Brzegi w testowaniu oprogramowania

Marek Żukowicz

12 kwietnia 2017

Streszczenie

Analiza wartości brzegowych oraz samo pojęcie brzegu to elementy testowania, z którymi na ogół spotyka się każdy tester podczas swojej pracy zawodowej. Pojęcie brzegu kojarzy się każdej osobie najczęściej z jednowymiarową osią oraz pewnymi wyszczególnionymi punktami. W pewnych przypadkach jest to poprawne skojarzenie, ale nie jest wystarczające dla występujących przypadków brzegów w testowaniu oprogramowania. Celem artykułu jest zwrócenie uwagi czytelnika na brzeg z perspektywy analizy matematycznej oraz relacji pomiędzy zbiorami. Artykuł prezentuje również koncpcję testowania brzegów, których nie da się opisać punktowo.

1 Matematyczne znaczenie brzegu testowanego zbioru

W tym rozdziale zostanie przedstawiony materiał, za pomocą którego można udowodnić, że nie da się wyznaczyć dwóch różnych brzegów zbioru danych testowych w procesie projektowania testów dziedziny, czy klas równoważności (w przypadku zbiorów dających się opisać teorią mnogości) w zależności od punktu widzenia analityka testów. Samo pojęcie analizy wartości brzegowych jest bardzo dobrze znane w środowisku testerskim, dlatego nie będziemy o tym wspominać. Mimo wszystko warto podkreślić, że matematyka pozwala na dowodzenie ciekawych własności pewnych zjawisk w naukach ścisłych. Zaczniemy od prezentacji matematycznych pojęć niezbędnych do wykazania jednoznaczności brzegu:

Definicja 1 *[1] Przestrzenią topologiczną nazywamy parę* (X, τ) *złożoną ze zbioru* X *i jego rodziny podzbiorów* τ *spełniającej warunki:*

1. $\emptyset \in \tau$ *oraz* $X \in \tau$,
2. *Jeśli* $A \in \tau$ *oraz* $B \in \tau$, *to* $A \cap B \in \tau$,
3. *Jeśli* $\mathcal{A} \subset \tau$, *to* $\bigcup \mathcal{A} \in \tau$.

Zbiory należące do rodziny τ będziemy nazywali *zbiorami otwartymi.* Warunki 1-3 możemy sformułować następująco:

1. Zbiór pusty i cała przestrzeń są zbiorami otwartymi,
2. Iloczyn dwóch zbiorów otwartych jest zbiorem otwartym,
3. Suma dowolnie wielu zbiorów otwartych jest zbiorem otwartym.

W celu zrozumienia przez czytelnika pojęcia brzegu (z punktu widzenia matematyki), podane są definicje:

Definicja 2 *[1] Zbiorem domkniętym nazywamy zbiór, którego dopełnienie jest zbiorem otwartym.*

Inaczej mówiąc A jest zbiorem domkniętym wtedy i tylko wtedy, gdy $X \setminus A$ jest zbiorem otwartym.

Definicja 3 *[1] Niech (X, τ) będzie przestrzenią topologiczną. Domknięciem zbioru $A \subset X$ nazwywamy najmniejszy w sensie inkluzji zbiór domknięty $\overline{A}$ zawierający A.*

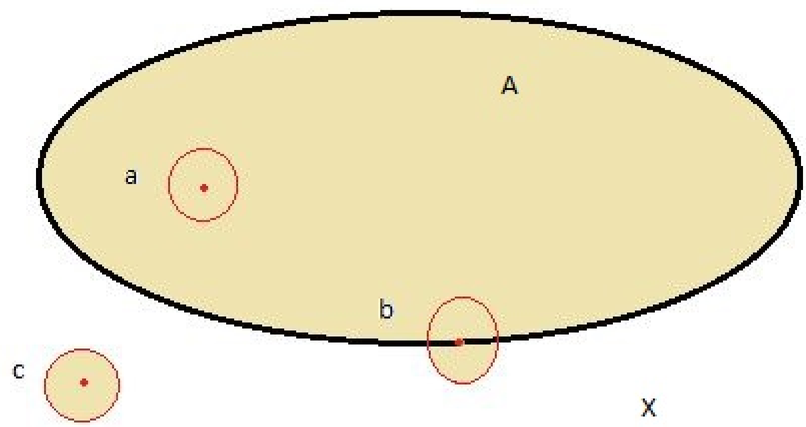
Zbiór domknięty oraz domknięcie zbioru nie są tymi samymi pojęciami. Sam brzeg, to pojęcie topologiczno-geometryczne formalizujące intuicję punktów „granicznych" danego zbioru, czy figury. W matematyce (jak również w testowaniu oprogramowania) zachowanie funkcji na brzegu dziedziny może się znacząco różnić od zachowania w jego wnętrzu. Formalne definicje punktu brzegowego i brzegu są następujące:

Definicja 4 *[3] Niech (X, τ) będzie przestrzenią topologiczną. Punktem brzegowym nazywamy taki punkt a, dla którego dowolny zbiór otwarty zawierający a ma punkt wspólny ze zbiorem A i z dopełnieniem zbioru A (czyli $X \setminus A$).*

Niech clA, $intA$, bdA oznaczają odpowiednio domknięcie, wnętrze oraz brzeg zbioru A. Wówczas mają miejsce następujące wyrażenia:

1. $bdA = bd(X \setminus A)$,
2. $bdA \supseteq bd(bdA)$.
3. $bdA = clA \setminus int\ A$.

Graficznie punkt brzegowy oraz brzeg zbioru można zaprezentować tak, jak na obrazku poniżej:



Punkt a jest punktem, który należy do wnętrza zbioru ponieważ można go otoczyć innym zbiorem zawartym w A. Punkt b jest punktem brzegowym, ponieważ ma niepustą część wspólną ze zbiorem A. Punkt c natomiast nie należy do zbioru A.

Z własności 1 oraz 2 można wyciagnąć następujące wnioski dla analizy testów:

Punkty brzegowe dla dziedziny testowanej funcjonalności są równe wszystkim punktom brzegowym obszaru spoza dziedziny. Wobec tego nie może mieć miejsca taka sytuacja, że dwóch różnych analityków testów wyznaczy dwa różne zbiory punktów brzegowych dla testów dziedziny. Zbiory danych, z którymi spotykamy się podczas testowania, są najczęściej zbiorami liczbowymi lub wektorami danych. W takim przypadku w prosty sposób można wprowadzić strukturę przestrzeni topologicznej wykorzystując metrykę np. euklidesową (własności 1, 2, 3 zostaną zachowane w takiej przestrzeni). W przypadku przestrzeni topologicznej generowanej przez metrykę własność 2 na ogół będzie miała znak równości. Tą własność można zinterpretować tak, że mając poprawnie wyznaczony brzeg zbioru opisanego punktowo nie ma sensu doszukiwać się innego brzegu, ponieważ operacja wyznaczania brzegu od zbioru który już jest brzegiem, daje ten sam brzeg. Z kolei punkt 3 dowodzi, że nie istnieje niepusty zbiór rozważany w teorii mnogości, w którym wnętrze może mieć chociaż jeden punkt wspólny z brzegiem tego zbioru. Wobec tego, podczas analizy wartości brzegowych nie ma możliwości zakwalifikowania danej testowej jednocześnie do punktu brzegu i do wnętrza dziedziny, czy klasy równoważności.

2 Brzegi bezpunktowe

Ten rozdział poświęcony został analizie wartości brzegowych (dokładnie testowaniu skrajnych sytuacji) w przypadkach, gdy wspomniane brzegi nie dają się jawnie opisać za pomocą zbiorów znanych z teorii mnogości.

Wyobraźmy sobie taką sytuację, że należy zaprojektować testy integracyjne aplikacji, która jest pewnym dodatkiem lub nakładką usprawniającą procesy biznesowe, które nie są udostępnione w dużym systemie klasy ERP lub należy wykonać dużą ilość kroków aby te czynności zrealizować. Na pierwszy rzut oka nie widać w takiej sytuacji wartości brzegowych. Ale odchodząc od tradycyjnego pojęcia brzegu, możemy pomyśleć o testowaniu skrajnych przypadków (pewnego marginesu zdefiniowanego dla danych testowych lub dla testów wdrożenia aplikacji).

Załóżmy, że mamy do zaprojektowania testy wdrożenia integracji małej aplikacji, która jest dodatkiem do pewnego systemu np. ERP (opartego o bazę danych). Wymagane są dodatkowe tabele dołożone do bazy wspomnianego ERP-a. Takie testy na ogół wymagają sprawdzenia poprawności działania triggerów, mechanizmów synchronizacji wpisów pomiędzy tabelami w bazie danych. Skrajne sytuacje testów integracji obu aplikacji są następujące:

1. Tabele ERP-a oraz tabele małej aplikacji są puste,
2. Tabele ERP-a są wypełnione „dużą" ilością danych, natomiast tabele małej aplikacji są puste,
3. Tabele ERP-a są puste, natomiast tabele małej aplikacji są wypełnione „dużą" ilością danych,
4. Tabele obu aplikacji są wypełnione „dużą" ilością danych.

Wyżej opisane sytuacje nie wyczerpują wszystkich możliwości. Dotyczą one tylko przypadków skrajnych (brzegów bezpunktowych). Z pewnością nie zaszkodzi wykonanie dodatkowych testów, które nie są przypadkami skrajnymi, a bardziej rzeczywistymi. Słowa „dużą" ilością danych nie dają się sprecyzować, ponieważ testowanie jest zależne od przeznaczenia i kontekstu. Wspomniana „duża" ilość może mieć różne znaczenia dla różnych aplikacji.

Sytuacje skrajne (istnienie brzegów bezpunktowych) w testowaniu wdrożenia, wydajności czy funkcjonalności występuje podczas projektowania testów:

- Testowania odczytów dużej ilości przefiltrowanych danych prezentowanych na formularzu w aplikacji. W takim przypadku zapytania do bazy danych muszą być optymalne, aby aplikacja nie utrudniała odczytu. W takiej sytuacji wartość skrajna jest pewną z góry ustaloną liczbą wypełnionych rekordów w bazie danych np. 1 000 000,
- Testy generowania raportów finansowych dla dużej liczby pracowników zarejestrowanych w bazie danych np. 2 000 pracowników,
- Testy przeciążeniowe i wydajnościowe aplikacji webowych,

- Generowanie różnego rodzaju dokumentów magazynowych bądź finansowych z olbrzymią ilością pozycji (brzegiem w tym przypadku są pozycje na dokumencie, które muszą być poprawnie przeliczone i dobrze prezentowane wizualnie),
- Testowanie prezentacji oraz obliczania wartości danych zależnych od siebie w sposób hierarchiczny, np. kontrolka TreeList, która umożliwia zagnieżdżenie danych do dowolnego poziomu oraz umożliwia tworzenie dowolnej ilości gałęzi [2]. Brzegami są tutaj poziom zagnieżdżenia danych oraz ilość gałęzi w drzewie,
- Testowanie aplikacji mobilnych na najmniejszych oraz największych dostępnych rozdzielczościach ekranu,
- Testowanie drukowania w zależności od rozdzielczości ekranu.

Wyżej wymienione sytuacje z pewnością nie wyczerpują wszystkich przypadków brzegów bezpunktowych. Testowanie aplikacji może wykazać istnienie wielu brzegów bezpunktowych. Należy jednak pamiętać, że testowanie przypadków brzegowych niekoniecznie musi zaspokoić wszystkie wymagane warunki testowe.

3 Rozmyte brzegi w testowaniu

W poprzednim rozdziale wspomniane zostały testy wydajnościowe i przeciążeniowe. Takie testy przeprowadza się żeby sprawdzić jak zachowa się aplikacja w przypadku „dużej” liczby jednocześnie działających użytkowników. Kolejny raz pojawia się słowo „dużej”. Można powiedzieć, że duża liczba osób to 100000. Wobec tego ktoś może zapytać dalej: a nie warto robić testów dla 11000, czy 10100, czy 9500 użytkowników, bo przecież one nie zaszkodzą aplikacji? W przypadku portalu internetowego nie zawsze można przewidzieć ilość zainteresowanych użytkowników. Pojawia się pytanie jak wobc tego zaprojektować testy wydajnościowe oraz trafić w optymalną granicę liczby zainteresowanych użytkowników? W takim celu można wykorzystać:

- doświadczenie osób, które spotkały się z podobnym problemem,
- zrobić analizę potrzeb rynku,
- zdefiniować odpowiednie miary dla rezultatów testów.

W matematyce istnieje takie pojęcie jak zbiór rozmyty, dzięki któremu można określić, jak bardzo pewien element zbioru jest zbliżony do ustalonej z góry wartości progowej. Jego definicja jest następująca:

Definicja 5 *[4] Zbiorem rozmytym A w przestrzeni X jest zbiór uporządkowanych par*

$$A = \{(x, \mu(x)) | x \in X\},$$

gdzie $\mu(x) : X \to [0, 1]$.

Funkcję μ interpretuje się jako stopień bliskości dla pewnego z góry określonego elementu zbioru X.
Przykładem zbioru rozmytego może być zbiór $A = \{(x, \mu(x)) | x \in \mathbb{R}\}$, gdzie

$$\mu(x) = \begin{cases} 1 - x, a \in [0, 1] \\ 0, x \in \mathbb{R} \setminus [0, 1] \end{cases}.$$

Dla liczby wirtualnych użytkowników podczas implementacji testów obciążeniowych można również utworzyć funkcję, która określi nam, na ile przydatna do testów jest liczba wirtualnych użytkowników, którą chcemy zastosować. Przykładowa taka funkcja dla 10000 użytkowników może mieć postać:

$$\mu(x) = \begin{cases} \frac{1}{10000} * (10000 - x), x \in [0, 10000] \\ \frac{-1}{10000} * (x - 10000), x \in (10000, 20000] \\ 0, x \notin [0, 20000] \end{cases}.$$

Mając taką funkcję możemy przyjąć, że jeśli $\mu(x) > 0.8$, to liczba użytkowników jest optymalna dla testów i takie testy będą słuszne. Proces wyznaczanie takiej funkcji wymaga przypomnienia własności elementarnych funkcji znanych ze szkoły średniej. Załóżmy, że chcemy wyznaczyć rozmytą funkcję przynależności $\mu(x)$ dla liczby 100. Chcemy, aby to była parabola. Z opisanych założeń wynika, że funkcja musi mieć gałęzie skierowane w dół, a współrzędne wierzchołka to $(100, 1)$, współczynnik kierunkowy a będzie miał wartość ujemną. Zbiór wartości oczywiście musi mieścić się w przedziale $[0, 1]$, więc z założenia przyjmiemy wartość funkcji równą 0 dla argumentów, dla których wartości paraboli będą ujemne. Dziedziną funkcji muszą być liczby naturalne (bo liczba użytkowników wirtualnych nie może być ani ujemna, ani ułamkowa). Postać kanoniczna funkcji kwadratowej prezentuje się następująco:

$$f(x) = a(x - p)^2 + q,$$

gdzie p, q są współrzędnymi wierzchołka funkcji. Wobec tego

$$\mu(x) = \begin{cases} a(x - 100)^2 + 1, x \in A \\ 0, x \notin \mathbb{N} \setminus A \end{cases}.$$

Współczynnik a nie jest możlilwy do wyznaczenia w prosty sposób, więc można go po prostu oszacować. Przyjmijmy $a = -\frac{1}{50}$. Po podstawieniu a do wzoru paraboli otrzymujemy równanie kwadratowe

$$\mu(x) = \frac{-x^2}{50} + 4x - 199.$$

Ponieważ $\Delta = 0.08$, istnieją dwa miejsca zerowe funkcji μ. Ich przybliżone wartości 93 oraz 107. Wobec tego zbiór $A = [93, 94, 95, ..., 107]$, a nasza funkcja określająca bliskość względem liczby 100 użytkowników wirtualnych ma postać:

$$\mu(x) = \begin{cases} -\frac{1}{50}(x-100)^2 + 1, x \in [93, 94, 95, ..., 107] \cap \mathbb{N} \\ 0, x \notin \mathbb{N} \setminus [93, 94, 95, ..., 107] \end{cases}.$$

4 Podsumowanie

Artykuł został napisany w dwóch celach. Pierwszym z nich było udowodnienie za pomocą matematycznych tożsamości pewnych niepodważalnych własności brzegów zbiorów (które dają się opisać liczbowo) i odniesienie się do tych własności w kontekście testowania. Drugim celem było przedstawienie koncepcji brzegów nieliczbowych oraz zaproponowanie pewnego podejścia, które można wykorzystać podczas projektowania testów wdrożeniowych, niefunkcjonalnych oraz funkcjonalnych. Z brzegami nieliczbowymi tak naprawdę testerzy spotykają się często w pracy.

Literatura

[1] Engelking R., *Topologia Ogólna*, PWN Warszawa 1976.

[2] https://msdn.microsoft.com/en-us/library/dd285329.aspx

[3] https://pl.wikipedia.org/wiki/Brzeg_(matematyka)

[4] https://pl.wikipedia.org/wiki/Zbi%C3%B3r_rozmyty