Лекция №1
Строганов

Формальное определение тестирования

Тестирование является одним из основных инструментов обеспечения качества продуктов. Оно также является одной из фаз разработки (~40%).

Программу можно рассматривать как некоторую математическую формулу
$f = f_1*f_2*f_3*\ldots*f_4$
Некоторую функцию f можно представить как суперпозицию её операторов $f_1..f_n$. Тестирование сводится к доказательству равенства (равенство программы своей спецификации)

Методы доказательства тождеств:

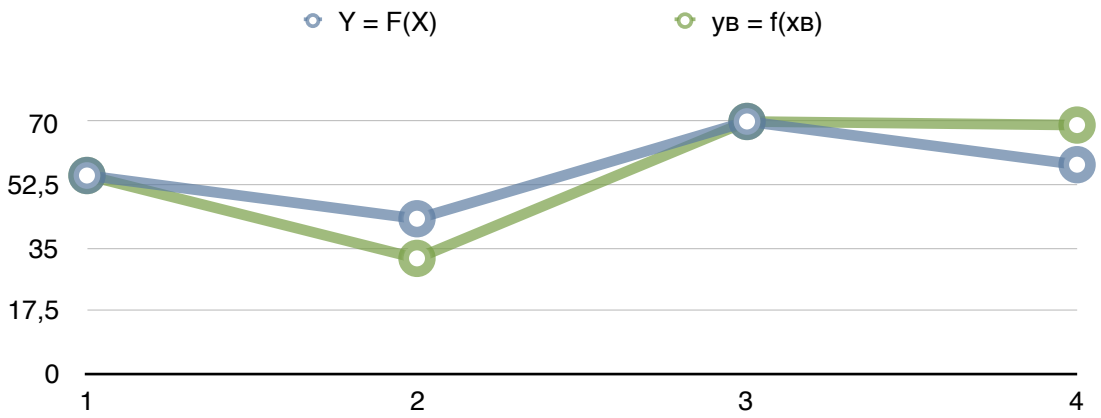
1. Формальных преобразования – используя формальные процедуры и аксиом преобразовать левую и правую часть к одному и тому же виду. Позволяет избежать работы с большими наборами значений переменных, но к тестированию программ этот подход не применим.
2. Интерпретационный – тождество проверяется на всех возможных наборах переменных. Недостаток в том, что наборов может быть очень много.

Основные понятия тестирования:

- Под отладкой программы понимается процесс поиска, локализации и исправления ошибок программы. Если программа не содержит синтаксических ошибок, она может быть скомпилирована и выполнена, то она в любом случае выполняет какую-то функции (отображение множества входных данных на множество выходных). При этом во время выполнения программы ВУ на своих ресурсах до-определяет частично-определённую функцию. Судить о правильности работы этой функции можно сравнивая спецификацию функции с её вычислениями.

Организация тестирования

Перед началом тестирования задаются множество входных данных X и множество ожидаемых результатов Y. В процессе тестирования, подавая на вход элементы Xв



получаем некоторые выходные элементы Yв. Тест считается пройденным, если всё множество точек $(x_в;y_в) \in (X; Y)$. Если обнаруживаются точки, не принадлежащие ожидаемой траектории, необходимо запустить процедуру отладки.

Подходы к отладке:

1. Выполнение программы в уме
2. Добавление в программу операторов отладочной печати или протоколированная промежуточных результатов
3. Пошаговое выполнение программы
4. Выполнение программы с заранее заданными точками останова (Breakpoints)
5. Реверсивное выполнение программы

Тестирование – это процесс выполнения программного обеспечения, системы или компонента в условии анализа или записи получаемых результатов с целью проверки или оценки некоторых свойств тестируемого объекта.

Тестирование – процесс анализа пункта требований к программному обеспечению с целью фиксацией различия между существующей программой и требуемой при экспериментальной проверке соответствующего пункта требований.

Тестирование (IEEE Std 829-1983) – это контролируемое выполнение программы на конечном множестве тестовых данных и анализ результатов этого выполнения для поиска ошибок.

Тестирование включает 3 фазы:

1. Создание тестового набора вручную или с помощью автоматической генерации
2. Прогон программы на ранее определённых тестовых данных
3. Оценка результатов тестирования с целью обнаружения ошибок

Основная проблема тестирования – определение минимального набора тестов для проверки правильности программ.

Проблемы:

1. Невозможно тестировать на всех вариантах значений входных данных (их может быть неограниченно много)
2. Невозможно тестировать на всех путях выполнения программы
3. Нет способов гарантировать завершение программы на любом тесте

Задача выбора набора конечных тестов для тестирования в общем случае неразрешима. Поэтому на практике используются частные решения этой задачи.

Критерии тестирования

Используются для оценки качества тестов. Существует абстрактная модель «Идеальный критерий тестирования»:

1. Достаточность – т.е. определять когда набора тестов достаточно для полной проверки программ
2. Полноту тестового набора – для каждой ошибки существует покрывающий её тест
3. Надёжность – 2 разных набора тестов, отвечающих этому критерию должны одновременно либо обнаруживать, либо не обнаруживать одну и ту же ошибку в программе.
4. Легко проверяемый

Для любых нетривиальных программ полного и надёжного критерия не бывает. Из-за этого используются некоторые частные решения.

Классы критериев тестирования

1. Структурные (белый ящик) – используется информация о структуре программ (хороший, если выполняются все ветви программы)
2. Функциональные (чёрный ящик) – по спецификации требований (хороший, если выполняются все функции спецификации)
3. Стохастический – проверка наличия заданных статистических свойств у тестируемого приложения с помощью метода проверки ?
4. Мутационные

Структурные критерии:

Используется модель программы – белый ящик (предполагается, что известен код или хотя бы его модель в виде графа потоков управления). Структурные критерии используются при модульном и мутационном тестировании.

Варианты структурных критериев:

- Критерий тестирования команд – набор тестов должен обеспечить прохождения каждой команды не меньше одного раза (применяется в больших системах, когда другие критерии применить невозможно)
- Критерий тестирования ветвей – набор тестов должен обеспечить прохождение каждой ветви не менее одного раза (достаточно экономичный критерий, так как путей на графе потоков управления не так много)
- Критерий тестирования путей – если путь на графе включает циклический оператор, то тест строится так, что бы проходилась только одна итерация цикла

Функциональные критерии:

Рассматривают программный продукт в целом и поэтому учитывают взаимодействие приложения с его окружением. Основная проблема – трудоёмкость (слишком много тестов).

Варианты функциональных критериев:

- Тестирование пунктов спецификации – каждый пункт спецификации проверяется по меньшей мере один раз
- Тестирование классов входных данных (области эквивалентности)
- Тестирование классов выходных данных – аналогично областям эквивалентности входных данных, можно построить их для выходных

- Тестирование функций – каждая функция должна быть проверена хотя бы один раз (могут возникать интеграционные ошибки)

Стохастические критерии:

Применяется для тестирования сложных модулей с большими объемами входных данных, которые нельзя разбить на области эквивалентности. Разрабатываются программы-имитаторы случайных входных данных.

1. Генерация набора входных данные
2. Независимое получение набора ожидаемых выходных данных
3. Тестирование
   - 3.1. Детерминированны – сравнение ожидаемых с фактическими выходными данными
   - 3.2. Стохастический (если невозможно получить ожидаемые выходные данные) – сравниваются не фактические значения, а их законы распределения. В качестве используются методы проверки гипотез (Стьюдента, хи-квадрат)

Мутационные критерии:

Основывается на том допущении, что программисты пишут «почти» правильные программы (программа не содержит больших логических ошибок, а только незначительные опечатки и недосмотры?). Мутанты – это программы, которые отличаются друг от друга мутациями.

Изначально в программу P вносятся некоторые искусственные мутации (мелкие ошибки в программе). Программа P и её набор мутантов тестируется на одном и том же наборе тестов. Если набор тестов подтверждает правильность программы P и также выявляет все искусственные ошибки в программах мутантах, то этот тест является качественным. Если не все мутации выявлены, то набор тестов нужно расширить.