Реализация маршрута \bar{b}_{i_s} обнаруживает ошибку блока i_s при реализации k -й передачи управления в случае положительных результатов исполнения всех предыдущих маршрутов $\bar{b}_{i_1}, \bar{b}_{i_2}, \dots$, охватывающих блоки с меньшими номерами $i_1 < i_s, i_2 < i_s, \dots$, и маршрутов \bar{b}_{i_s} , тестирующих передачи управления $1, 2, \dots, k-1$. Ошибка, содержащаяся в данной структурной компоненте, устраняется.

После локализации и устранения обнаруженных ошибок следует выполнить контрольное тестирование, задача которого состоит в подтверждении правильности выполненной корректировки программы.

Процедура контрольного тестирования заключается в повторении тестирования маршрутов, содержащих блок i_s для всех передач управления. При этом проверяются также маршруты, содержащие блок i_s , но не вошедшие в матрицу B^{er} , т.к. при корректировке блока могли быть внесены новые ошибки, проявляющиеся при реализации ранее верно работающих маршрутов.

Назовем строку матрицы маршрутов определяющей, если в ней содержится только один ненулевой элемент. Для локализации ошибки блока i_s очередной тестируемый маршрут должен давать определяющую строку в матрице маршрутов, множество ненулевых столбцов которой соответствует множеству непроверенных блоков программы.

Разработанный алгоритм последовательного выделения определяющих строк матрицы тестируемых маршрутов определяет порядок подачи тестовых воздействий для обнаружения ошибок произвольной кратности. При этом построенная процедура тестирования программного модуля включает тестирование

для обнаружения ошибок и контрольное тестирование правильности выполненных корректировок.

Очевидно, что рассматриваемая проблема устранения ошибок программного кода далеко не тривиальна, но предложенная методика локализации кратных ошибок отличается простотой реализации и способствует повышению эффективности обнаружения и устранения дефектов программных модулей.

Список литературы

1. Майерс Г. Искусство тестирования программ. - М.: Финансы и статистика, 1982. -178с.
2. Липаев В.В. Надежность программных средств. -М:Синтег, 1998.-232С.
3. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование. -М.:Вильямс, 2002.-383с.
4. Сагунов В.И., Соколова Э.С., Бушуева М.Е. О поиске кратных ошибок в программных модулях // Контроль и диагностика, №8, 2001г. С. 11-13.

Эффективность лечения больных хроническим гепатитом с препаратом "Украин"

Сологуб Т.В.¹, Волчек И.В.², Новицкий Я.В.³, Григорьева Т.Д.¹

¹Медицинская академия им. И.И. Мечникова, ²ООО «Терра Медика», Санкт-Петербург; ³Украинский противораковый институт, Вена

Под нашим наблюдением находилось 60 пациентов с хроническим гепатитом С (ХГС), которые получили полусинтетический препарат «Украин» («Nowisky Pharma», Австрия), содержащий алкалоиды чистотела большого и триазириндид тиофосфорной кислоты внутривенно в дозах 0,5; 1; 2,5 и 5 мг и подкожно по 1 мг 3 раза в неделю в течение 1 месяца, 3 больных получили 3 курса внутривенно по 5 мг, всего по 30 инъекций. У 20 пациентов доза Украина определялась индивидуально на основании влияния на SH/SS соотношение крови *in vitro* (патенты РФ № 2150700, US 6,627,452, EP 1,182,455), которое определялось методом амперометрического титрования. Контрольная группа, в которую входили 25 больных ХГС получала препарат рекомбинантного интерферона альфа-2b (ИФН, Реаферон, «Вектор») в стандартной дозе 3 МЕ 3 раза в неделю.

Больные ХГС с генотипом 1b были чувствительны к Украину *in vitro* в 92,3%, тогда как к ИФН – только в 16,7% случаев. После 4 недель значительно более высокая частота полного ответа (негативация ПЦР-ВГС и нормализация АЛТ) была отмечена при лечении Украином (33,3%) по сравнению с ИФН (20%). Полный ответ был достигнут у 45% пациентов, леченных Украином после индивидуального подбора дозы. Частичный ответ (нормализация АЛТ или негативация ПЦР-ВГС) был достигнут у 31,6% пациентов, леченных Украином по сравнению с 20% в контрольной группе. В группе пациентов, леченных ИФН 60% были нечувствительными к терапии по сравнению с 35% больных, получавшими Украин. Больные ХГС с генотипом 1b были чувствительными к Украину как *in vitro*, так и *in vivo*. Полученные данные выявили преимущество индивидуализированной терапии подобранными дозами Украина.

У 3 больных ХГС с длительной вирусемией и количеством копий > 1 млн., получивших 3 курса лечения Украином по 5 мл (30 внутривенных инъекций) через год результаты вирусологического исследования были отрицательными. При использовании короткого курса препарата (10 инъекций) через 6 месяцев у всех больных отмечались рецидивы заболевания.

Таким образом, данные представленного исследования свидетельствуют о более высокой эффективности Украина по сравнению с ИФН в лечении ХГС, особенно при индивидуальном подборе доз. Представляется целесообразным использовать по меньшей мере 3 курса лечения Украином по 10 инъекций для получения стабильной ремиссии заболевания.

Использование современных методик проприоцептивной коррекции в реабилитации детей с сочетанной нейро-ортопедической патологией

Торишнев Е.Ю., Ушаков А.А.

Институт биологии и природопользования АГТУ, Астрахань

По данным статистики, за последние три года в г. Астрахани и Астраханской области отмечается рост заболеваемости детского населения на 6% (с 1253,1 до 1327,8 на 1000 детского населения). В целом по классам; заболеваемость нервной системы у детей занимает третье место.

Наши наблюдения показали, что из 1052 обследованных детей – 86% имели сочетанную нейро-ортопедическую патологию. Из них по основному диагнозу - наиболее частым является Детский церебральный паралич (ДЦП) – 58,8%, самой распространённой формой ДЦП была спастическая диплегия, которая в каждый из 3-х лет наблюдения, выходила на первое место (2000 г. – 12,49%, 2001г.- 22,45%, 2002 г. – 32,14%). Вместе с тем отмечается рост частоты встречаемости больных детей с гиперкинетической формой (с 2,42% по 8,56%), а наименее встречаемой формой ДЦП является двойная гемиплегия (0,32%). Сочетанная ортопедическая патология распределилась следующим образом: I место – деформация стоп - 39,5%, II место – нарушение осанки – 32,2%, III место – сколиозы - 8,7%, IV место – контрактуры нижних конечностей - 7,95%.

Несмотря на возрастание нейро-ортопедической патологии наибольшая обращаемость, направляемость на реабилитацию (по данным медико-социальной экспертизы г. Астрахани) детей в центры приходится на возраст от 3 до 7 лет (42,9%), наименьшая – на первом году жизни (3,6%).

Цель нашей работы явилось подобрать современные, наиболее эффективные методики, корригирующие биомеханику движений и позы сложившегося патологического стереотипа у больных с сочетанной нейро-ортопедической патологией.

В течение 3-х лет нами велись наблюдения за контрольной группой больных, состоящих из 36 человек. В основе их комплексной реабилитации лежали метод «ДИПРОКОР» - динамической проприоцеп-