Zastosowanie logiki matematycznej w procesie weryfikacji wymagań oprogramowania

Testerzy oprogramowania lub osoby odpowiedzialne za zapewnienie jakości oprogramowania oprócz wykonywania testów mogą zostać zaangażowani do sprawdzania poprawności zdefiniowanych wymagań. W specyfikacji oprogramowania a konkretnie w opisie wymaganych funkcjonalności można wyróżnić różne rodzaje zdań, w tym zdania logiczne, które są często niezbędne do opisania działania aplikacji. Może zaistnieć taka sytuacja, że pewne zdania logiczne umieszczone w specyfikacji oprogramowania są ze sobą sprzeczne.

Ale stosując logikę matematyczną można wyłapać pewne nieścisłości, a przez to znacznie podnieść jakość oprogramowania, zanim zostanie ono wytworzone.

Pisząc artykuł autor zakłada, że czytelnik zna pojęcia funkcji zdaniowych stosowanych w logice matematycznej jak *zaprzeczenie zdania, koniunkcja zdań, alternatywa zdań, implikacja, równoważność zdań* oraz tabele prawdy dla tych funkcji.

1. Wstęp

Aby stosować logikę matematyczną, należy znać oraz rozumieć jej podstawowe pojęcia oraz znać podstawowe funkcje zdaniowe. W niniejszym rozdziale zostaną przypomniane wiadomości, które mają za zadanie uporządkowanie wiedzy czytelnika.

1.1 Formy zdaniowe

Definicja 1. [1] Dla dowolnej przestrzeni (*tutaj zbiór zawierający pewne dane*) $X \neq \emptyset$, wyrażenie $w(x)$, w którym występuje zmienna *x* i które staje się zdaniem prawdziwym lub fałszywym nazywamy funkcją zdaniową (formą zdaniową) jednej zmiennej, której zakresem zmienności jest przestrzeń X.

Definicja 2. [1] Funkcją zdaniową określoną w pewnym zbiorze nazywamy każde zdanie zawierające zmienną, takie, że po wstawieniu w miejsce zmiennej dowolnego elementu z tego zbioru zdanie to staje się zdaniem logicznym.

Uwaga 1! W jednym zdaniu logicznym może istnieć więcej niż jedna zmienna.

Mówimy, że dla dowolnej funkcji zdaniowej $w(x)$ element dziedziny funkcji *spełnia funkcję zdaniową* wtedy i tylko wtedy, gdy po podstawieniu go do tej funkcji zdaniowej w miejsce zmiennej otrzymamy zdanie prawdziwe. Zbiór tych wszystkich wartości zmiennej $x \in X$, przy których funkcja

zdaniowa $w(x)$, staje się zdaniem prawdziwym, czyli zbiór tych x, które spełniają tę funkcję zdaniową, oznaczamy $\{ x \in X: w(x)\}$.

Przykład. 1

Dla formy zdaniowej $w(x)$: $x - 12 < 9$ dziedziną jest R – zbiór liczb rzeczywistych. Wstawiając za x liczbę 3, otrzymujemy $-4 < 9$, czyli zdanie logicznie prawdziwe. Wstawiając za x liczbę 30, otrzymujemy $18 < 9$ - zdanie logiczne fałszywe. Mówimy, że liczba 3 spełnia tę funkcję zdaniową, a liczba 30 jej nie spełnia.

Przykład 2.

Równanie $r(x)$: $2x + 7 = 0$ jest formą zdaniową jednej zmiennej x, której dziedziną jest zbiór liczb rzeczywistych, a elementem spełniającym liczba $-3,5$.

1.2 Reguły dowodzenia i twierdzenia

Przeprowadzanie rozumowania w dowodach matematycznych składa się na ogół z bardzo prostych kroków polegających na stwierdzeniu poprawności pewnych zdań, czy też funkcji zdaniowych. Te elementarne ogniwa rozumowań dedukcyjnych nazywają się **regułami dowodzenia** [2]. Dowody matematyczne przeprowadza się najczęściej w celu potwierdzenia prawdziwości twierdzenia, które definiuje się następująco:

Definicja 3. [3] Twierdzeniem jest każde zdanie prawdziwe w teorii nie jest będące aksjomatem (*aksjomat to przyjęty warunek, który jest zawsze prawdziwy*). Twierdzenia często przyjmują postać implikacji:

$$(p1 \wedge p2 \wedge \ldots \wedge pn) \Rightarrow q \ (1),$$

gdzie zdania $p1, p2, \ldots pn$ nazywamy założeniami twierdzenia, a zdanie q - tezą twierdzenia. Inną postacią twierdzeń jest postać równoważności: $p \Leftrightarrow q$. Takie twierdzenia są równoważne parze twierdzeń: $p \Rightarrow q$ i $q \Rightarrow p$. Tu zdania p i q przyjmują na zmianę rolę założenia i rolę tezy [2]. Reguła dowodzenia może być przedstawiona również w postaci:

$$\frac{p1\ ,\ p2\ ,\ldots,\ pn}{q}\ (2).$$

Po zastąpieniu przecinków koniunkcją, kreski ułamkowej implikacją powstaje wyrażenie (1). Jeśli można udowodnić, że utworzone zdanie jest **tautologią [3].** (zawsze zdaniem prawdziwym, niezależnie od wartości zmiennych), to reguła (2) jest regułą dowodzenia.

Przykład 3. (Reguła odrywania)

Reguła odrywania mówi, że jeżeli prawdziwe są zdania p oraz $p \Rightarrow q$, to prawdziwe jest zdanie q.

Przykład 4. (Reguła dowodu nie wprost)

Przypuśćmy, że chcemy udowodnić zdanie p. W tym celu zaprzeczamy zdaniu p i dowodzimy, że z zaprzeczenia zdania p wynika fałsz, w postaci zdania $r \wedge \sim r$. Jeżeli zdanie p ma postać implikacji $q \Rightarrow s$, to zaprzeczeniem tego zdania jest zdanie $q \wedge \sim s$. Ta własność wynika z tautologii $(q \Rightarrow s) \Leftrightarrow (\sim p \vee s)$. Czyli zakładamy prawdziwość założeń, fałszywość tezy i dowodzimy, że stąd wynika fałsz z założeniami.

Istnieje sporo reguł dowodzenia. Inne reguły dowodzenia oraz przykłady tautologii można znaleźć w pozycjach [3,4].

2. Kwantyfikatory, zasięg zmiennych

Często w definicjach, twierdzeniach lub wnioskach pojawiają się zwroty „Dla każdego x", „istnieje y", „istnieje x dla każdego y". Takie sformułowania są często związane z zasięgiem zmiennych występujących w funkcji zdaniowej. Takie wyrażenia są używane, aby określić pewne relacje pomiędzy zmiennymi. Pojawiają się one również w wymaganiach funkcjonalnych podczas projektowania wymagań. Wspomniane sformułowania są oznaczane znakami, które nazywane są w matematyce **kwantyfikatorami**:

Definicja 4. [4] Zwroty *dla każdego x, dla każdego y* nazywamy kwantyfikatorami ogólnymi (dużymi) i notujemy

$$\bigwedge_x, \bigwedge_y (3).$$

Zwroty *dla pewnego x , istnieje y* nazywamy kwantyfikatorami szczegółowymi (małymi) oraz notujemy

$$\bigvee_x, \bigvee_y (4).$$

Zwroty *dla każdego x, dla wszystkich x* są równoważne. Analogicznie zwroty *istnieje x, dla pewnego x* też są równoważne. Z pojęciem kwantyfikatorów związany jest ich zasięg oraz związanie zmiennej z kwantyfikatorem.

Definicja 5. [4] Zasięgiem kwantyfikatora (ogólnego lub szczegółowego) jest ta część wyrażenia, będąca również funkcją zdaniową, **ujęta w parę jednakowych nawiasów**, z których pierwszy występuje bezpośrednio po kwantyfikatorze. Zmienna x, występująca w danym wyrażeniu jest **zmienną wolną** tego wyrażenia wtedy i tylko wtedy, gdy nie występuje w zasięgu danego kwantyfikatora.

Definicja 6. [4] Zmienna x jest związana przez dany kwantyfikator, którego wskaźnikiem jest ta zmienna, wtedy i tylko wtedy, gdy występuje w jego zasięgu oraz w zasięgu tym jest zmienną wolną.

Przykład 5. W wyrażeniu

$$\bigwedge_x [(\underline{x} + 1 = 3) \Rightarrow \bigvee_x (2\underline{x} = 4)] \ (5)$$

litera x podkreślona jeden raz jest zmienną związaną przez duży kwantyfikator, gdyż występuje w jego zasięgu zamkniętym nawiasami kwadratowymi i jest w tym zasięgu zmienną wolną. Ale zmienna x podkreślona dwa razy nie jest związana z kwantyfikatorem dużym, chociaż występuje w jego zasięgu, ale w tym zasięgu nie jest zmienną wolną (jest związana przez mały kwantyfikator).

Budowanie wyrażeń matematycznych opartych na kwantyfikatorach jest dokładniej opisane w pozycji [4]. Nie każde wyrażenie matematyczne musi być poprawnie zbudowane, dlatego chcąc stosować kwantyfikatory oraz dowodzić prawdziwości postawionych tez należy znać reguły tworzenia poprawnych wyrażeń. W przypadku dowodzenia prawdziwości wyrażeń matematycznych ważne są dwie własności związane z zaprzeczeniem kwantyfikatorów:

$$\sim(\bigwedge p(x)) \ \Leftrightarrow \sim \bigvee \sim p(x) \ (6),$$

oraz

$$\sim(\bigvee p(x)) \Leftrightarrow \sim \bigwedge \sim p(x) \ (7).$$

3. Zastosowanie teorii w praktyce

Określanie prawdziwości zdań oraz stosowanie reguł dowodzenia logiki matematycznej może okazać się przydatne podczas weryfikowania wymagań dla oprogramowania. **Nie oznacza to jednak, że zawsze można w ten sposób postępować**. Niżej przedstawione przykłady są próbą przekonania osób związanych z zapewnieniem jakości oprogramowania, że warto podjąć próby stosowania logiki matematycznej, przynajmniej w niektórych przypadkach. Za pomocą kilku przykładów zostanie zaprezentowany sposób postępowania dla dowodzenia poprawności wymagań.

Przykład 5. Załóżmy, że należy zaprogramować metodę, która oblicza premię za miesiąc pracy dla pracownika pewnego magazynu. Wymagania zdefiniowane dla tej funkcjonalności są podane w tabeli:

a	Premia dla pracownika jest zależna od trzech wartości: czasu - t, ilości - x, długości - trasy - l, za pomocą których obliczane jest wyrażenie $p = \frac{xl}{t}$,
b	**Premia nie może być większa niż 1000 zł,**
c	Jeśli $p \in [0, 10)$, to nie należy się premia,
d	Funkcja licząca premię to dana jest wzorem $f(p) = \begin{cases} 0, & p \in [0, 10) \\ 10 * p, & p \geq 10 \end{cases}$.

Zdania a, b, d, można przyjąć jako założenia, natomiast zdanie c jako tezę pewnego twierdzenia. Logiczna postać wspomnianego twierdzenia ma postać:

$$(a \wedge c \wedge d) \Rightarrow b. (8)$$

Implikacja jest fałszywa tylko wtedy , gdy prawdziwy jest jej poprzednik $(a \wedge c \wedge d)$, a fałszywy następnik (b). Przyjęte postępowanie to metoda dowodzenia nie wprost. Chcąc sprawdzić, czy wymagania dla obliczania premii nie są ze sobą sprzeczne, zakładamy, że fałszywe jest zdanie b i będziemy próbowali dojść do sprzeczności z którymś ze zdań a, c, d. Zamiast formalnego dowodu można najpierw próbować znaleźć przykład, który pokazuje że wymagania są niepoprawne lub poprowadzić dowodzenie zgodnie z zasadami matematyki.
W tym przykładzie założymy, że premia może być większa niż 1000 zł.

Krok 1: $\sim b$ – Premia jest większa niż 1000 zł, np. 1500.

Krok 2: Wobec tego $f(p) = 1500 = 10 * p = 1500 \Leftrightarrow p = 150.$

Krok 3: Biorąc $x = 15, l = 10, t = 1$, a wtedy $p = \frac{xl}{t} = 150.$

Okazało się, że istnieją takie wartości zmiennych w zdaniu a, że twierdzenie (8) staje się nieprawdziwe. W takim przypadku wymagania muszą być uzupełnione o pewne ograniczenia dla wartości zmiennych lub zmienić przepis funkcji f.

Przykład 6. Załóżmy, że pewna platforma z telewizją cyfrową udostępnia zniżki dla długoletnich klientów, zgodnie z zasadami:

W1	Klienci, którzy korzystają z usług platformy co najmniej 2,5 roku dostają zniżkę 30% na pakiet z programami dla dzieci
W2	Klienci długoletni, którzy korzystają z usług platformy co najmniej 3,5 roku dostają zniżkę 30% na pakiet z programami sportowymi
W3	**Klient długoletni ma zostać poinformowany, że należy mu się zniżka na oba pakiety (sport i programy dla dzieci) jeśli nie korzysta ze zniżki na programy dla dzieci.**

Zadaniem programistów jest napisanie małej aplikacji, która będzie informowała o możliwości skorzystania ze zniżek. Wymagania 1,2 należy przyjąć jako założenia, natomiast wymagania 3 jako tezę. Należy sprawdzić, czy w wyżej zdefiniowanych wymaganiach nie występuje sprzeczność. W tym celu również można zastosować metody logiki matematycznej, zaczynając od definiowania pojedynczych zdań:

o: Klient korzysta z usług platformy co najmniej 2,5 roku,
p: Klient korzysta z usług platformy co najmniej 3,5 roku,
q: Klient ma prawo do zniżki na programy dla dzieci,
r: Klient ma prawo do zniżki na pakiet sportowy,
s: Klient nie korzysta ze zniżki na programy dla dzieci.

Twierdzenie, które należy wykazać ma postać:

$$[(o \Rightarrow q) \wedge (p \Rightarrow r)] \Rightarrow [(p \wedge s) \Rightarrow (q \wedge r)]\ (5).$$

Wobec tego:

$[(o \Rightarrow q) \wedge (p \Rightarrow r)] = 1, [(p \wedge s) \Rightarrow (q \wedge r)] = 0.$

Biorąc następnik implikacji

$$(p \wedge s) = 1, (q \wedge r) = 0.$$

1. Przyjmujemy, że zdanie $r = 0,\ q = 1$. Zdania p, s muszą mieć wartość 1. Podstawiając wartości. Podstawiając znane wartości do zdania (5), otrzymujemy:

$$[(o \Rightarrow 1) \wedge (1 \Rightarrow 0)] \Rightarrow [(1 \wedge 1) \Rightarrow (1 \wedge 0)].$$

Niezależnie od wartości zdania o wyrażenie (5) jest prawdziwe, ponieważ fałszywy jest poprzednik $[(o \Rightarrow 1) \wedge (1 \Rightarrow 0)]$, a wtedy cała implikacja jest prawdziwa.

2. Przyjmujemy, że zdanie $r = 1,\ q = 0$. Zdania p, s również muszą mieć wartość 1. podstawiając wartości. Podstawiając znane wartości do zdania (5), otrzymujemy:

$$[(o \Rightarrow 0) \wedge (1 \Rightarrow 1)] \Rightarrow [(1 \wedge 1) \Rightarrow (0 \wedge 1)]\ .$$

Jeśli $o = 0$, to zdanie jest prawdziwe. Jeśli $o = 1$, to niestety ale zdanie okazuje się fałszywe. Wobec tego istnieje sprzeczność w wymaganiach lub brakuje jakiegoś wymagania. Analizując sytuację, łatwo zauważyć, że prawdziwe jest wyrażenie $p \Rightarrow o$, ponieważ jeśli dana osoba jest klientem $3{,}5$ roku, to na pewno jest klientem $2{,}5$ roku.

Taka sytuacja powinna być sygnałem dla osoby piszącej wymagania do napisania warunku, że klient długoletni **również jest klientem nie krócej niż $2,5$ roku** i taki warunek powinien zostać zaimplementowany w aplikacji. Dzięki prawdziwość zdanie p automatycznie czyni prawdziwym zdanie o. Problem wydaje się „banalny”, ale często w praktyce zapomina się o rzeczach „banalnych”.

Przykład 7. Niech dane będą następujące wymagania dla portalu internetowego oferującego możliwość zamówienia soków. Wymagania dotyczące portalu prezentuje następująca tabela:

W1	Portal umożliwia wybór opakowań, w których butelki z sokiem pakowane są po 2, 4, 6 albo 12 sztuk.
W2	Portal pozwala zamówić zalogowanemu użytkownikowi pakiet zawierający nie więcej butelek, niż rozmiar pakietu.
W3	Można zamawiać tylko całe pakiety.

Łatwo zauważyć, że wyżej przedstawione wymagania W1 oraz W2 mogą być różnie zinterpretowane przez osobę, która programuje lub testuje.

Niech X będzie zbiorem użytkowników, a Y zbiorem pakietów z sokami. Analizując wyżej przedstawione wymagania, można ułożyć wyrażenia:

$q(y)$: pakiety $y \in Y$ są pakowane po 2, 4, 6 albo 12 sztuk,

$p(x,y)$: zalogowany każdy użytkownik $x \in X$ może dokonać zakupu tylko całych pakietów $y \in Y$,

$r(x,y)$: każdy użytkownik $x \in X$ może dokonać zakupu pakietu $y \in Y$, który zawiera nie więcej butelek niż rozmiar pakietu.

Językiem logiki matematycznej można zapisać zdanie $q(y)$ następująco:

$$\bigwedge_{y \in Y} [2|y \vee 4|y \vee 6|y \vee 12|y] \ (6).$$

Dla każdego zalogowanego użytkownika istnieje do wyboru pakiet, który można kupić, a więc prawdziwe jest wyrażenie:

$$\bigwedge_{x \in X} \bigvee_{y \in Y} [(2|y \vee 4|y \vee 6|y \vee 12|y) \Rightarrow p(x,y)] \ (7)$$

oraz, zgodnie z założeniami, prawdziwe jest również wyrażenie:

$$\bigwedge_{x \in X} \bigvee_{y \in Y} [(2|y \vee 4|y \vee 6|y \vee 12|y) \Rightarrow r(x,y)] \ (8) \, .$$

Niech symbol $\overline{\overline{y}}$ oznacza ilość elementów w pakiecie. Jeżeli wyrażenie (8) jest prawdziwe oraz $q(y)$ jest prawdą, to również $r(x,y)$ musi być prawdą, ponieważ implikacja:

$$(2|y \vee 4|y \vee 6|y \vee 12|y) \Rightarrow r(x,y)$$

byłaby fałszywa w przeciwnym przypadku. Jeśli $r(x,y)$ jest prawdą, to:

$$\bigwedge_{y \in Y} (\overline{\overline{y}} \leq 2 \vee \overline{\overline{y}} \leq 4 \vee \overline{\overline{y}} \leq 6 \vee \overline{\overline{y}} \leq 12),$$

a to z kolei oznacza, że:

$$\bigvee_{y \in Y} (\overline{\overline{y}} = 1 \vee \overline{\overline{y}} = 3 \vee \overline{\overline{y}} = 5 \vee \overline{\overline{y}} = 7),$$

co jest sprzeczne ze zdaniem $q(y)$, ponieważ nie istnieją pakiety, które mają 1, 3, 5 lub 7 butelek. Z wyżej przeprowadzonego rozumowania wynika, że nie powinny być napisane jednocześnie wymagania W1 oraz W2, ponieważ mogą być one różnie zinterpretowane.

4. Podsumowanie

Celem napisania artykułu była propozycja podjęcia prób zastosowania logiki matematycznej podczas weryfikacji wymagań dla oprogramowania, skierowana do osób związanych z kontrolą jakości oprogramowania, w tym także testerów oprogramowania.
Z artykułu wynika, że istnieją wymagania, które można opisać formami zdaniowymi, zamieniać te formy w zdania logiczne, przeprowadzać rozumowanie, które pozwala krok po kroku sprawdzić, czy wymagania są poprawnie zdefiniowane czy są zdefiniowane w taki sposób, że ich interpretacja może doprowadzić do sprzeczności pomiędzy funkcjonalnościami w oprogramowaniu. Takie podejście wymaga jednak znajomości zasad logiki matematycznej oraz znajomości zasad poprawnego budowania wyrażeń zdaniowych.

Z artykułu nie wynika również, że logika matematyczna zawsze zadziała podczas weryfikacji poprawności wymagań. Takie „śmiałe" stwierdzenie wymagałoby dowodu, aby uznać je za prawdziwe.

5. Bibliografia

[1] http://www.math.edu.pl/funkcja-zdaniowa, data dostępu 01.06.2016.

[2] http://www.math.edu.pl/twierdzenia-reguly-dowodzenia, data dostępu 01.06.2016.

[3] H. Rasiowa, *Wstęp do matematyki współczesnej*, PWN 2015.

[4] J. Słupecki, K. Hałkowska, K. Piróg-Rzepecka, *Logika i teoria mnogości,* PWN, W-wa 1978.

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 akordeonie oraz na perkusji.