Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

КУРСОВАЯ РАБОТА

Конечно-автоматное модульное тестирование программных реализаций на языке Java

Разенков Семен Игоревич

Направление подготовки 10.05.01 Компьютерная безопасность

Направленность (профиль) «Анализ безопасности компьютерных систем»

Руководитель работы

канд. физ.-мат. наук

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Твардовский

*подпись*

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

Автор работы

студент группы № 931825

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.И. Разенков

*подпись*

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

Томск – 2022

**ОГЛАВЛЕНИЕ**

Введение 3

1 Основные понятия и определения 4

2 Платформа JUnit5 4

Выводы

Список использованных источников и литературы

**ВВЕДЕНИЕ**

Конечно-автоматные методы широко используются в тестировании программного обеспечения, поскольку имеют гарантированную полноту покрытия неисправностей. Однако, высокая сложность (длина) этих тестов подталкивает к автоматизации тестирования. Одним из объектов тестирования являются реализации на объектно-ориентированных языках программирования, для проверки которых может быть использовано модульное тестирование.

В данной работе мы решаем задачу автоматизации модульного тестирования автоматными методами для классов на языке Java.

**1 ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ**

Конечным автоматом M называется синхронная система с конечным входным алфавитом , с конечным выходным алфавитом , с конечным множеством состояний и двумя характеристическими функциями и :

,

,

где и - соответственно входной символ, выходной символ и состояние автомата M в момент [1]. Система называется синхронной, поскольку её переменные рассматриваются в дискретные моменты времени и не зависят от его текущего значения, а только от номера .

Автомат называется полным или полностью определенным (детерминированным), если области определения его характеристических функций совпадают с ; в противном случае автомат называется частичным [2]. В данной работе будут рассматриваться детерминированные конечные автоматы.

// Отношения переходов, классификация

Автомат называется *приведенным*, если любые два состояния в нем различимы[2], т.е. Для них существует такая последовательность входных символов, при подаче которой выходные последовательности автомата, находившегося изначально в каждом из этих двух состояний, будут разными.

Пример. Рассмотрим конечный автомат, описывающий поведение муравья. Входной алфавит автомата представляет собой множество всех 3-битных чисел. Каждый разряд числа, начиная со старшего, соответственно показывает истинность утверждений: «у муравейника», «видит листик», «видит хищника». Выходной алфавит представляет собой множество 2-битных чисел, характеризующих скорость движения муравья. Множество состояний автомата следующее: «сидит в муравейнике», «ищет листик», «тащит листик», «убегает от хищника». Сопоставим каждому состоянию соответственно числа от 0 до 3. Тогда получим следующий конечный автомат (рис. 1):

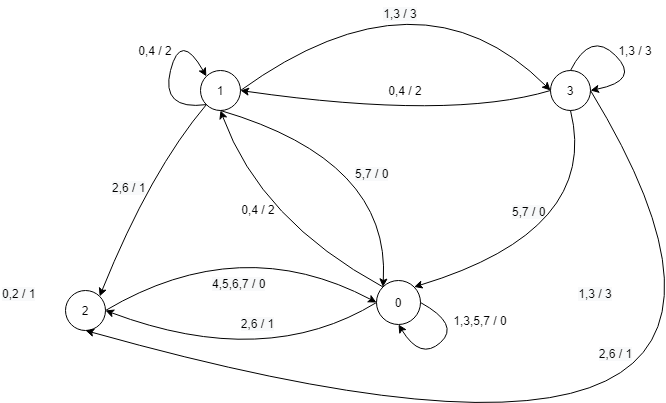


Рисунок 1. Диаграмма переходов конечного автомата.

Состояния на рисунке отмечены окружностями, представляющими собой вершины ориентированного графа, а дуги отражают действие характеристических функций. Каждая дуга подписана входными значениями, вызывающими данный переход из одного состояния в другое, а также получающимися при этом соответствующие выходные значения.

Модель неисправности конечного автомата представляет собой тройку: эталон – конечный полностью определенный детерминированный приведённый инициальный автомат с n состояниями; область неисправности – множество конечных полностью определенных детерминированных инициальных автоматов с числом состояний не более, чем у эталонного автомата, и тем же выходным алфавитом; отношение конформности – отношение эквивалентности.

Полным проверяющим тестом называется конечное множество входных последовательностей конечной длины, по реакциям на которые можно отличить всякую неконформную реализацию из области неисправности. Если проверяемый автомат на последовательности теста реагирует так же, как эталонный автомат, то гарантируется, что на все остальные последовательности он будет реагировать, как эталон.

Примерами методов построения проверяющих тестов являются метод Василевского (имеет много модификаций), transition tour.

**2 ПЛАТФОРМА JUNIT 5**

JUnit 5 представляет собой программный комплекс для тестирования программного обеспечения на языке Java.

**ВЫВОДЫ**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ**

1 Гилл А. Введение в теорию конечных автоматов / А. Гилл. – М. : Издательство Наука, 1966. – 272с.

2 Агибалов Г. П. Лекции по теории конечных автоматов / Г. П. Агибалов, А. М. Оранов. – Томск : Издательство ТГУ, 1984. – 185 с.

3 Дорофеева

4 Chow

5 JUnit 5 User Guide – URL: https://junit.org/junit5/docs/current/user-guide/#writing-tests-dynamic-tests