OSTIS-2013
(Open Semantic Technologies for Intelligent Systems)

УДК 658.26:004.8

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ТЕСТИРОВАНИЯ ПРОГРАММНЫХ СРЕДСТВ НА ОСНОВЕ МОДИФИЦИРОВАННЫХ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Бурдо Г.Б.[*], Палюх Б.В.[*],Мельникова В.В.[**].

[*]*Тверской государственный технический университет, г.Тверь, Россия*
gbtms@yandex.ru
pboris@tstu.tver.ru

[**]*ООО «Главный испытательный сертификационный центр программных средств вычислительной техники», г. Тверь, Россия*
viklipse@mail.ru

В статье рассмотрены проблемы проведения сертификационных тестирований программных средств в скомпилированном виде. Предложен подход к созданию тестовой базы с использованием генетических алгоритмов.
Ключевые слова: сертификационное тестирование, автоматизированная тестовая база, генетические алгоритмы, тестовое покрытие.

Введение

При разработке программных средств (ПС), к качеству которых предъявляются повышенные требования, перед вводом системы в эксплуатацию требуется подтверждение со стороны уполномоченного органа (обычно государственного) соответствия ее эксплуатационных характеристик заданным критериям. Такое соответствие определяется в ходе сертификации ПС.

В настоящее время наблюдается достаточно высокий рост программных средств, причем очень часто программные средства разрабатываются в сжатые сроки и при ограниченных бюджетах проектов. Затем они постоянно дорабатываются с целью исправления выявленных ошибок или отслеживания изменений предметной области применения ПС (например, меняется законодательство). В результате ПС становятся «версионными», причем обновления могут происходить довольно часто. Каждая новая версия ПС требует повторной сертификации.

При этом обновления ПС могут происходить быстрее, чем проведение сертификации очередной его версии. Кроме того, затраты на проведение сертификации постоянно обновляющегося ПС могут свести на нет все преимущества от его использования.

В связи с этим остро встает проблема как можно более быстрого прохождения сертификации очередной версии такого ПС. Особенно трудным представляется проведение тестирования ПС без наличия исходных текстов, например, в рамках сертификации ПС по ГОСТ Р.

Сертификация ПС по ГОСТ Р, принятая в России, предусматривает проведение их тестирования, т.е. контролируемое выполнение программы на конечном множестве наборов данных (КМНД) и анализ этого выполнения с целью обнаружения ошибок [Котов, 2010].

Становится понятным, что рациональное ограничение КМНД во многом предопределяет трудоемкость и корректность тестирования, причем неавтоматизированное определение КМНД приводит к дополнительным трудозатратам.

Следовательно, задача формирования оптимального КМНД является актуальной.

Постановка задачи

Известны способы тестирования ПС на основе использования генетических алгоритмов (ГА) [Мельникова, 2011]. Критериями полноты тестового покрытия обычно являются: количество покрытых операторов в исходном коде, количество покрытых ветвей в графе потока управления, количество покрытых путей графа потока управления и т.д.

[Котляров, 2004]. Однако проверка становится возможной при наличии исходных текстов программ, в то время как государственные органы сертификации работают со скомпилированными программными средствами.

Поэтому, логичной является постановка задачи создания автоматизированной системы формирования рационального КМНД для тестирования скомпилированных ПС. В качестве аппарата для создания КМНД авторами предлагается использование модифицированных ГА.

Методика представления (топология) исходных данных

В основу создания КМНД для контролируемых расчетов положен анализ структуры и параметров исходных данных (ИД), используемых в ПС при их выполнении. Их предлагаемое представление позволит при сертификации ПС смоделировать их тестирование с использованием исходных текстов программ.

Топология ИД была рассмотрена и представлена в виде следующих типов переменных:

1. Множества ИД вещественного случайного вида, $B \equiv \{B_к\}, k = \overline{1,m}$ принимающие случайные значения в определенном интервале:

$$B_к = \{b_{кρ}\},\ B_к^{min} \le b_{кρ} \le B_к^{max},\ ρ = \overline{1,\infty} \qquad (1)$$

где $b_{кρ}$ – случайное значение входной переменной $B_к$;

$B_к^{min}, B_к^{max}$ - минимальное и максимальное значение входной переменной $B_к$; m - их количество.

Множество $B = \cup B_к$ - все массивы ИД случайного типа, определяющие их возможные наборы. Данные 1-го типа, в основном, подчиняются закону нормального распределения.

2. Множества ИД целочисленного вида, $A = \{A_i\}, i = \overline{1,n}$ (2-й тип):

$$A_i = \{A_{ij}\},\ A_i^{min} \le a_{ij} \le A_i^{max},\ j = \overline{1,t_i},\ a_{ij} - \text{int}, \qquad (2)$$

где n – число целочисленных переменных; t – число значений A_i-ой переменной; A_i^{min} и A_i^{max} - минимальное и максимальное значение A_i-ой переменной.

В отличии от данных 1-го типа, используемых непосредственно для получения численного значения результата при использовании ПС, данные 2-го типа используются для определения основного пути графа вычислений. К этим же данным, естественно, относятся и ИД символьного типа (буквы, слова, фразы) из определенного ограниченного набора. Это корректно, т.к. с точки зрения влияния на ход выполнения программы они - переменные одного смыслового содержания.

3. Множества логических (булевых) переменных в составе ИД, $L = \{L_q\},\ q = \overline{1,d}$, d – их количество.

$$L_q = \{l_{qr}\},\ r = 1,2,\ l_{qr} = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}, \qquad (3)$$

Эти исходные данные (3-й тип) определяют разветвление пути графа вычислений. Им так же по смыслу соответствуют и синтаксические переменные, состоящие из наборов по 2 буквы (по 2 слова, фразы).

Методика и алгоритм создания КМНД

Методика создания КМНД состоит в том, чтобы при тестировании скомпилированных программ смоделировать такой вычислительный процесс, который бы косвенно отражал процесс и критерии тестирования ПС в исходных текстах. Такие критерии, как количество покрытых ветвей и путей в графе потока управления, косвенно определяется числом переборов ИД переменных из типов 2 и 3.Корректность работы программы в диапазоне возможных числовых значений исходных данных, используемых для непосредственных расчетов, определяется тем, насколько перекрыта и с какой дискретностью область значений переменных 1-го типа.

Рассмотрим генотип кодирования ИД для тестирования. Топологию исходных данных удобнее представлять в виде генотипа, состоящего из трех хромосом, каждая из хромосом отвечает за ИД соответствующего типа. Структура хромосом показана на рис. 1.

Генами хромосомы являются переменные A_i, B_k, L_q.

1.	B_1	....	B_k	....	B_m
2.	A_1	....	A_i	....	A_n
3.	L_1	....	L_q	....	L_d

Рисунок 1 - Структура хромосом

Алгоритм создания КМНД построен следующим образом.

На первом этапе создаются наборы для обеспечения тестового покрытия ветвей и путей в графе потока информации, т.е. варьируют хромосомами типов 2 и 3 при фиксированном значении хромосомы 1. В этом случае, диапазон всех значений переменных B_k разбивается на нечетное одинаковое число интервалов I (рекомендуется 7, 9 или 11).

Значение каждой величины B_k случайным образом генерируется из срединного интервала.

На первом этапе наборы хромосом 2-го и 3-го типов строятся из предположения, что распределение значений этих ИД подчиняется закону равной вероятности. Учитывая, что мы не знаем места ветвления программы, для полного тестового покрытия необходимы 100% генерация возможных вариантов, но на практике бывает достаточным ограничить перебор, задав для формирования КМНД процент перебора данных в переменных целочисленного 2-го типа и обеспечив полный перебор данных 3-го булева типа.

Исходная популяция хромосом для этапа 1 получается следующим образом. Из хромосомы 1 со значениями B_k из срединного интервала I_{cp}, хромосом 2 и 3 со случайными значениями $\{a_{kj}\}$ и $\{l_{qr}\}$ получается первый набор из трех хромосом.

Следующий, III-й набор из трех хромосом получается следующим образом: а) 1 и 2-я хромосомы остаются неизменными; б) в третьей хромосоме гены l'_{qr} дополняют гены l_{qr} до полного множества $\{l_{qr}\} = L_q$ (т.е. берется второй элемент бинарного множества):

$$l'_{qr} = L_q \setminus l_{qr}. \qquad (4)$$

Четвертый и последующие (производные) наборы хромосом на первом этапе строятся по следующему алгоритму:

1 шаг. Берется из исходной популяции хромосом набор I.

2 шаг. Указанный набор фиксируется в новом производном наборе хромосом.

3 шаг. В хромосоме 2 производного набора случайным образом мутируется n-ый ген (не допуская повторов с предыдущими вариантами), получаем новый тестовый набор. Тестовый набор заносится в память.

4 шаг. В полученном на шаге 3 тестовом наборе хромосома 3 заменяется хромосомой из исходного набора II. Заносим тестовый набор в память;

5 шаг. Проверяется условие окончания мутации гена n (по проценту покрытия значений интервала от A_i^{min} до A_i^{max}). Если нет, то возврат на шаг 3, иначе – шаг 6.

6 шаг. Проверяется условие окончания мутации предшествующего (n - 1) гена. Если нет, то переход к оператору 7, иначе – шаг 8.

7 шаг. В последнем тестовом наборе случайным образом, исключая повторы, в хромосоме 2 мутируется предшествующий (n - 1) ген. Переход на шаг 3.

8 шаг. Проверяется условие окончания мутации предшествующего (n - 2) гена. Если нет, то переход к оператору 9, иначе – шаг 10;

9 шаг. В последнем тестовом наборе случайным образом, исключая повторы, в хромосоме 2 мутируется предшествующий (n - 2) ген. Переход на шаг 3.

10 шаг. Проверяется условие окончания мутации (n - 3) гена. Если нет, то переход к шагу 11, иначе к шагу 12;

11 шаг. В последнем тестовом наборе случайным образом, исключая повторы, в хромосоме 2 мутируется (n - 3) ген. Переход на шаг 3.

…и т.д.

Предпоследний шаг. Проверить окончание мутации (n – (n - 1)) гена. Если нет, то переход к последнему шагу, иначе окончание генерации тестовых наборов.

Последний шаг. В тестовом наборе случайным образом мутируется (n – (n - 1)) ген. Переход на шаг 3.

Отдельного разговора заслуживает вопрос о процентном соотношении мутируемых вариантов генов во второй хромосоме. Для гена A_1, очевидно, должен быть обеспечен полный перебор. Для гена A_2 достаточным будет обеспечить проверку для $K_2 \cdot \frac{100\%}{t_1}$ процента вариантов, где K_2 - коэффициент запаса, t_1 - количество вариантов гена A_1. K_2 исходя из наихудшего варианта вероятностного сложения полей рассеивания переменных, рекомендуется брать равным 1,73.

Тогда процент мутации i-го гена, P_{ij}; в общем случае:

$$P_i = \frac{k^{i-1} \cdot 100\%}{t_1 \cdot t_2 \cdot \ldots \cdot t_{i-1}} = \frac{k^{i-1} \cdot 100\%}{\prod_1^{i-1} t_i}, \qquad (5)$$

где $\prod$ - знак произведения.

На втором этапе выполняется формирование КМНД для тестирования ПС с точки зрения выполнения корректности расчетов в области существования переменных, определяющих числовые данные вычислений. С этой целью строятся варианты наборов из мутируемой хромосомы 1 и постоянного состава генов в хромосомах 2 и 3 из набора I первого этапа.

Исходная популяция хромосом состоит из двух наборов, определяемых хромосомой 1. В первом наборе в хромосоме присутствуют максимальные значения B_k, во втором – минимальные.

Остальные тестовые наборы получаются мутацией хромосомы 1, помня, что диапазоны каждого гена B_k разбиты на I число интервалов.

Предположив, что вероятность ввода значений исходных данных подчиняется закону Гаусса, и, учтя, что крайние значения B_k имеются в базовой (исходной) популяции при числе $I = 7 \div 11$ мутация генов в пределах срединного и двух смежных интервалов обеспечит перекрытие всего диапазона изменения B_k не менее 60%.

Таким образом, алгоритм формирования КМНД на втором этапе следующий.

Шаг 1. Формируем исходную популяцию хромосом.

Шаг 2. Берем из исходной популяции хромосом набор I.

Шаг 3. В наборе I выполняем случайным образом неповторяющуюся мутацию в гене B_m хромосомы 1 (в пределах срединного и в каждом смежном интервале). Запоминаем тестовый набор.

Шаг 4. Определяем окончание мутации гена B_m (по 1 значению в срединном и по 1-му в каждом смежном интервале). Если да – переход к шагу 5, нет – возврат к шагу 3.

Шаг 5. Проверить окончание мутации B_{m-1} (предшествующего) гена аналогично гену B_m. Если нет, переход на шаг 6, если да – переход на шаг 7.

Шаг 6. Случайным образом осуществляем мутацию гена B_{m-1} (аналогично гену B_m). Возврат на шаг 3.

Шаг 7. Проверяем окончание мутации гене B_{m-2}. Если нет – переход на шаг 8, если да – переход на шаг 9.

и т.д.

Предпоследний шаг. Проверить окончание мутации по $B_{m-(m-1)}$ гену. Если да, то окончание генерации тестовых наборов, нет – переход к последнему шагу.

Последний шаг. Случайным образом (аналогично гену B_m) мутируется $B_{m-(m-1)}$ ген. Возврат на шаг 3.

Методика тестирования ПС

Предлагаемая методика тестирования ПС следующая. Первоначально тестируются ПС с использованием КМНД, полученных на первом этапе. Его успешное завершение подтверждает, как минимум, корректность ветвей и путей в графе потока информации.

Во вторую очередь тестируются программные средства с использованием КМНД, полученных на этапе 2, что позволит, как минимум, утверждать корректность выполнения расчетов на всем диапазоне возможных значений входных данных.

Заключение

Изложенные положения были подтверждены сравнением тестирования ПС в скомпилированном виде с тестированием программ в исходных текстах. Сгенерированные наборы данных обеспечивали необходимую полноту тестового покрытия.

В настоящее время методика и разработанные на ее основе ПС, проходит опытную проверку в ООО «Главный сертификационный испытательный центр программных средств вычислительной техники» (г. Тверь, ул. Ржевская, д.10, 170023).

Библиографический список

[Мельникова, 2011] Мельникова В.В. Тестирование программ с использованием генетических алгоритмов. // В.В. Мельникова, Б.В. Палюх, С.Л. Котов, М.А. Проскуряков. - Программные продукты и системы.-2011.-№4.-С.114- 117.

[Котляров, 2004] Котляров В.П. Основы современного тестирования программного обеспечения, разработанного на c#:учебное пособие// В.П.Котляров, Т.В. Коликова. - СПб, 2004.- С. 55-57.

[Котов, 2010] Котов С.В. Проведение сертификационных испытаний часто обновляющихся ПС с использованием автоматизированной тестовой базы.// С.В. Котов, А.А. Демирский, В.В. Мельникова.- Вестник ВНИИМАШ, 2010.

AUTOMATED SYSTEM OF TESTING OF SOFTWARE ON THE BASIS OF THE MODIFIED GENETIC ALGORITHMS

Burdo G.B. *, Palyukh B.V. *,
Melnikova V.V. **

**Tver State Technikal University, Tver, Russia*

gbtms@yandex.ru

***Chief Center for Software Certification and Testing (GIC PS VT), Tver, Russia*

viklipse@mail.ru

The article considers the problems of carrying out certification tests of software in compiled form. An approach is proposed to create a test database using genetic algorithms

Keywords: certification test, automated test framework, genetic algorithms, test coverage.