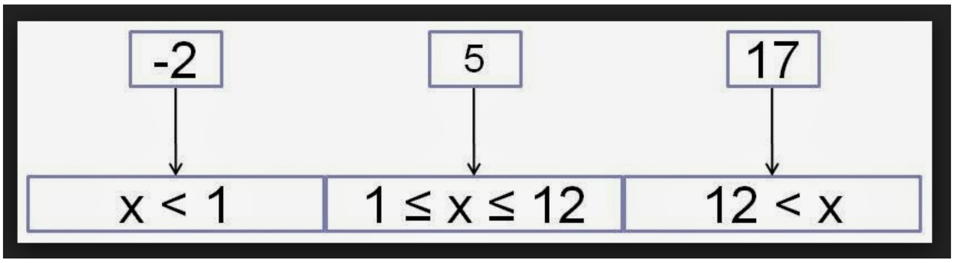
O pewnych problemach analizy wartości brzegowych

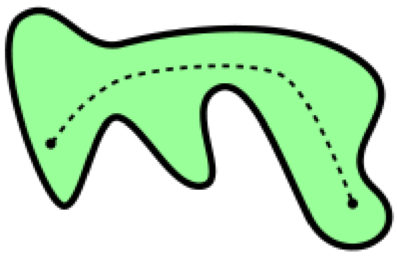


1. Wstęp

Klasa równoważności w testowaniu jest to zbiór danych o podobnym sposobie przetwarzania w oprogramowaniu dla konkretnej funkcjonalności, używanych do przeprowadzenia testu. Wykonanie testu z użyciem kilku elementów zbioru (ISTQB proponuje jedną wartość na klasę), powoduje uznanie całej klasy za poprawną i zwalnia osobę testującą od testowania wszystkich elementów w np. 100-elementowym zbiorze. Rozwinięciem testów z użyciem klas równoważności jest ***testowanie wartości brzegowych***. Wartość brzegowa to wartość znajdująca się wewnątrz, pomiędzy lub tuż przy granicy danej klasy równoważności. Istnieje duże prawdopodobieństwo, że oprogramowanie będzie się błędnie zachowywać dla wartości na krawędziach klas równoważności. **Wartości brzegowe** to na ogół minimum i maksimum (o ile istnieją) klasy równoważności.

2. Sformułowanie problemów

Mówiąc o analizie wartości brzegowych bardzo często mamy na myśli ***wartości brzegowe dyskretne.*** Istnieją w systemach informatycznych funkcje, których dziedziną jest zbiór nieskończony oraz spójny (nie jest sumą dwóch rozłącznych zbiorów otwartych). Obrazowo zbiór spójny może przedstawić tak:



Widać na obrazku, że nie da się wskazać konkretnych wartości brzegowych. Nie jest zatem wykluczona potrzeba przeprowadzenia analizy wartości brzegowych w dziedzinie, która ma takie własności. Ale powstaje pytanie, jak to zrobić, gdy dziedzina funkcji ma postać zbioru otwartego przynajmniej z jednej strony, np. $(2; 10)$, $(1; 5) \times (3; 7)$, $[2; 9)$ itd.? Drugi problem, to jak w przypadku funkcji złożonych znaleźć argumenty, dla których funkcja wewnętrzna „dostanie” wartości brzegowe? Odpowiedzią na te pytania są pewne rozwiązania przedstawione w kolejnych rozdziałach.

3. Propozycja wyznaczenia wartości brzegowych dla funkcji o dziedzinie domkniętej i ciągłej

Załóżmy, że mamy do czynienia z funkcją $f : D \to V$, gdzie $D \subset R$, $V \subset N$. Przepis funkcji f jest następujący:

$$f(x) = \begin{cases} 0, & x \in A_0 \\ 1, & x \in A_1 \\ & \dots \\ n, & x \in A_n \end{cases},$$

gdzie $A_i = [a_i, b_i), V = \bigcup_{i=0}^{n} A_i$ przy czym $A_i \cap A_j = \emptyset$ dla $i \neq j$ oraz $a_i < a_j, b_i < b_j$ dla $i < j$. To, czy zbiór jest otwarty z prawej czy z lewej strony nie jest najważniejsze. Przede wszystkim należy zwrócić uwagę na samo podejście. W przypadku naszej funkcji f zbiory A_i nie mogą mieć elementów wspólnych parami i to jest tutaj bardzo ważne.

Klasami równoważności dla funkcji f będą zbiory z ciągu $\{A_0, A_1, \dots, A_n\}$. Z pewnością ***wartościami stojącymi na granicy klas*** dla zbiorów A_i będą liczby a_i, b_i, dla $i \in \{0, \dots, n\}$, ale nie tylko te. Pojawia się natomiast problem, jeśli chcemy podać ***wartości brzegowe prawostronne*** dla zbiorów $\{A_0, A_1, \dots, A_n\}$. W celu rozwiązania tego problemu wykorzystamy własności liczb takie, że dla każdej liczby $s \in R$ zachodzi

$$s - \frac{1}{n} < s \text{ dla } n \in N, \ (1)$$
$$s + \frac{1}{n} > s \text{ dla } n \in N, (2)$$
$$\lim_{n \to \infty} \left(s - \frac{1}{n}\right) = s, \ (3)$$
$$\lim_{n \to \infty} \left(s + \frac{1}{n}\right) = s. \ (4)$$

Mówiąc o analizie wartości brzegowych mamy na myśli wartości znajdujące się blisko s, Wobec tego można wykorzystać własności (1) - (4) ale dla dużych liczb n, np. $n > 10000$. Te własności należy rozumieć tak, że przy dość dużej liczbie n dzięki własnościom (3) i (4) podejdziemy bardzo blisko pod liczbę s. Takie postępowanie ma sens, ponieważ istnieją systemy informatyczne, dla których dokładność obliczeń jest jednym z podstawowych kryteriów akceptacji testów.

Problem wyznaczenia wartości brzegowych rozwiązać poprzez zaprojektowanie i zastosowanie algorytmu ewolucyjnego, który sam losuje liczby podchodzące bardzo blisko wartości granicznej, lub napisać takie wartości samemu. Podejście tak naprawdę zależy od tego, kto pisze testy.

Rozwiązanie postawionego problemu można jasno i czytelnie przedstawić na przykładzie magazynu. Istnieją hurtownie oraz magazyny, w których dla każdego pracownika, który kompletuje zamówienia, obliczany jest współczynnik wydajności (za pomocą urządzenia elektronicznego):

W pewnym magazynie są przyznawane premie dla pracowników w wysokości 300 zł, 400 zł, 500 zł, 600 zł. Zależy to od wysokości współczynnika W:

1) jeśli $W < 3$, nie jest przyznawana premia,

2) jeśli $W \in [3; 4)$, to premia wynosi 300 zł,

3) jeśli $W \in [4;5)$, to premia wynosi 400 zł,

4) jeśli $W \in [5;6)$, to premia wynosi 500 zł,

5) jeśli $W \geq 6$ to premia wynosi 600 zł.

Należy przetestować funkcjonalność jak najdokładniej, która zwraca kwotę premii. Wobec tego wykorzystamy własności (1) – (4). Przyjmiemy $n = 10000$. Wobec tego ciąg naszych wartości brzegowych będzie taki:

$\{2{,}9999; 3; 3{,}0001; 3{,}9999; 4; 4{,}0001; 4{,}9999; 5; 5{,}0001; 5{,}9999; 6; 6{,}0001\}$. Oczywiście nic nie stoi na przeszkodzie, żeby dołożyć liczby do naszego zbioru np. $3{,}0011$, $3{,}0003, 3{,}0031$ itd. Zależy to od tego, jak bardzo chcemy potwierdzić jakość testowanej funkcjonalności.

4. Propozycja wskazania wartości brzegowych funkcji złożonej

4.1 Wprowadzenie do problemu

Załóżmy teraz, że mamy funkcję $g : R^m \to D$ oraz funkcję $h = f(g)$. Funkcja f ma przepis taki, jak w rozdziale 3, czyli taką jaką rozważamy. Jak teraz można wyznaczyć wartości ze zbioru R^m, dla których zostaną przeprowadzone testy wartości brzegowych? Problem wydaje się niebanalny, jednak istnieje pewne formalne postępowanie, za pomocą którego można ten problem rozwikłać. Generowanie lub napisanie wartości brzegowych kończy się tym, że zwrócony zostaje skończony ciąg elementów $\{s_1, s_2, s_3, \dots, s_N\}$.

Formalnie wyznaczenie przeciwobrazu ustalonej wartości brzegowej s_i (czyli elementy wejściowe dla funkcji g) można zapisać tak:

$$g^{-1}(s_i) = \{(x_1, x_2, \dots x_2) \in R^m;\ g\big((x_1, x_2, \dots x_2)\big) = s_i; 1 \leq i \leq p\}$$

Mówiąc mniej formalnie, przeprowadzenie analizy wartości brzegowych w testach dla funkcji h sprowadza się do wyznaczenia przeciwobrazów funkcji g ale dla tych wartości, które są wartościami brzegowymi dla funkcji f. Nie jest to jednak takie proste i wymaga dostosowania rozwiązania do konkretnego przykładu.

Załóżmy, że dane są funkcje f, g opisane wzorami:

$$f(t) = \begin{cases} 0, & t \in [2;4) = A \\ 1, & t \in [4;10) = B \end{cases},$$

$$g(x, y, z) = x + y - z,$$

gdzie $(x, y, z) \in [1;4] \times [4;6] \times (0;3]$. Dana jest również funkcja $h = f[g(x, y, z)]$. Do analizy wartości brzegowych funkcji f został wskazany następujący ciąg liczb:

$$Z = \{2; 2{,}001; 1{,}999; 1{,}9599; 2{,}0501; 1{,}9995; 1{,}9876; 2{,}0002; 4; 3{,}999; 4{,}001; 4{,}1200; 4{,}0012;$$
$$4{,}0101; 4{,}0002; 10; 10{,}001; 9{,}999; 9{,}9988; 9{,}9979; 10{,}011; 10{,}0009; 10{,}0021\}.$$

Skoro już są wygenerowane wartości brzegowe dla funkcji f, to można zająć się wyznaczeniem wartości brzegowych funkcji h. Weźmy dla przykładu wartość 2 ze zbioru Z. Wyznaczenie elementów brzegowych względem wartości brzegowej 2 dla funkcji f dla funkcji h, sprowadza się do rozwiązania równania:

$$g(x, y, z) = 2,$$

czyli

$$x + y - z = 2.$$

Nasze równanie jest równoważne zapisowi:

$$z = x + y - 2.$$

Teraz możemy ustalić x, y oraz obliczyć z. Niech np. $x = 1, y = 4$. Wtedy $z = 3$. Wartość z należy do dziedziny funkcji g, więc trójka $(1{,}4{,}3)$ jest wartością brzegową funkcji h. Trzeba jednak pamiętać, że takich trójek może być więcej dla funkcji. Obliczyć to można w taki sposób, że ustalamy np. x oraz z, a następnie obliczamy brakujący y. Przy takim podejściu analiza wartości brzegowych funkcji h będzie trwała dłużej, ale za to będzie bardziej pewna. Analogicznie postępujemy z pozostałymi wartościami. Można (ale nie w każdym przypadku) dla pewnej wartości brzegowej funkcji f wyznaczyć więcej niż jeden argument funkcji g, taki, który spełni warunek wartości brzegowej również dla funkcji h. Weźmy kolejny raz równanie

$$g(x, y, z) = 1{,}999,$$

Niech $x = 1, y = 4$. Wtedy $z = 3{,}001$. Okazało się, że trójka $(1{,}1; 4{,}1; 3{,}001)$ jest spoza dziedziny funkcji g. Ale taki przypadek też jest potrzebny, żeby wiedzieć, jak funkcja h zachowa się w przypadku otrzymania wartości spoza dziedziny. Zwiększając x oraz y otrzymamy w wyniku rozwiązania równanie jeszcze większą wartość z, więc wobec tego nie będzie przypadku, że funkcja h dostanie argument $1{,}999$. Analogiczna sytuacja zajdzie dla wartości większych od 10, jeśli je podstawimy do równania. Wynika to z ograniczeń dziedziny funkcji g. Weźmy teraz równanie

$$g(x, y, z) = 4,$$

Ustalmy $x = 2{,}2, y = 4{,}01$. Ponieważ $z = x + y - 4$, $z = 2{,}21$. Można wziąć $x = 2{,}5566, y = 4{,}12$, wobec czego $z = 2{,}6766$. Tym razem dwie trójki (2,2;4,01;2,21) oraz $(2{,}5566; 4{,}12; 2{,}6766)$ będą poprawne w celu sprawdzenia wartości brzegowej równej 4.

4.2. Przykład zastosowania wyżej przedstawionej koncepcji

W celu zrozumienia wyżej przedstawionych rozważań wykorzystamy przykład poprzedni z magazynami:

Wiemy już, jak działa funkcja licząca premie. Teraz pójdziemy o krok dalej. Współczynnik W zależy od następujących czynników:

a) s - trudność trasy dla magazyniera,

b) x - ilość towarów,

c) t - czas składania palety łącznie z oznakowaniem jej kodem kreskowym.

Współczynnik ten wyraża się wzorem:

$$W = h(s, x, t) = \frac{sx}{t}.$$

Problemy jest oczywiście analogiczny do tego, który przedstawiony jest w rozdziale wcześniejszym, z tą różnicą, że mamy mnożenie i dzielenie zamiast dodawania i odejmowania. Chcemy teraz sprawdzić, czy system właściwie liczy premię na granicy przedziałów, ale wygenerowanych poprzez liczby s, x, t. Wobec tego należy dobrać takie s, x, t, żeby za ich pomocą wygenerować wartości np. takie jak poprzednio:

$\{2{,}9999; 3; 3{,}0001; 3{,}9999; 4; 4{,}0001; 4{,}9999; 5; 5{,}0001; 5{,}9999; 6; 6{,}0001\}$.

Rozważmy wartość współczynnika W równą 2,9999. Ustalmy, że $x = 89, s = 1{,}3445$. Chcemy obliczyć t, tak aby otrzymać wartość współczynnika 2,9999. Po przekształceniu wzoru

$$t = \frac{sx}{W} = \frac{1{,}3445 * 89}{2{,}9999} = 39{,}8882.$$

Sama funkcja obliczająca współczynnik W jest funkcją wewnętrzną, więc z punktu widzenia postawionego problemu testowania wartości brzegowych w przypadku złożenia funkcji nie interesuje nas jak ona działa, tylko jaki wynik zwróci funkcja licząca premię za pomocą współczynnika W. Z obliczeń wynika, że trójka $(1{,}3445; 89; 2{,}9999)$ wygeneruje taką wartość brzegową dla funkcji, która zwróci komunikat: brak premii. Możemy wyznaczyć więcej trójek, dla których testowany program ma zwrócić komunikat o braku premii. Wystarczy przykładowo założyć, że $x \in [70,121]$, ustalić że $s = 1{,}3445, W = 2{,}9999$, a wartość t obliczamy ze względy na x. Wobec tego można to przedstawić w postaci tabeli:

x	t
70	31,37271
73	32,71726
76	34,0618
79	35,40635
82	36,75089
85	38,09544
88	39,43998
91	40,78453
94	42,12907
97	43,47362
100	44,81816
103	46,16271
106	47,50725
109	48,8518
112	50,19634
115	51,54088
118	52,88543
121	54,22997

Teraz już wstawiając do testowanej funkcji wartości $s = 1{,}3445$, natomiast x oraz t parami z tabeli, funkcja za każdym razem powinna zwrócić wartość : brak premii. Analogicznie postąpić można z pozostałymi wartościami brzegowymi dla funkcji liczącej premię. Ilość trójek dla każdej wartości brzegowej jest zależna od tego, jak dużo testów chcemy przeprowadzić.

5. Dodatek: Propozycja wskazania wartości brzegowych kombinacji funkcji

Załóżmy teraz, że mamy dany następujący ciąg funkcji $\{f_1, f_2, \dots f_k\}$ oraz mamy daną funkcję $F: R^m \to R$, gdzie $F(x_1, x_2, \dots x_2) = f(f_1, f_2, \dots f_k)$. Wyobraźmy sobie, że każda funkcja f_i posiada tylko dwie klasy równoważności. Liczba kombinacji klas dla funkcji F wynosi 2^m, a liczba wartości brzegowych będzie większa niż 2^m. Testowanie wszystkich kombinacji klas jest dość pracochłonne. Ale można temu zaradzić stosując redukcję par testowych metodą ***pairwise testing.***

Testowanie par (*ang. Pairwise testing*) to czarno-skrzynkowa technika projektowania przypadków testowych w której przypadki testowe są projektowane tak, aby wykonać wszystkie możliwe dyskretne kombinacje każdej pary parametrów wejściowych. Jest wiele metod i algorytmów, które rozwiązują zagadnienie testowania parami. Istnieje sporo artykułów dotyczących tego zagadnienia. Przykładowe takie prace są wymienione w bibliografii, dlatego pominiemy tutaj opis tej metody, tylko zostaną przedstawione konkrety.

Korzystając z poprzednich założeń, niech dane będą następujące funkcje

$$g(x) = \begin{cases} 0, & x \in A_0 = [0;1) \\ 1, & x \in A_1 = [1;2) \end{cases},$$

$$h(y) = \begin{cases} 2, & y \in A_2 = [1;2) \\ 3, & y \in A_3 = [2;3) \end{cases},$$

$$j(z) = \begin{cases} 4, & z \in A_4 = [0;1) \\ 5, & z \in A_5 = [1;2) \end{cases},$$

oraz niech przykładowo

$$F(g,h,j) = \frac{g(x) + h(x)}{j(x)}.$$

Tabela poniżej przedstawia kombinacje klas dla funkcji F.

ID	$g(x)$	$h(y)$	$j(z)$
1	A_0	A_2	A_4
2	A_0	A_2	A_5
3	A_0	A_3	A_4
4	A_0	A_3	A_5
5	A_1	A_2	A_4
6	A_1	A_2	A_5
7	A_1	A_3	A_4
8	A_1	A_3	A_5

Po zastosowaniu *pair wise testing* tabela zostanie zmniejszona do czterech przypadków:

ID	$g(x)$	$h(y)$	$j(z)$
1	A_0	A_2	A_5
2	A_0	A_3	A_4
3	A_1	A_2	A_4
4	A_1	A_3	A_5

Dla ID 1 wartości brzegowe (jako trójki) mogą być np. takie:

1) $(-0;0001;\ 0{,}9999;0{,}9999)$;

2) $(0;1;1)$;

3) $(0{,}0001;\ 1{,}0001;1{,}0001)$;

4) $(0{,}9999;\ 1{,}9999;1{,}9999)$;

Analogicznie postępować można przy pozostałych przypadkach. Można również takich trójek do jednego przypadku napisać więcej. Będą one dodatkowo bliższe wartościom granicznym. Przede wszystkim po zastosowaniu redukcji jest o wiele mniej przypadków do zapisania.

6. Podsumowanie

Celem napisania tego artykułu było zaproponowanie pewnego sposobu wyznaczania wartości brzegowych w przypadku funkcji, które mają dziedzinę niedyskretną oraz spójną. Cechą takich funkcji jest to, że mając wartość znajdującą się blisko brzegu zawsze można znaleźć wartość stojącą obok niej, która będzie również wartością brzegową. Należy dlatego szukać pewnych metod, które wskażą takie wartości. Nie jest oczywiście koniecznością stosowanie algorytmu do generowania wartości brzegowych. Można to zrobić ręcznie, czyli po prostu obliczyć i jest to kolejny sposób postępowania. Wybór zależy od tego, kto pisze testy.

Autor chciał również zwrócić uwagę na problem analizy wartości brzegowych w przypadku złożenia lub kombinacji funkcji i zaproponować pewny sposób postępowania. Przedstawione rozwiązanie jest bardziej matematyczne niż typowo testerskie, ale ciekawe i w niektórych przypadkach może być pomocne.

7. Bibliografia:

1. **Tai K., Lei Y.,** *A Test Generation Strategy For Pair Wise Testing*, **IEEE Transactions on Software Enginiering, January 2002,**

2. *Software Testing: A Craftsman's Approach*, Fourth Edition , Paul C. Jorgensen, Hardcover – **October 18, 2013.**

Autor

Marek Żukowicz jest absolwentem matematyki na Uniwersytecie Rzeszowskim. Obecnie pracuje jako tester. Jego zainteresowania skupiają się wokół testowania, matematyki, zastosowania algorytmów ewolucyjnych oraz zastosowania matematyki w procesie testowania. Interesuje się również muzyką, grą na akordeonach oraz na perkusji.