УДК 004.031.4

ПРИМЕНЕНИЕ ОРТОГОНАЛЬНЫХ МАССИВОВ В ТЕХНИКЕ ПОПАРНОГО ТЕСТИРОВАНИЯ

Лобанкова Т.С., маг., Лобанков А.А., к.т.н., доц.
Самарский университет, г. Самара, Российская Федерация

Реферат. *В статье рассмотрено применение ортогональных массивов при составлении тестовых случаев по технике попарного тестирования, предложено несколько вариантов реализации алгоритма для проверки матрицы на ортогональность, описывается алгоритм нахождения тестовых случаев по технологии попарного тестирования.*

Ключевые слова: попарное тестирования, тест-дизайн, ортогональный массив, Prolog.

Технология попарного тестирования – это методика создания набора тестовых случаев, позволяющая изучать отношения между элементами программы, влияющими на ее работу [1]. Она предполагает создание тестовых сценариев, в которых проверяются все возможные комбинации пар значений, использующихся при работе программы. Однако просто перебирать все возможные комбинации может быть не только затруднительно, но и неэффективно по времени и ресурсам. Именно для этого используется идея попарного тестирования. Это позволяет значительно сократить количество тест-кейсов, которое нужно протестировать, при этом сохраняя достаточное покрытие для выявления большинства ошибок.

Алгоритм нахождения тестовых случаев по технологии попарного тестирования можно описать следующим образом [2]:

1. Определить множество параметров, влияющих на работу программы. Как правило, параметры могут иметь разные типы данных и логические связи между собой.
2. Для каждого параметра определить множество возможных значений. Это может быть, например, набор значений параметра «режим работы», который может принимать значения «автоматический», «ручной» и «смешанный».
3. Составить таблицу взаимодействия параметров, в которой перечислены все возможные комбинации значений параметров.
4. Для каждой строки таблицы создать тестовый случай, используя значения параметров из этой строки. Получится набор тестовых случаев, который покрывает все комбинации значений параметров.
5. Протестировать программу на созданных тестовых случаях и анализировать результаты. При необходимости можно внести изменения в программу и повторить тестирование.

Важно отметить, что использование алгоритма попарного тестирования не гарантирует 100 %-го полного покрытия всех возможных вариантов программной работы. Однако он является эффективным инструментом для выявления большинства возможных ошибок и недочетов, а также экономит время на тестирование и снижает риски ошибок.

Для попарного тестирования используют алгоритмы, которые базируются на построении ортогональных матриц, или алгоритмы All-Pairs.

Давайте подробнее остановимся ортогональных массивах. Ортогональным называется такой массив, в котором каждый столбец и каждая строка являются линейно независимыми. То есть нет такой строки или столбца, которые можно представить, как линейную комбинацию других строк или столбцов.

Наиболее часто ортогональные массивы используются в экспериментах для оптимизации и испытаний систем или процессов. В ортогональном массиве каждая колонка представляет один фактор, который влияет на результат теста, а каждая строка представляет одну конфигурацию факторов для проведения теста.

Использование ортогональных массивов позволяет существенно сократить количество рассматриваемых тестовых случаев. Например, пусть нам необходимо составить тестовые случаи для трех параметров (F1, F2, F3), каждый из которых может принимать три различных значения. В этом случае полная матрица тестовых случаев будет состоять из 3 * 3 = 27 возможных комбинаций. Использование ортогональных массивов позволит сократить количество проверяемых комбинаций до 6. Пример такого ортогонального массива представлен в таблице 1.

Таблица 1 – Пример ортогонального массива для трех параметров.

F1	F2	F3
A1	B2	C3
B1	C2	A3
C1	A2	B3
A1	B2	B3
B1	C2	C3
C1	A2	A3

Для того, чтобы проверить, что массив является ортогональным необходимо выполнить следующие шаги [2]:

1. Возьмите любые два различных столбца и умножьте их попарно.
2. Сложите полученные произведения.
3. Если сумма равна нулю, то массив является ортогональным.

Пример программы на Python, реализующий этот алгоритм проверки приведен ниже.

```
def is_orthogonal(arr):
    n = len(arr)
    for i in range(n):
        if len(arr[i]) != n:
            return False
    for i in range(n - 1):
        for j in range(i + 1, n):
            dot_product = sum([arr[k][i] * arr[k][j] for k in range(n)])
            if dot_product != 0:
                return False
    return True
```

Эта функция возвращает True, если массив является ортогональным, и False – в противном случае. К сожалению, данная программа работает только для числовых массивов.

Давайте рассмотрим немного модифицированный алгоритм проверки массива на ортогональность, написанный на языке логического программирования SWI-Prolog. Prolog основан на математической теории исчисления предикатов, которая описывает отношения, функции и операции над множествами. SWI-Prolog – это одна из самых популярных и широко используемых реализаций языка Prolog [3].

```
check_orthogonal(M) :-
    transpose(M, MT),
    check_rows(M),
    check_rows(MT).

check_rows([]).
check_rows([H|T]) :-
    check_rows(T, H),
    check_rows(T).

check_rows([], _).
check_rows([H|T], H0) :-
    dot_product(H, H0, 0),
    check_rows(T, H0).

dot_product([], [], Acc) :- Acc =:= 0.
dot_product([H1|T1], [H2|T2], Acc) :-
    Acc1 is Acc + H1 * H2,
    dot_product(T1, T2, Acc1).
```

Для запуска данной программы необходимо вызвать предикат check_orthogonal (M), который принимает на вход массив M. Он возвращает True, если массив является ортогональным, и False в противном случае. Предикат transpose (M, MT) – транспонирует нашу матрицу M и сохраняет результат в переменную MT. Предикат check_rows (M) проверяет, что каждая строка в матрице M является линейно независимой, а предикат check_rows (MT) проверяет что каждый столбец исходной матрицы M является линейно независимым. Для проверки строк и столбцов матрицы на линейную независимость используется дополнительный предикат dot_product/3, который вычисляет скалярное произведение строки или столбца матриц и переданного ему элемента, и возвращает True, если оно равно 0.

Таким образом, обе реализации нашего алгоритма проверки массива на ортогональность работают только с числовыми массивами. То есть для проверки массива, элементы которого являются строками, надо каждой строке поставить в соответствие число.

Преимущества применения ортогональных массивов в тестировании:

1. Сокращение времени и затрат на тестирование, поскольку ортогональные массивы позволяют проводить тестирование с минимальным количеством тестовых примеров.
2. Увеличение эффективности тестирования, поскольку позволяет быстро и эффективно обнаруживать ошибки.
3. Сокращение вероятности ошибок при разработке системы.
4. Универсальность использования ортогональных массивов, они могут применяться в различных областях, таких как программное обеспечение, конструирование, маркетинг и т. д.

Недостатки применения ортогональных массивов в тестировании:

1. Ортогональные массивы не могут быть применены для тестирования всех возможных комбинаций переменных, поскольку максимальное количество переменных ограничено.
2. Ортогональные массивы могут быть сложными для понимания и использования, особенно для непрофессионалов.
3. Ограничение на количество переменных может привести к пропуску некоторых сценариев тестирования.

Список использованных источников

1. Казючиц, П. Ю. Использование ортогональных массивов при проверке качества программного обеспечения / П. Ю. Казючиц, Ю. А. Скудняков // XXII международная научно-техническая конференция «Информационные системы и технологии» ИСТ-2016 (Нижний Новгород, 22 апреля 2016 г.). – Нижний Новгород : Нижегородский государственный технический университет им. Р. Е. Алексеева, 2016. – С. 244..
2. Copeland, L. A Practitioner's Guide to Software Test Design / L. Copeland. – Miami, FL : Artech House, 2004. – 355 S.
3. SWI-Prolog [Electronic resource. – Mode of access: https://www.swi-prolog.org. – Date of access: 10.03.2023.

УДК 339.19

ВНЕДРЕНИЕ ЭЛЕМЕНТОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ФГИС «МЕРКУРИЙ»

Орехова А.А., студ., Васильева М.В., к.э.н., доц.

Псковский государственный университет, г. Псков, Российская Федерация

Реферат. *В статье рассмотрена проблематика реализации ветеринарного и таможенного контроля с применением элементов искусственного интеллекта в системе «Меркурий». Автоматизация процессов проверки и принятия документов имеет важное значение как для таможенных органов, так и для работников ветеринарного контроля, обеспечивая национальную безопасность, прослеживаемость продуктов и чистоту рынка продуктов животного происхождения.*

Ключевые слова: ФГИС «Меркурий», ветеринарный и таможенный контроль.