

РЕАЛИЗАЦИЯ АЛГОРИТМА ОПРЕДЕЛЕНИЯ МЕРЫ РАЗЛИЧИЯ ДВОИЧНЫХ ТЕСТОВЫХ НАБОРОВ

Ильина А.Ю., магистрант

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Ярмолик В.Н. – д-р. техн. наук, профессор

Аннотация. Рассматривается алгоритм определения меры различия $AD(T_i, T_k)$ двоичных тестовых наборов равной длины. Описывается реализация данного алгоритма с использованием битовых масок.

Ключевые слова. Мера различия, расстояние, тестовый набор, управляемое вероятностное тестирование.

Вероятностное тестирование (random testing) – одна из распространенных технологий тестирования по методу черного ящика [1, 2]. При вероятностном тестировании на исследуемый объект подаются случайные или псевдослучайные входные последовательности [1]. Таким образом, тестовые случаи выбираются, генерируются случайным образом из набора входных данных [3, 4, 5]. Вероятностное тестирование широко применяется на практике [1, 6].

Для повышения эффективности вероятностных тестов используется управляемое вероятностное тестирование (controlled random testing) [2, 5]. Под управляемым вероятностным тестированием понимается случайная тестовая последовательность, в которой очередной тестовый набор формируется с учетом ранее сгенерированных предыдущих наборов [1]. Ключевая особенность контролируемого генерирования случайных тестовых наборов – информация, которая извлекается в виде определенных характеристик (метрик) из ранее сгенерированных тестовых наборов. Эта информация используется для формирования очередного тестового набора [1, 5, 7, 8].

Главный недостаток управляемого вероятностного тестирования заключается в сложности определения очередного тестового набора по отношению к предыдущим наборам теста [1, 2]. В управляемом вероятностном тестировании очередной тестовый набор формируется максимально отличным от всех наборов, сгенерированных ранее [1].

При контролируемом генерировании случайных тестовых наборов в качестве меры различия (distance measure) чаще всего используются расстояние Хэмминга и декартово (евклидово) расстояние [1]. Предложена новая мера различия $AD(T_i, T_k)$ двух символьных тестовых наборов на основе интервала, используемого в теории строки цепи последовательных событий [7, 8]. В качестве меры различия рассчитывается расстояние $AD(T_i, T_k)$ между тестовыми наборами T_i и T_k , использующее характеристику интервала и основанное на определении независимых пар одинаковых (тождественных) символов, принадлежащих двум наборам, и вычислении интервалов между ними. Предложенная мера различия расширяет возможности генерирования тестовых последовательностей при генерировании управляемых вероятностных тестов [8].

Для двоичных тестовых наборов задача вычисления метрики $AD(T_i, T_k)$ сводится к задаче о назначении. Данная задача решается с помощью венгерского алгоритма, вычислительная сложность которого оценивается соотношением $O(n^4)$. Как альтернатива венгерскому алгоритму предложен алгоритм определения $AD(T_i, T_k)$ – алгоритм вычисления рассматриваемой меры, вычислительная сложность которого существенно меньше и имеет оценку $O(n^2)$ [8].

Для случая двоичных данных равной длины исходными являются тестовые наборы T_i и T_k , состоящие из n данных $t_{i,j}, t_{k,r} \in \{0, 1\}$, $j, r \in \{0, 1, \dots, n-1\}$.

Сущность предложенного алгоритма вычисления меры различия $AD(T_i, T_k)$ заключается в определении количества тождественных пар данных $t_{i,j}$ и $t_{k,r}$ тестовых наборов T_i и T_k , имеющих определенное значение расстояния $D(t_{i,j}, t_{k,r})$. Учитывая, что длина тестовых наборов T_i и T_k одинакова и равна n , значение $D(t_{i,j}, t_{k,r})$ равняется минимальному значению из двух величин $|j-r|$ и $n-|j-r|$. Тогда необходимо рассмотреть только $\lfloor n/2 \rfloor + 1$ возможных значений $D(t_{i,j}, t_{k,r}) = 0 \div \lfloor n/2 \rfloor$, где квадратные скобки обозначают целую часть числа [8].

Генерируются циклические сдвиги T_i влево и вправо с последующим определением количества совпадающих данных, имеющих одинаковые индексы $j=r$. В результате операций циклического сдвига T_i индекс j для каждого данного $t_{i,j}$ набора T_i принимает все возможные значения $j \in \{0, 1, \dots, n-1\}$. Таким образом, каждому данному $t_{k,r}$ набора T_k последовательно сопоставляются все данные $t_{i,j}$ и анализируются на предмет совпадения. Совпадение данных $t_{i,j}$ и $t_{k,r}$ свидетельствует о наличии пары тождественных данных с расстоянием $D(t_{i,j}, t_{k,r})$, равным количеству циклических сдвигов набора T_i . Независимость пар тождественных данных обеспечивается исключением данных, входящих в выявленную пару, из дальнейшего рассмотрения [8].

Алгоритм определения меры различия $AD(T_i, T_k)$ двоичных данных равной длины состоит из следующих шагов [7, 8]:

1 В исходных тестовых наборах T_i и T_k определяются совпадающие данные $t_{i,j}$ и $t_{k,r}$, имеющие расстояние $D(t_{i,j}, t_{k,r}) = 0$ и число совпадающих данных $q(0)$. Выделенные данные исключаются из дальнейшего рассмотрения. В случае, когда $q(0) = n$, т. е. при полном совпадении данных набора T_i с данными набора T_k , задается $q(t) = 0$, и осуществляется переход к пункту 5.

2 Если количество данных n (длина тестовых наборов) является нечетным числом, то последовательно для $v = 1, 2, \dots, [n/2]$ формируются циклические сдвиги тестового набора T_i относительно набора T_k на v позиций влево и вправо. Следует отметить, что операция циклического сдвига эквивалентна уменьшению и увеличению значения индекса j на величину v по модулю n . Далее определяется количество $q(v)$ тождественных пар данных с совпадающими индексами, а именно с модифицированным в результате сдвига $j = (j \pm v) \bmod n$ и индексом r . Величина $q(v)$ определяет количество пар данных $t_{i,j}$ и $t_{k,r}$, имеющих расстояние $D(t_{i,j}, t_{k,r}) = v$. При рассмотрении очередного значения сдвига v набора T_i выделенные тождественные данные исключаются из дальнейшего рассмотрения. Выполнение данного шага прекращается в случае, если количество пар эквивалентных данных, выявленных в наборах T_i и T_k , достигло величины n (т. е. все данные в наборах совпали) и после задания $q(t) = 0$ осуществляется переход к пункту 5.

3 Если n четное число, выполняются действия шага 2, однако вместо пары циклических сдвигов тестового набора T_i на $v = [n/2]$ влево и вправо формируется единственный циклический сдвиг на $v = [n/2]$ позиций влево либо вправо.

4 Определяется количество $q(t)$ пар данных $t_{i,j}$ и $t_{k,r}$ наборов T_i и T_k , которые не участвовали в формировании пар тождественных данных на предыдущих шагах алгоритма. При равенстве числа данных $t_{i,j}$ и $t_{k,r}$ в наборах T_i и T_k для каждого из них существует пара либо тождественных данных, либо несовпадающих [8].

5 Вычисляется значение меры различия $AD(T_i, T_k)$. Для двоичных тестовых наборов равной длины n в случае несовпадения данных расстояние между ними считается равным значению $[n/2]$, поэтому значение меры различия $AD(T_i, T_k)$ вычисляется согласно выражению [8]

$$AD(T_i, T_k) = q(t) [n/2] + \sum_{v=1}^{[n/2]} v \cdot q(v). \quad (1)$$

Полученное значение $AD(T_i, T_k)$ является искомой величиной меры различия наборов T_i и T_k .

В таблице 1 представлено применение алгоритма определения $AD(T_i, T_k)$ для случая восьмибитных двоичных тестовых наборов $T_i = 00010111$ и $T_k = 11101010$.

Таблица 1 – Применение алгоритма определения меры различия $AD(T_i, T_k)$ для восьмибитных двоичных тестовых наборов $T_i = 00010111$ и $T_k = 11101010$.

| Номер шага и величина сдвига влево или вправо v | Данные тестового набора | Данные тестового набора с индексами позиций j для t_i и r для t_k | | | | | | | | Количество совпадений на данном шаге $q(v)$ и после выполнения данного шага $ov(v)$ |
|---|--|---|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| $j = (j - v) \bmod 8 = r$ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| Шаг 0, $v_0 = 0$ | $t_{i,j}$ | 0 ₀ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | $q(v_0) = 1,$ $ov(v_0) = 1$ |
| | $t_{k,r}$ | 1 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 1, $v_{left,1} = 1$ | $t_{i,j (left,1)}$ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₀ | $q(v_{left,1}) = 4,$ $ov(v_{left,1}) = 5$ |
| | $t_{k,r}$ | 1 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 2, $v_{right,1} = 1$ | $t_{i,j (right,1)}$ | 1 ₇ | 0 ₆ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | $q(v_{right,1}) = 1,$ $ov(v_{right,1}) = 6$ |
| | $t_{k,r}$ | 1 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 3, $v_{left,2} = 2$ | $t_{i,j (left,2)}$ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₆ | 0 ₁ | $q(v_{left,2}) = 0,$ $ov(v_{left,2}) = 6,$ |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 4, $v_{right,2} = 2$ | $t_{i,j (right,2)}$ | 1 ₆ | 1 ₇ | 0 ₆ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | $q(v_{right,2}) = 0,$ $ov(v_{right,2}) = 6$ |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 5, $v_{left,3} = 3$ | $t_{i,j (left,3)}$ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₆ | 0 ₁ | 0 ₂ | $q(v_{left,3}) = 0,$ $ov(v_{left,3}) = 6$ |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 6, $v_{right,3} = 3$ | $t_{i,j (right,3)}$ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₆ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | $q(v_{right,3}) = 1,$ $ov(v_{right,3}) = 7$ |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 7, $v_{left,4} = 4$ ($v_{right,4} = 6$) | $t_{i,j (left,4)}$ ($t_{i,j (right,4)}$) | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₆ | 0 ₁ | 0 ₂ | 1 ₃ | $q(v_{left,4}) = q(v_{right,4}) = 0,$ $ov(v_{left,4}) = ov(v_{right,4}) = 7$ |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |

В таблице 1 подчеркнуты и выделены полужирным шрифтом совпадающие на текущем шаге данные тестовых наборов, неиспользованные ранее. Зачеркнуты ячейки, содержащие данные, совпавшие на более ранних шагах алгоритма.

Алгоритм определения меры различия $AD(T_i, T_k)$ может быть реализован с использованием битовых масок. Битовая маска – последовательность бит, применяемая для маскирования, т. е. выбора отдельных бит из двоичных данных.

В описываемой реализации алгоритма определения $AD(T_i, T_k)$ используются две битовые маски M_i и M_k для выполнения операций над тестовыми наборами T_i и T_k . Длины масок равны длине тестовых наборов и составляют n . Тестовые наборы состоят из данных $t_{i,j}, t_{k,r} \in \{0, 1\}$ маски – из данных $m_{i,j}, m_{k,r} \in \{0, 1\}$, причем $j, r \in \{0, 1, \dots, n - 1\}$.

Шаги алгоритма определения меры различия $AD(T_i, T_k)$ представляются следующим образом:

1 Искомая мера различия AD инициализируется нулем, а данные маски M_i , соответствующей тестовому набору T_i – единицами. Все единицы в маске M_i означают, что данные набора T_i не были использованы в тождественных парах ранее.

2 Вычисляется вспомогательный набор $result$. Значение $result$ рассчитывается по формуле $result = NOT(T_i XOR T_k)$. Данное выражение означает следующее. Операция исключающее ИЛИ применяется к тестовым наборам T_i и T_k для выделения тождественных пар данных. Таким образом получается набор данных длины n , где совпадающие пары данных тестовых наборов T_i и T_k обозначены нулями, а несовпадающие – единицами на соответствующих позициях. К полученному результату применяется инверсия (операция логическое НЕ), и совпадающие пары данных тестовых наборов T_i и T_k в конечном наборе обозначены единицами, а несовпадающие – нулями.

3 Количество совпавших пар данных $overlap$ инициализируется значением, равным количеству единиц в найденном на шаге 2 наборе $result$.

4 Находится новая маска M_i по формуле $M_i = M_i XOR result$. Операция исключающее ИЛИ обнуляет значения единиц, обозначающих найденные пары совпадающих данных в маске, что позволяет отметить их как исключенные из рассмотрения. Маска M_k , соответствующая тестовому набору T_k , инициализируется найденным новым значением маски M_i , поскольку до текущего шага сдвигов не производилось, и значения масок M_i и M_k совпадают.

5 Если длина тестовых наборов n – нечетное число, последовательно для $v = 1, 2, \dots, [n / 2]$ и направлений влево и вправо выполняется процедура нахождения количества пар тождественных данных, находящихся на определенном расстоянии друг от друга. Выполнение данного шага прекращается, если маски M_i и M_k полностью обнулились, т. е. все данные наборов были использованы. Следует отметить, что обнуление обеих масок происходит одновременно.

6 Если длина тестовых наборов n – четное число, выполняются действия шага 5, однако последовательно для $v = 1, 2, \dots, [n / 2] - 1$ влево и вправо и для $v = [n / 2]$ влево либо вправо.

7 Находится разность между длиной набора n и количеством совпавших пар данных $overlap$. Значение меры AD увеличивается на произведение найденной разности и значение $[n / 2]$.

Выполнение шагов 5 и 6 алгоритма основано на процедуре нахождения количества пар тождественных данных, находящихся на определенном расстоянии друг от друга. На входе процедура получает тестовые наборы T_i и T_k , соответствующие им маски M_i и M_k , направление сдвига $direction$ (влево или вправо) и количество позиций v , на которые осуществляется сдвиг.

Алгоритм нахождения количества пар тождественных данных, находящихся на определенном расстоянии друг от друга состоит из следующих шагов:

1 Формируются циклические сдвиги $T_{i(direction,v)}$ и $M_{i(direction,v)}$ тестового набора T_i и маски M_i в заданном направлении $direction$ (влево или вправо) на заданное количество позиций v .

2 Вычисляется вспомогательный набор $result$. Значение $result$ рассчитывается по формуле $result = (NOT(T_{i(direction,v)} XOR T_k)) AND M_{i(direction,v)} AND M_k$. Данное выражение, по сути, означает следующее. Операция исключающее ИЛИ применяется к тестовым наборам $T_{i(direction,v)}$ и T_k с целью выделения тождественных пар данных. В результате выполнения операции $T_{i(direction,v)} XOR T_k$ получается набор данных длины n , совпадающие пары данных тестовых наборов $T_{i(direction,v)}$ и T_k обозначены нулями, а несовпадающие – единицами. К полученному результату применяется инверсия. Результатом выполнения операций $NOT(T_{i(direction,v)} XOR T_k)$ является набор данных длины n , где совпадающие пары данных тестовых наборов $T_{i(direction,v)}$ и T_k обозначены уже единицами, а несовпадающие – нулями. Далее к полученному набору применяется операция логического И с масками $M_{i(direction,v)}$ и M_k . Это действие необходимо для исключения из конечного результата тождественных пар данных, уже учтенных на более ранних шагах алгоритма. Поскольку в масках единицами обозначены еще не использованные данные, а нулями – уже учтенные и потому исключенные из рассмотрения, после выполнения операций логического И в значении $result$ единицами обозначаются только пары совпадающих данных, которые не были использованы ранее. Все остальные пары данных в $result$ отмечены нулями. Таким образом, в наборе $result$ единицами обозначаются обнаруженные пары эквивалентных данных, которые не были исключены из рассмотрения, а нулями – все прочие пары данных.

3 Формируются новые маски $M_i^{(direction,v)}$ и M_k с учетом найденных на текущем шаге пар совпадающих данных, которые не были учтены ранее. Новые маски вычисляются следующим образом: $M_i^{(direction,v)} = M_i^{(direction,v)} \text{ XOR result}$; $M_k = M_k \text{ XOR result}$. Операция исключаящее ИЛИ обнуляет значения единиц, обозначающих найденные пары тождественных данных в масках, что позволяет отметить использованные на текущем шаге данные как исключенные из рассмотрения.

4 Находится количество единиц во вспомогательном наборе result, т. е. количество найденных на текущем шаге пар совпадающих данных, которые не использовались на более ранних шагах.

5 Рассчитывается слагаемое метрики AD, полученное на текущем шаге алгоритма. Для этого количество единиц в наборе result умножается на заданное количество позиций сдвига v, равное расстоянию между тождественными данными. Т. е. количество пар совпадающих данных, найденных на текущем шаге, не исключенных ранее, умножается на расстояние между совпадающими данными.

С помощью описанной процедуры находятся значения, на которые необходимо увеличить метрику AD и количество совпавших пар данных overlap. Процедура вычисляет маски M_i и M_k , необходимые для следующих шагов алгоритма. Следует учесть, что в процедуре находится $M_i^{(direction,v)}$, т. е. маска M_i , циклически сдвинутая на v позиций влево или вправо.

В таблице 2 приведено применение реализации алгоритма определения $AD(T_i, T_k)$ с использованием битовых масок для случая, рассмотренного в таблице 1 – восьмибитные двоичные тестовые наборы $T_i = 00010111$ и $T_k = 11101010$.

Таблица 2 – Применение реализации алгоритма определения меры различия $AD(T_i, T_k)$ с использованием битовых масок для восьмибитных двоичных тестовых наборов $T_i = 00010111$ и $T_k = 11101010$.

| Номер шага и величина сдвига влево или вправо v | Данные тестового набора, маски или вспомогательного набора result | Данные тестового набора, маски или вспомогательного набора result с индексами позиций j для t_i и r для t_k | | | | | | | | Количество совпадений на данном шаге $q(v)$ и после выполнения данного шага $ov(v)$ |
|---|---|---|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| $j = (j - v) \text{ mod } 8 = r$ | | | | | | | | | | |
| Шаг 0, $v_0 = 0$ | $t_{i,j}$ | 0 ₀ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | $q(v_0) = 1,$ $ov(v_0) = 1$ |
| | $t_{k,r}$ | 1 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| | $m_{i,j(k,r)}$ | 1 ₀ | 1 ₁ | 1 ₂ | 1 ₃ | 1 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | |
| | result _{r(0)} | 0 ₀ | 0 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| Шаг 1, $v_{left,1} = 1$ | $t_{i,j(left,1)}$ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₀ | $q(v_{left,1}) = 4,$ $ov(v_{left,1}) = 5$ |
| | $m_{i,j(left,1)}$ | 1 ₁ | 1 ₂ | 1 ₃ | 1 ₄ | 1 ₅ | 0 ₆ | 1 ₇ | 1 ₀ | |
| | $t_{k,r}$ | 1 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| | $m_{k,r}$ | 1 ₀ | 1 ₁ | 1 ₂ | 1 ₃ | 1 ₄ | 1 ₅ | 0 ₆ | 1 ₇ | |
| Шаг 2, $v_{right,1} = 1$ | $t_{i,j(right,1)}$ | 1 ₇ | 0 ₆ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | $q(v_{right,1}) = 1,$ $ov(v_{right,1}) = 6$ |
| | $m_{i,j(right,1)}$ | 1 ₇ | 0 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 0 ₄ | 0 ₅ | 0 ₆ | |
| | $t_{k,r}$ | 1 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| | $m_{k,r}$ | 1 ₀ | 1 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 1 ₅ | 0 ₆ | 0 ₇ | |
| Шаг 3, $v_{left,2} = 2$ | $t_{i,j(left,2)}$ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₀ | 0 ₁ | $q(v_{left,2}) = 0,$ $ov(v_{left,2}) = 6,$ |
| | $m_{i,j(left,2)}$ | 1 ₂ | 0 ₃ | 0 ₄ | 0 ₅ | 0 ₆ | 0 ₇ | 0 ₀ | 1 ₁ | |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| | $m_{k,r}$ | 0 ₀ | 1 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 1 ₅ | 0 ₆ | 0 ₇ | |
| Шаг 4, $v_{right,2} = 2$ | $t_{i,j(right,2)}$ | 1 ₆ | 1 ₇ | 0 ₀ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | 1 ₅ | $q(v_{right,2}) = 0,$ $ov(v_{right,2}) = 6$ |
| | $m_{i,j(right,2)}$ | 0 ₆ | 0 ₇ | 0 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 0 ₄ | 0 ₅ | |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| | $m_{k,r}$ | 0 ₀ | 1 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 1 ₅ | 0 ₆ | 0 ₇ | |
| Шаг 5, $v_{left,3} = 3$ | $t_{i,j(left,3)}$ | 1 ₃ | 0 ₄ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₀ | 0 ₁ | 0 ₂ | $q(v_{left,3}) = 0,$ $ov(v_{left,3}) = 6$ |
| | $m_{i,j(left,3)}$ | 0 ₃ | 0 ₄ | 0 ₅ | 0 ₆ | 0 ₇ | 0 ₀ | 1 ₁ | 1 ₂ | |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| | $m_{k,r}$ | 0 ₀ | 1 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 1 ₅ | 0 ₆ | 0 ₇ | |
| Шаг 6, $v_{right,3} = 3$ | $t_{i,j(right,3)}$ | 1 ₅ | 1 ₆ | 1 ₇ | 0 ₀ | 0 ₁ | 0 ₂ | 1 ₃ | 0 ₄ | $q(v_{right,3}) = 1,$ $ov(v_{right,3}) = 7$ |
| | $m_{i,j(right,3)}$ | 0 ₅ | 0 ₆ | 0 ₇ | 0 ₀ | 1 ₁ | 1 ₂ | 0 ₃ | 0 ₄ | |
| | $t_{k,r}$ | 1 ₆ | 1 ₁ | 1 ₂ | 0 ₃ | 1 ₄ | 0 ₅ | 1 ₆ | 0 ₇ | |
| | $m_{k,r}$ | 0 ₀ | 1 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 1 ₅ | 0 ₆ | 0 ₇ | |
| Шаг 6, $v_{right,3} = 3$ | result _{r(right,3)} | 0 ₀ | 0 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 1 ₅ | 0 ₆ | 0 ₇ | |

| | | | | | | | | | | |
|---|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|---|
| Шаг 7, $V_{\text{left},4} = 4$ ($V_{\text{right},4} = 6$) | $t_{i,j} (\text{left},4)$ ($t_{i,j} (\text{right},4)$) | 0₄ | 1₅ | 1₆ | 1₇ | 0₆ | 0 ₁ | 0₂ | 1₃ | $q(V_{\text{left},4}) =$ $= q(V_{\text{right},4}) = 0,$ $ov(V_{\text{left},4}) =$ $= ov(V_{\text{right},4}) = 7$ |
| | $m_{i,j} (\text{left},4)$ ($m_{i,j} (\text{right},4)$) | 0 ₄ | 0 ₅ | 0 ₆ | 0 ₇ | 0 ₀ | 1 ₁ | 0 ₂ | 0 ₃ | |
| | $t_{k,r}$ | 1₆ | 1 ₁ | 1₂ | 0₃ | 1₄ | 0₅ | 1₆ | 0₇ | |
| | $m_{k,r}$ | 0 ₀ | 1 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 0 ₅ | 0 ₆ | 0 ₇ | |
| | $result_r (\text{left},4)$ ($result_r (\text{right},4)$) | 0 ₀ | 0 ₁ | 0 ₂ | 0 ₃ | 0 ₄ | 0 ₅ | 0 ₆ | 0 ₇ | |

По сравнению с таблицей 1, в таблице 2 помимо тестовых наборов T_i и T_k рассмотрены также маски M_i и M_k и вспомогательный набор $result$.

Как и в таблице 1, в таблице 2 совпадающие на текущем шаге данные тестовых наборов, неиспользованные ранее, подчеркнуты и выделены полужирным шрифтом. Таким же образом отмечены и соответствующие данные масок и результатов выполнения шагов. Зачеркнуты ячейки, содержащие совпавшие на более ранних шагах алгоритма данные. Следует отметить, что нулевые данные в масках соответствуют зачеркнутым ячейкам, содержащим исключенные из дальнейшего рассмотрения данные тестовых наборов $t_{i,j}$ и $t_{k,r}$. Единичные данные в масках соответствуют не зачеркнутым ячейкам, содержащим доступные для рассмотрения данные.

Рассмотрена реализация алгоритма определения меры различия $AD(T_i, T_k)$ для случая двоичных тестовых наборов равной длины с использованием битовых масок.

Совпадающие данные идентифицируются с помощью логических операций исключающее ИЛИ (XOR), НЕ (NOT), И (AND) над тестовыми наборами T_i , T_k и масками M_i , M_k . Над тестовым набором T_i и соответствующей ему маской M_i выполняются операции циклического сдвига.

При определении совпадающих данных в тестовых наборах T_i и T_k единичное значение в маске указывает на доступность данного под тем же номером в соответствующем тестовом наборе. Нулевое значение в маске сигнализирует об исключении данного соответствующего набора из рассмотрения по причине его использования в тождественной паре. Маски M_i и M_k – индикаторы использования данных тестовых наборов в совпадающих парах. Когда все данные задействованы в тождественных парах, обе маски одновременно обнуляются. Последовательное формирование циклических сдвигов тестового набора T_i относительно набора T_k останавливается при равенстве значения любой из масок (обеих масок) нулю.

Значение меры различия AD и количества использованных в тождественных парах данных $overlap$, необходимое для вычисления итогового значения меры различия AD , последовательно накапливаются в шагах алгоритма. Отдельное слагаемое значения $overlap$ на определенном шаге вычисляется как количество единиц в наборе бит, что может быть реализовано на основе сдвигов и/или логической операции И.

Двоичные тестовые наборы данных равной длины – распространенный случай требуемых на практике тестовых данных. В рассмотренной реализации алгоритма определения меры различия $AD(T_i, T_k)$ используются сдвиги чисел и логические операции XOR, NOT, AND над числами. Вычислительная сложность выполнения сдвигов и логических операций в сравнении, например, с реализацией алгоритма на массивах, крайне низкая. Следует отметить, что в большинстве современных вычислительных систем используемые в алгоритме битовые сдвиги и побитовые логические операции реализованы на аппаратном уровне.

Список использованных источников:

1. Ярмолик, В.Н. Контроль и диагностика вычислительных систем / В.Н. Ярмолик. – Минск : Бестпринт, 2019. – 387 с.
2. Леванцевич, В.А. Многократное управляемое вероятностное тестирование / В.А. Леванцевич, В.Н. Ярмолик // Доклады БГУИР. – 2019. – № 3 (121). – С. 65-69.
3. A cost-effective software testing strategy employing online feedback information / Z.Q. Zhou [et al.] // Information Sciences. – 2018. – Vol. 422. – P. 318-335.
4. An empirical comparison of combinatorial testing, random testing and adaptive random testing / H. Wu [et al.] // IEEE Transactions on Software Engineering. – 2020. – Vol. 46, № 3. – P. 302-320.
5. Ильина, А.Ю. Управляемое вероятностное тестирование / А.Ю. Ильина // Современная наука: актуальные вопросы, достижения и инновации: сборник статей XXVIII Международной научно-практической конференции, Пенза, 10 декабря 2022 г. : в 2 ч. / МЦНС «Наука и Просвещение»; редкол.: Г.Ю. Гуляев [и др.]. – Пенза, 2022. – Ч. 1. – С. 28-30.
6. A Survey on Adaptive Random Testing / R. Huang [et al.] // IEEE Transactions on Software Engineering. – 2021. – Vol. 47, № 10. – P. 2052-2083.
7. Ярмолик, В.Н. Мера отличия для управляемых вероятностных тестов / В.Н. Ярмолик, Н.А. Шевченко, В.В. Петровская // Доклады БГУИР. – 2022. – Т. 20, № 6. – С. 52-60.
8. Ярмолик, В.Н. Мера различия для тестовых наборов при генерировании управляемых вероятностных тестов / В.Н. Ярмолик, В.В. Петровская, И. Мрозек // Информатика. – 2022. – № 19 (4). – С. 7-26.

UDC 004.415.533

IMPLEMENTATION OF THE ALGORITHM FOR DETERMINING THE DISTANCE MEASURE OF BINARY TEST SETS

Ilyina A. Y.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Yarmolik V.N. – D. Sc. (Eng.), Professor

Annotation. An algorithm for determining the distance measure $AD(T_i, T_k)$ of binary test sets of equal length is considered. The implementation of this algorithm using bitmasks is described.

Keywords. Distance measure, distance, test set, controlled random testing.