КАЧЕСТВО
ИННОВАЦИИ
ОБРАЗОВАНИЕ

ISSN 1999-513X

№10
2011

журнал в журнале

КАЧЕСТВО и ИПИ (CALS)-технологии

www.quality-journal.ru

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

АДРЕС РЕДАКЦИИ И ИЗДАТЕЛЯ
109028, Москва, Большой Трехсвятительский пер., д. 3/12
Тел.: +7 (495) 916-28-07, +7 (495) 916-8929, факс: + 7 (495) 916-8865
E-mail: quality@miem.edu.ru (для статей), nii@miem.edu.ru (по общим вопросам)
www.quality-journal.ru; www.quality21.ru

УЧРЕДИТЕЛИ
Российский государственный университет инновационных технологий и предпринимательства (РГУИТП)
Московский государственный институт электроники и математики (МИЭМ)
МАТИ – «Российский государственный технологический университет им. К.Э. Циолковского»
«Европейский центр по качеству»

ПРЕДСЕДАТЕЛЬ СОВЕТА УЧРЕДИТЕЛЕЙ
Быков Д.В.

ИЗДАТЕЛЬ
Европейский центр по качеству

НАУЧНЫЙ РЕДАКТОР
Леохин Ю.Л.

АВТОР ДИЗАЙН-ПРОЕКТА
Логинов К.В.

ОТВЕТСТВЕННЫЙ СЕКРЕТАРЬ
Савин Е.С.

ЖУРНАЛ ЗАРЕГИСТРИРОВАН
в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций. Свидетельство о регистрации ПИ № 77-9092.

ПОДПИСНОЙ ИНДЕКС
в каталоге агентства «Роспечать» 80620, 80621; в каталоге «Пресса России» 14490.

ОТПЕЧАТАНО
«Полиграфическая компания "Принтико"». Москва, ул. Краснобогатырская, д. 6., www.sts-print.ru

Журнал входит в перечень ВАК РФ

Статьи рецензируются

КАЧЕСТВО ИННОВАЦИИ ОБРАЗОВАНИЕ

Номер 10 (77), октябрь, 2011

Журнал выходит при содействии Министерства образования и науки РФ

Журнал осуществляет информационную поддержку научно-технических программ и научно-технических мероприятий Министерства образования и науки РФ

СОДЕРЖАНИЕ

МЕНЕДЖМЕНТ И СИСТЕМЫ КАЧЕСТВА ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЙ

В.В. БЫСТРОВ, А.В. МАСЛОБОЕВ
Информационная поддержка управления качеством образования в контексте глобальной безопасности развития региона 2

Н.И. БАРСУКОВА, В.А. ЮРЬЕВ
Разработка системы мониторинга показателей качества образовательного процесса ВГТУ 11

А.В. ГАНИЧЕВА
Индексный анализ структурных компонент и показателей качества учебного процесса 15

ПОДГОТОВКА СПЕЦИАЛИСТОВ В ОБЛАСТИ ИПИ (CALS)-ТЕХНОЛОГИЙ

В.Н. АЗАРОВ, А.Н. ТИХОНОВ
Система аттестации и сертификации специалистов в области информационно-коммуникационных технологий – Руководство по качеству (продолжение) 18

МЕНЕДЖМЕНТ КАЧЕСТВА И ИННОВАЦИОННЫЙ МЕНЕДЖМЕНТ

С.А. ПЕТРОВА, И.Л. ЯКОВЛЕВА
Управление качеством профессионального образования как фактор обеспечения конкурентоспособности образовательных услуг 27

С.А. ГРИШАЕВА, А.В. МАТВЕЕВ
Стратегическое планирование в системе менеджмента качества 33

О.В. НИКУЛИНА
Системный подход к управлению инновационным развитием промышленных предприятий 37

КАЧЕСТВО и ИПИ(CALS)-ТЕХНОЛОГИИ

ПРИБОРЫ, МЕТОДЫ И ТЕХНОЛОГИИ

ФИЛИППОВ В.А., ХАТЬКО Е.Е.
Алгоритмы генерации тестовых сценариев для повышения качества программного обеспечения многозадачных пользовательских комплексов 47

Д.Ю. ОРЛОВ, Г.И. ЭЙДЕЛЬМАН, Р.А. КУЗНЕЦОВ, Е.В. АРЕФЬЕВ
Сертификационные испытания программного обеспечения автомобильного транспорта 52

Е.Н. ЧУМАЧЕНКО, В.Р. ШАШМУРИНА, Л.Э. ДЕВЛИКАНОВА, И.В. ЛОГАШИНА
Автоматизация обоснования оценки качества депульпирования премоляров при различных состояниях костных тканей нижней челюсти 55

ТЕОРИЯ И ПРАКТИКА КОНТРОЛЯ, ИЗМЕРЕНИЙ, ИСПЫТАНИЙ И ДИАГНОСТИКИ

Ю.И. БИТЮКОВ, Ю.И. ДЕНИСКИН, П.В. МИРОШНИЧЕНКО
Использование статистических методов и геометрического моделирования в контроле качества изготовления конструкций из волокнистых композиционных материаловя 61

Ю.Э. ВАСИЛЬЕВ
Автоматизация и управление результатами межлабораторных сравнительных испытаний прочности цементобетона 65

ЭКОНОМИКА И УПРАВЛЕНИЕ

Т.Н. ГРОМОВА
Экономический механизм управления межотраслевым вертикально интегрированным холдингом 72

Е.П. КОПТЕВА
Финансовая политика инновационной деятельности 75

Сведения о членах редколлегии и об авторах статей можно найти на сайте журнала www.quality-journal.ru

В.А. Филиппов, Е.Е. Хатько

АЛГОРИТМЫ ГЕНЕРАЦИИ ТЕСТОВЫХ СЦЕНАРИЕВ ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МНОГОЗАДАЧНЫХ ПОЛЬЗОВАТЕЛЬСКИХ КОМПЛЕКСОВ

В тестировании на основе моделей широко применяется генерация тестов на основе расширенных конечных автоматов. Задача генерации тестов сводится к задаче обхода графа расширенного конечного автомата. В статье рассмотрены две вариации алгоритма A*, которые можно применять для решения задачи "Китайского почтальона" в случае динамически меняющихся графов. Представлены основные параметры алгоритмов. Алгоритмы подвержены оценке с точки зрения длины результирующего пути и временной сложности.

Ключевые слова: планирование качества, система менеджмента качества, принципы планирования качества ВУЗа

V.A. Filippov, E.E. Hatko

TEST CASE GENERATION ALGORITHMS TO IMPROVE THE QUALITY OF SOFTWARE FOR MULTITASKING USER COMPLEXES

The most common technique is to derive tests from the application's model. Models can be built using extended finite state machines, which can be represented as dynamic graphs with parameters and conditions. The most often coverage criterion used in testing is the full edge coverage. To achieve such coverage, the Chinese postman problem should be solved for a dynamic graph. Two variations of the algorithm, optimized for the Chinese Postman Problem for the dynamic graphs are presented in the paper. Costs functions are proposed. Algorithms are compared in terms of resulting path length.

Keywords: quality planning, system of quality management, principles of quality planning of high school management

Постановка задачи

Для обеспечения качества программного обеспечения можно применять подход тестирования на основе моделей. Это направление является перспективным с точки зрения автоматизации процессов тестирования. В тестировании на основе моделей широко применяется генерация тестов на основе расширенных конечных автоматов. Расширенный конечный автомат – это усовершенствованная модель конечного автомата (КА). В традиционном КА переход из состояния в состояние связан с набором входных булевых условий и набором выходных булевых функций. В расширенной модели переход может быть выражен "IF" выражением. Если все условия перехода соблюдены, то происходит переход, переводя автомат в следующее состояние, при этом производятся требуемые операции с данными [1]. Такой автомат представляет собой направленный динамически меняющийся граф с условными переходами. Узлы графа – это состояния автомата, а ветви графа – переходы между состояниями.

Задача нахождения полного тестового набора сводится к задаче реализации алгоритма траверса графа [2]. Рассмотрим задачу нахождения минимального пути на динамически меняющемся графе, который позволит покрыть все переходы, т.е. пройти все ветви графа и при этом сделать общую длину пути минимальной. Эта задача актуальна в тестировании, когда нужно протестировать заданный функционал, используя минимальное количество действий.

Традиционный подход

Для случая обычного, статичного графа эта задача известна как задача «Китайского почтальона». В общем случае, её решение состоит из следующих этапов [3]:

1. Эйлеризация графа G:

- образуем матрицу D цепей наименьшего веса для вершин графа с нечетными степенями, используя, например, алгоритм Флойда-Уоршела [4];
- используя матрицу D, находим цепное паросочетание с наименьшим весом, используя алгоритм Эдмондса минимального паросочетания;
- добавлением в граф дублирующих ребер, найденных в предыдущем шаге, получаем новый граф G', в котором существует эйлеров цикл.

2. Нахождение Эйлерова цикла по алгоритму Флёри

Сложность рассматриваемой задачи заключается в том, что граф расширенного конечного автомата содержит параметры и условия. Совершая переход из одной вершины в другую, мы устанавливаем значения параметров графа. С новыми значениями пара-

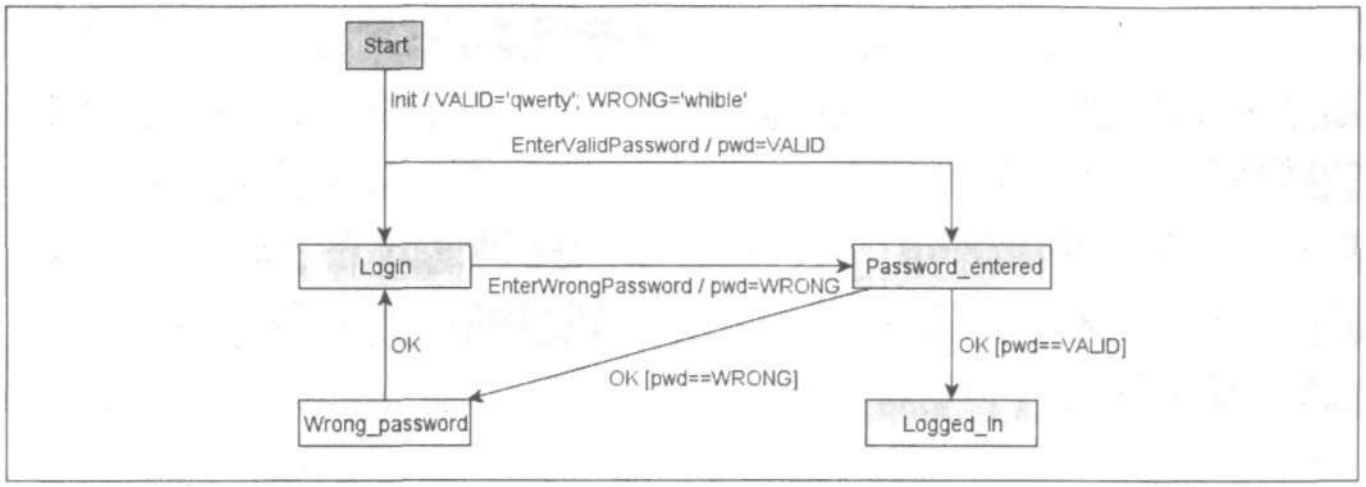

Рис. 1

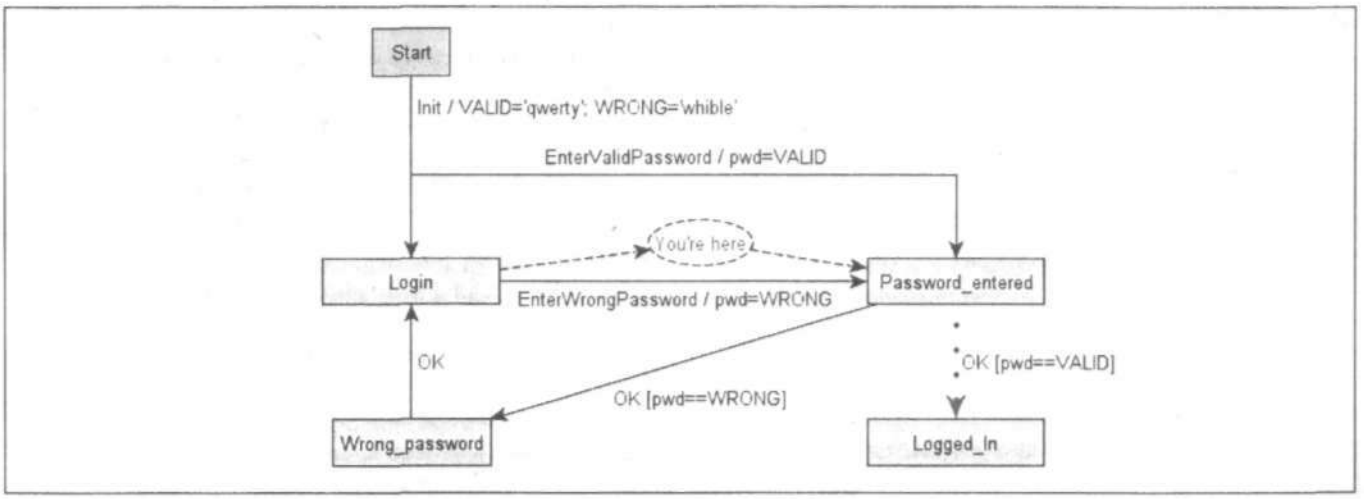

Рис. 2

метров некоторые ребра графа становятся непроходимыми.

Из рисунка 2 видно, что если совершается переход по ребру **EnterWrongPassword**, то при этом устанавливается значение параметра **pwd**, равное **WRONG**. В этом случае ребро **OK [pwd==VALID]** становится непроходимым. Аналогичные ситуации возникают при прохождении других ребер.

Предлагаемый подход

Традиционный подход не позволяет решить задачу «Китайского почтальона» для динамически меняющегося графа, поскольку в соответствии с этим подходом исходный граф G фиксируется, потом строится граф G' и потом находится Эйлеров цикл.

Предлагаемый подход состоит в том, чтобы использовать неинформированный алгоритм поиска пути на графе, т.е. находясь в некоторой вершине графа решать задачу о наименьшем пути локально, используя эвристику. Предлагается использовать модифицированный алгоритм . Алгоритм позволяет эффективно находить кратчайшие пути на графе из «начальной» вершины в «конечную» [5]. Проблема использования алгоритма для решения поставленной задачи заключается в отсутствии «конечной» вершины, поскольку задача начинается с некоторой начальной (заданной априори) вершины графа, далее происходит процесс траверса до тех пор, пока не будут покрыты все переходы. Предлагается модификация алгоритма для решения задачи «Китайского почтальона» в случае динамически изменяющихся графов. В алгоритме основой является функция стоимости, которая позволяет определить «следующее» ребро графа, находясь в «текущем» (берется ребро с наименьшим значением). Также алгоритм использует закрытый список (*cList*), который содержит все покрытые переходы, и открытый (*oList*) – список ещё непокрытых переходов. Предлагаются две вариации алгоритма A*, в которых используются различные функции F.

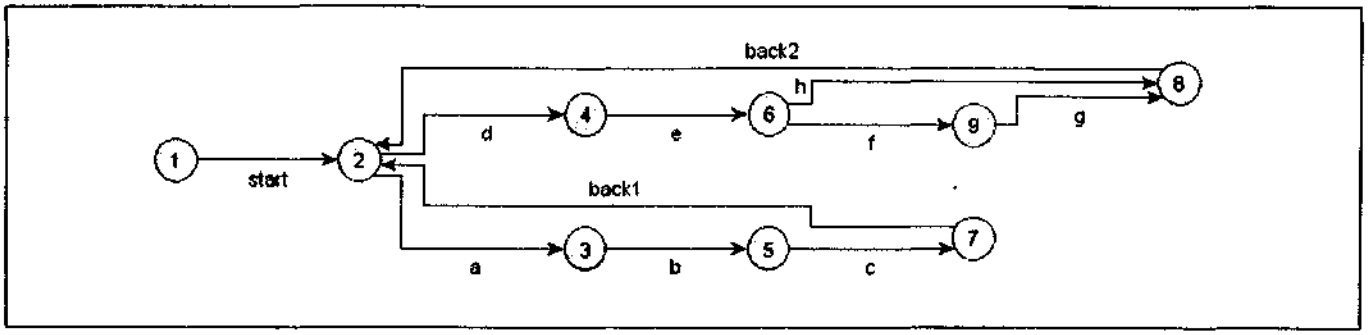

Рис. 3

Вариация №1

Рассмотрим алгоритм A^*_1.

Введём условия для выбора «следующей» ветви, находясь в «текущей»:

1. Выбирается ветвь, которая ещё не покрыта (для всех ветвей вводится флаг *unc_edge* равный 0 – если ветвь непокрыта, и 1 – в противном случае);
2. Выбирается ветвь, входящая в узел, который имеет наибольшее значение $\delta = (out - in)$, где *out* – количество исходящих ветвей, *in* – количество входящих ветвей в узел;
3. Выбирается ветвь, условие которой «удаляет» наименьшее количество непокрытых ветвей (*edges_deleted_by_condition*);
4. Выбирается ветвь, условие которой делает недоступными наименьшее количество непокрытых ветвей (ветви становятся недоступными благодаря удалению других ветвей и 3 пункта) (*edges_unaccessible_by_condition*);
5. Выбирается ветвь, ведущая в узел с наибольшим количеством непокрытых исходящих ветвей (*unc_out_edges*).

Приоритет условий устанавливается при помощи специального масштабного коэффициента: $K = \max_{i \in Nodes}(in_i + out_i)$ – по всем узлам, где in_i / out_i - количество входящих/исходящих ветвей для i-го узла.

Таким образом:

$$F = unc\_edge \times K^3 - \delta \times K^2 + edges\_deleted\_by\_condition \times K + edges\_unaccessible\_by\_condition \times K - unc\_out\_edges \quad [2]$$

Вариация №2

Рассмотрим алгоритм A^*_2.

Предлагается представление $F = G + H$, где G – стоимость пути к следующей вершине из начальной. H

Таблица 1. «A^*_1»

Step	cList	oList	current	successors (F)	coverage
1	*empty*	start	start	**a(-1)** d(-1)	0,091
2	start	a d	a	**b(-1)**	0,182
3	start a	d b	b	**c(-1)**	0,273
4	start a b	d c	c	**back1(24)**	0,364
5	start a b c	d back1	back1	a(125) **d(-1)**	0,455
6	start a b c back1	d a	d	**e(-27)**	0,545
7	start a b c back1 d	a e	e	**f(-1)** h(24)	0,636
8	start a b c back1 d e	a f h	f	**g(24)**	0,727
9	start a b c back1 d e f	a h g	g	**back2(25)**	0,818
10	start a b c back1 d e f g	a h back2	back2	**a(125)** d(125)	0,909
11	start a b c back1 d e f g back2	a h d	a	**b(125)**	0,909
12	start a b c back1 d e f g back2 a	h d b	b	**c(125)**	0,909
13	start a b c back1 d e f g back2 a b	h d c	c	**back1(150)**	0,909
14	start a b c back1 d e f g back2 a b c	h d back1	back1	a(125) **d(125)**	0,909
15	start a b c back1 d e f g back2 a b c back1	h d a	d	**e(99)**	0,909
16	start a b c back1 d e f g back2 a b c back1 d	h a e	e	f(125) **h(25)**	0,909
17	start a b c back1 d e f g back2 a b c back1 d e	h a f	h	**back2(150)**	1,000

Таблица 2. «A^*_2»

Step	cList	oList	current	successors (G+H)	coverage
1	*empty*	start	start	**a(10+1)** d(10+1)	0,091
2	start	a d	a	**b(20+1)**	0,182
3	start a	d b	b	**c(30+1)**	0,273
4	start a b	d c	c	**back1(40+1)**	0,364
5	start a b c	d back1	back1	a(60+5) **d(50+1)**	0,455
6	start a b c back1	d a	d	**e(60+1)**	0,545
7	start a b c back1 d	a e	e	**f(70+1)** h(70+1)	0,636
8	start a b c back1 d e	a f h	f	**g(80+1)**	0,727
9	start a b c back1 d e f	a h g	g	**back2(90+1)**	0,818
10	start a b c back1 d e f g	a h back2	back2	a(110+7) **d(110+3)**	0,909
11	start a b c back1 d e f g back2	a h d	d	**e(110+2)**	0,909
12	start a b c back1 d e f g back2 d	a h e	e	f(110+6) **h(100+1)**	0,909
13	start a b c back1 d e f g back2 d e	a h f	h	**back2(120+11)**	1,000

– эвристическая функция, оценивающая длину пути до «конечной» вершины.

$G(e) = \langle\ |cList| + \begin{cases} e.price, e \in cList \\ 0, e \notin cList \end{cases} \rangle \times K$, где e – текущее ребро; $e.price$ - вес текущего ребра; K – масштабный коэффициент.

$K = \max_{i \in Nodes}(in_i + out_i)$ – по всем узлам, где in_i / out_i – количество входящих/исходящих ветвей для i - го узла.

$H = |path\_to\_closest\_unc|$ – длина пути до ближайшей непокрытой вершины.

При этом перед каждым расчетом G и H все веса ребер пересчитываются с учетом условия «следующего» ребра. Ребра, которые становятся недоступными из-за условия перехода, получают бесконечную стоимость. Путь до ближайшей нераскрытой вершины ищется при помощи алгоритма BFS [6].

Различия в логике выбора переходов

Алгоритм A^*_1 является менее информированным, чем A^*_2. Благодаря вышеописанному представлению F, A^*_1 находит оптимальный переход, «заглядывая» на 2 шага вперед. Он хорош для простых ($N \leq 10$, $M \leq 20$), несимметричных графов. Алгоритм A^*_2 является более надежным, и может применяться для сложных графов, но для простых графов может выдавать неоптимальные результаты. Также из-за BFS поиска ребер для расчета H алгоритм является более медленным, чем A^*_1. Для тестового графа, изображенного на рис. 2, в таблице представлены подробные результаты работы алгоритмов.

Сравнение предложенных алгоритмов

Следующие параметры являются входными для алгоритмов:

N – количество вершин графа;

M – количество ребер графа;

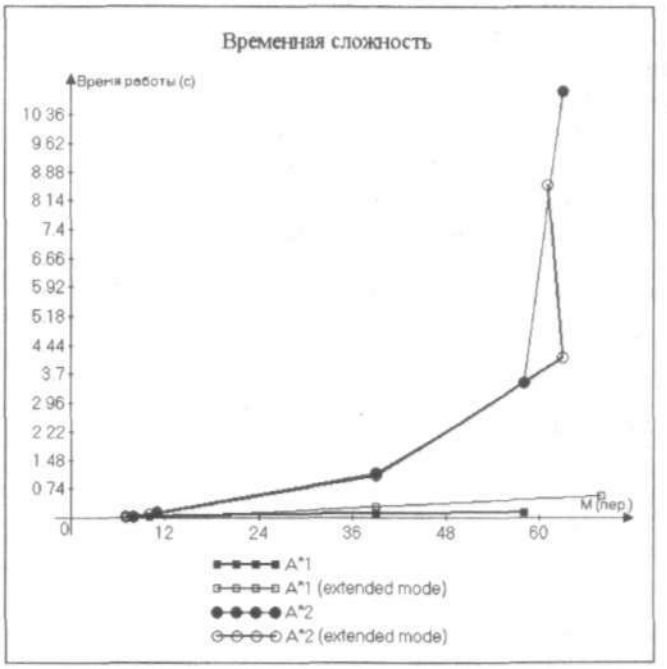

Рис. 4

P – количество параметров графа;

C – количество условий на переходах.

Рассмотренные алгоритмы применяются для тестирования ВЭБ приложений и «мобильных» приложений для Смартфонов и Коммуникаторов. В среднем количество состояний таких приложений лежит в следующих приделах: $N \in [2,...,20]$. Соответствующий граф расширенного конечного автомата должен быть связным, следовательно, наименьшее количество переходов равно: $M_{MIN} = N - 1$. Оценка максимального количества переходов: $M_{MAX} = \frac{K \times N}{2}$, где $K = \max_{i \in Nodes}(in_i + out_i)$ - максимальная по узлам сумма исходящих и входящих

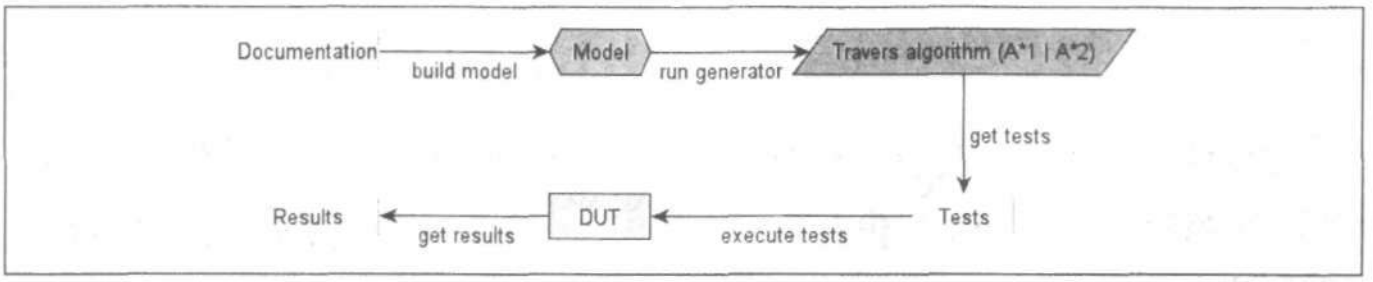

Рис. 5

Таблица 3. **Результаты**

Граф	*N*	*M*	*P*	*C*	*Длина пути 1*	*Длина пути 2*	*Время 1(с)*	*Время 2 (с)*
G1	5	7	0	0	8	8	0.015	0.015
G1 extended	5	7	3	2	7	7	0.016	0.031
G2	4	10	0	0	10	11	0.032	0.125
G2 extended	4	10	3	6	10	10	0.031	0.093
G3	10	19	0	0	39	39	0.109	1.141
G3 extended	10	19	3	3	39	39	0.282	1.078
G4	13	24	0	0	58	58	0.141	3.480
G4 extended	13	24	4	4	68	63	0.562	4.140
G5	15	35	0	0	94% covered	63	-	10.969
G5 extended	15	35	9	11	97% covered	61	-	8.547

ветвей в узел. Таким образом, среднее количество ветвей графа: $M = \frac{K \times N}{2} + N - 1 \approx \frac{K+2}{4} N$. Проведенный анализ различных моделей приложений, построенных на основе расширенных конечных автоматов, показывает, что среднее значение K порядка $K < 5$, поэтому $M \le \frac{7}{4} N \le 2N$.

Для сравнения алгоритмов были выбраны несколько графов, удовлетворяющих следующим входным условиям: $N \in [2,...,20]$, $M \approx 2N$. Для каждого графа траверс выполнялся с учетом условных переходов и без учета. Полученные результаты занесены в таблицу 3.

На рисунке 4 изображен график сравнения временной сложности рассмотренных алгоритмов. На оси абсцисс отложены различные длины результирующего пути – M (переходов). На оси ординат отложены соответствующие времена работы алгоритмов – Время работы (секунды).

Заключение

В статье рассмотрены две вариации алгоритма , которые можно применять для генерации тестовых сценариев, используя расширенные конечные автоматы. Представлены основные параметры алгоритмов. Алгоритмы подвержены оценке с точки зрения длины результирующего пути. Схема, изображенная на рис. 5, является обобщающей схемой применения рассмотренных алгоритмов.

ЛИТЕРАТУРА

1. http://www.networkdictionary.com/software
2. Е. Хатько: Один способ реализации алгоритма генерации тестов в тестировании на основе моделей. // 53-я научная конференция МФТИ, Секция микропроцессорных технологий.
3. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. С. 231-237.
4. http://algolist.manual.ru/maths/graphs/shortpath/floyd.php
5. http://www.policyalmanac.org/games/aStarTutorial_rus.htm
6. Skiena S. The algorithm design manual. London, 2008, Springer. PP. 162-169,
7. Pearl, Judea (1984). Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley.

Хатько Евгений Евгеньевич,
аспирант.
Московский Физико-Технический институт
Evgeniy.khatko@gmail.com

Филиппов Владимир Александрович,
канд. техн. наук, профессор, МИЭМ, МФТИ.
filbob@infoline.su