Matematyka Dyskretna

Wykład 1
Wprowadzenie

Wojciech Kordecki
wojciech.kordecki@collegiumwitelona.pl

Collegium Witelona
Wydział Nauk Technicznych i Ekonomicznych
Zakład Informatyki

Semestr letni 2023/24

Dissertatio de arte combinatoria

Gottfried Wilhelm Leibniz, 1690.

Gottfried Wilhelm Leibniz 1646 – 1716

Literatura podstawowa

[R] R. Rębowski,
Matematyka dyskretna dla informatyków,
PWSZ w Legnicy, Legnica 2008.

[RP] R. Rębowski, J. Płaskonka,
Zbiór zadań z matematyki dyskretnej dla informatyków,
PWSZ w Legnicy, Legnica 2017.

[W] R. J. Wilson,
Wprowadzenie do teorii grafów, wydanie 2,
PWN, Warszawa 2012.

[LZ] H. Lewis, R. Zax,
Matematyka dyskretna, niezbędnik dla informatyków,
PWN, Warszawa 2021.

Literatura dodatkowa

[JPS] J. Jaworski, Z. Palka, J. Szymański,
Matematyka dyskretna dla informatyków, tom I,
Wyd. Naukowe UAM, Poznań 2007.

[KLH] W. Kordecki, A. Łyczkowska-Hanćkowiak.
Matematyka Dyskretna.
Helion, Gliwice, 2018.

[RW] K. A. Ross, C. R. B. Wright,
Matematyka dyskretna, wydanie 5,
PWN, Warszawa 2013.

[Z] M. Zakrzewski,
Matematyka dyskretna.
Markowe Wykłady z Matematyki, GiS, Wrocław 2014.

Literatura dla ambitnych

[CLRS] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Wprowadzenie do algorytmów*, wydanie 2, WNT, Warszawa 2004.

[GKP] R. L. Graham, D. E. Knuth, O. Patashnik, *Matematyka konkretna*, wydanie 4, PWN, Warszawa 2003.

[K] D. E. Knuth, *Sztuka programowania*, tom I, WNT, Warszawa 2002.

[L] W. Lipski, *Kombinatoryka dla programistów*, WNT, Warszawa 2009.

Literatura – algorytmy i programy

[Ko] D. Kopec.
Klasyczne problemy informatyki w Pythonie.
PWN, Warszawa, 2020.
Rozdział 4: https://ebook.pwn.pl/LP/lp_ebooki_it/

[Se] R. Sedgewick,
Algorytmy w C++. Część 5. Grafy,
Wyd. RM, Warszawa 2003.

Wymagania

Wymagania dotyczące wykładu:
Egzamin pisemny i ustny.

Szczegółowa lista zagadnień będzie podane na ostatnim wykładzie.

Wymagania

Wymagania dotyczące wykładu:
Egzamin pisemny i ustny.

Szczegółowa lista zagadnień będzie podane na ostatnim wykładzie.

Po każdym wykładzie zostanie opublikowana lista zadań na ćwiczenia.
Część zadań będzie podawana jako numery z książek:

[RR] R. Rębowski, *Matematyka dyskretna dla informatyków*,
[RR-JP] R. Rębowski, J. Płaskonka, *Zbiór zadań z matematyki dyskretnej dla informatyków*.

Wymagania

Wymagania dotyczące wykładu:
Egzamin pisemny i ustny.

Szczegółowa lista zagadnień będzie podane na ostatnim wykładzie.

Po każdym wykładzie zostanie opublikowana lista zadań na ćwiczenia.
Część zadań będzie podawana jako numery z książek:

[RR] R. Rębowski, *Matematyka dyskretna dla informatyków*,
[RR-JP] R. Rębowski, J. Płaskonka, *Zbiór zadań z matematyki dyskretnej dla informatyków*.

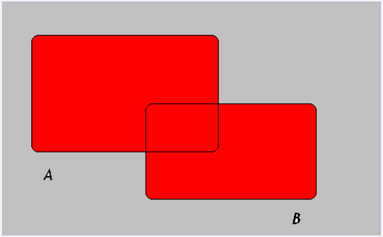
Zaliczenie ćwiczeń na ocenę co najmniej dobrą zwalnia z egzaminu.

Zdania, zbiory, relacje

- Rachunek zdań. Typy i operacje logiczne w językach programowania.
- Rachunek zbiorów. Działania mnogościowe w językach programowania.
- Teoria relacji. Zastosowania – relacyjne bazy danych.

Zdania, zbiory, relacje

- Rachunek zdań. Typy i operacje logiczne w językach programowania.
- Rachunek zbiorów. Działania mnogościowe w językach programowania.
- Teoria relacji. Zastosowania – relacyjne bazy danych.



Metody ilościowe

- Wybrane metody ilościowe. Metody zliczania.
- Rekurencje i iteracje.
- Algorytmy.

Metody ilościowe

- Wybrane metody ilościowe. Metody zliczania.
- Rekurencje i iteracje.
- Algorytmy.

Cauchy:

$$\binom{l+r}{k} = \sum_{t=0}^{k} \binom{l}{t} \binom{r}{k-t}.$$

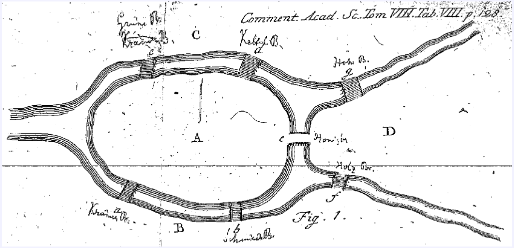
Teoria grafów

- Elementy teorii grafów. Grafy nieskierowane i skierowane. Drzewa.
- Macierze grafów. Komputerowa reprezentacja grafów. Kolorowanie grafów.
- Algorytmy teorii grafów. Drzewa minimalne. Najkrótsza droga.

Teoria grafów

- Elementy teorii grafów. Grafy nieskierowane i skierowane. Drzewa.
- Macierze grafów. Komputerowa reprezentacja grafów. Kolorowanie grafów.
- Algorytmy teorii grafów. Drzewa minimalne. Najkrótsza droga.

Euler:



Zdania i ich wartości logiczne

Zdanie przyjmuje tylko dwie wartości logiczne:
1 – prawda,
0 – fałsz.
Zdania będziemy oznaczać na ogół małymi literami: $p, q, r, s, \ldots$
Przykład. Zdaniami są
p: „księżyc świeci światłem odbitym”,
q: „słońce świeci na zielono”,
Jest oczywiste, że zdanie p jest prawdziwe, czyli ma wartość logiczną 1, zdanie q jest fałszywe – ma wartość logiczną 0.

Negacja

Negacja, czyli zaprzeczenie zdania p jest oznaczana jako $\neg p$ albo jako $\sim p$.
Negację określa tabela:

p	0	1
$\neg p$	1	0

Operacje logiczne dwuargumentowe:

$\wedge$ – p i q, (p oraz q), *koniunkcja*,
$\vee$ – p lub q, *alternatywa*,
$\oplus$ – p albo q (albo p albo q), *alternatywa wykluczająca*,
$\Longrightarrow$ – z p wynika q, *implikacja*,
$\equiv$ – p jest równoważne q,

Tabela działań dwuargumentowych:

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \Rightarrow q$	$p \equiv q$
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	0
1	1	1	1	0	1	1

Inna notacja operacji dwuargumentowych

$\wedge$ – p i q, (p oraz q), *koniunkcja*,
$\vee$ – p lub q, *alternatywa*,
$\veebar$ – p albo q (albo p albo q), *alternatywa wykluczająca*,
$\Longrightarrow$ – z p wynika q, *implikacja*,
$\Longleftrightarrow$ – p jest równoważne q,

Tabele działań dwuargumentowych:

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\veebar$	0	1
0	0	1
1	1	0

$\Longrightarrow$	0	1
0	1	1
1	0	1

$\Longleftrightarrow$	0	1
0	1	0
1	0	1

Operacje logiczne w programowaniu

Typy i wartości logiczne:

Typ	Język	0	1		
bool	C/C++	false	true	0	1
boolean	Pascal	false	true		
	Python	False	True		
	Maxima	false	true		

Operatory

	$\sim$	$\vee$	$\wedge$	$\veebar$
C/C++	`!`	`||`	`&&`	`^`
Pascal	`not`	`or`	`and`	`xor`
Python	`not`	`or`	`and`	
		`|`	`&`	`^`
Maxima	`not`	`or`	`and`	

Operatory logiczne i bitowe w języku Python

Operacje logiczne – przykłady

Zdanie „5 jest liczbą parzystą” $\vee$ „6 jest liczbą parzystą” jest prawdziwe.

Operacje logiczne – przykłady

Zdanie „5 jest liczbą parzystą” $\vee$ „6 jest liczbą parzystą” jest prawdziwe.
Zdanie „5 jest liczbą parzystą” $\wedge$ „6 jest liczbą parzystą” jest fałszywe.

Operacje logiczne – przykłady

Zdanie „5 jest liczbą parzystą” $\vee$ „6 jest liczbą parzystą” jest prawdziwe.
Zdanie „5 jest liczbą parzystą” $\wedge$ „6 jest liczbą parzystą” jest fałszywe.
Zdanie „n jest liczbą parzystą” $\iff$ „$n+1$ jest liczbą nieparzystą” jest prawdziwe.

Operacje logiczne – przykłady

Zdanie „5 jest liczbą parzystą” $\vee$ „6 jest liczbą parzystą” jest prawdziwe.
Zdanie „5 jest liczbą parzystą” $\wedge$ „6 jest liczbą parzystą” jest fałszywe.
Zdanie „n jest liczbą parzystą” $\iff$ „$n+1$ jest liczbą nieparzystą” jest prawdziwe.
Zdanie „x jest liczbą dodatnią” $\implies$ „istnieje $\sqrt{x}$” jest prawdziwe.

Tautologie

Zdanie, które jest zawsze prawdziwe, nazywa się prawem logicznym lub tautologią. Poniżej podano zestaw najważniejszych tautologii.

$\sim (\sim p) \iff p$	– prawo podwójnego przeczenia
$p \vee \sim p$	– prawo wyłączonego środka
$(p \wedge (q \vee r)) \iff ((p \wedge q) \vee (p \wedge r))$	– rozdzielność koniunkcji względem alternatywy
$(p \vee (q \wedge r)) \iff ((p \vee q) \wedge (p \vee r))$	– rozdzielność alternatywy względem koniunkcji
$(p \implies q) \wedge (q \implies r) \implies (p \implies r)$	– prawo przechodniości implikacji

Tautologie c.d.

$(\sim (p \vee q)) \iff ((\sim p) \wedge (\sim q))$ – prawo de Morgana dla alternatywy

$(\sim (p \wedge q)) \iff ((\sim p) \vee (\sim q))$ – prawo de Morgana dla koniunkcji

$(p \implies q) \iff ((\sim q) \implies (\sim p))$ – prawo transpozycji

$(p \wedge (p \implies q)) \implies q$ – reguła odrywania

Implikacja

Implikacja pełni rolę szczególną, a własność przechodniość implikacji jest podstawą dowodów matematycznych.

Zagadnienie dowodu matematycznego jest szczegółowo przedstawione w pkt. 1.4 książki R. Rębowskiego [R]

Czy to jest tautologia? (1)

Prawo przechodniości implikacji:

$$(p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow (p \Rightarrow r)\,.$$

Metoda *brute force*.

Czy to jest tautologia? (1)

Prawo przechodniości implikacji:

$$(p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow (p \Rightarrow r) \, .$$

Metoda *brute force*.

p	q	r	$(p \Rightarrow q)$	$(q \Rightarrow r)$	$(p \Rightarrow r)$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Czy to jest tautologia? (2)

Prawo przechodniości implikacji:

$$(p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow (p \Rightarrow r)\,.$$

Metoda bardziej wyszukana.

Czy to jest tautologia? (2)

Prawo przechodniości implikacji:

$$(p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow (p \Rightarrow r)\,.$$

Metoda bardziej wyszukana.

Implikacja jest fałszywa tylko wtedy, gdy poprzednik jest prawdziwy, a następnik jest fałszywy.

W tym przypadku musiałoby być $p = 1$ i $r = 0$.
Wtedy jeśli $q = 0$, to $(p \Rightarrow q) = 0$, a jeśli $q = 1$, to $(q \Rightarrow r)$.
W obu przypadkach $(p \Rightarrow q) \wedge (q \Rightarrow r) = 0$, co oznacza prawdziwość całej implikacji.

Formy zdaniowe

Forma zdaniowa to wyrażenie postaci $P(x)$, które staje się zdaniem po podstawieniu za x dowolnego elementu zbioru X (czyli $x \in X$), zwanego dziedziną formy zdaniowej $P(x)$. Zbiór tych elementów zbioru X, dla których $P(x)$ jest zdaniem prawdziwym, oznaczamy

$$\{x \in X : P(x)\}.$$

Inna nazwa formy zdaniowej, to predykat.

Przykład. Niech $\mathbb{R}$ oznacza zbiór liczb rzeczywistych.
Zbiór $\{x \in \mathbb{R} : x > 0\}$, to zbiór liczb rzeczywistych dodatnich.
Zbiór $\{x \in \mathbb{R} : x > 0 \wedge x < 1\}$, to odcinek $(0, 1)$ (bez końców).

Kwantyfikatory

Kwantyfikatorem nazywamy zwrot „dla każdego $x \in X$" (kwantyfikator ogólny) lub „istnieje $x \in X$ takie, że" (kwantyfikator szczegółowy).

Kwantyfikatory wiążą zmienną w formie zdaniowej, tworząc z niej zdanie.

Kwantyfikator ogólny oznaczamy symbolem $\bigwedge$, a kwantyfikator szczegółowy – symbolem $\bigvee$.

Kwantyfikatory

Kwantyfikatorem nazywamy zwrot „dla każdego $x \in X$” (kwantyfikator ogólny) lub „istnieje $x \in X$ takie, że” (kwantyfikator szczegółowy).
Kwantyfikatory wiążą zmienną w formie zdaniowej, tworząc z niej zdanie.
Kwantyfikator ogólny oznaczamy symbolem $\bigwedge$, a kwantyfikator szczegółowy – symbolem $\bigvee$.

Często używane symbole dla kwantyfikatorów, to $\bigwedge$ i $\bigvee$ dla kwantyfikatorów odpowiednio ogólnego i szczegółowego.

Dziedzina kwantyfikatora – przykład

Zasięg kwantyfikatora może być ograniczony tylko do pewnej dziedziny D.

Przykład. Zdanie

„dla każdej liczby rzeczywistej x istnieje liczba $y > 0$ taka, że $y^2 = x$",

nie jest prawdziwe.

Prawdziwe jest za to zdanie

„dla każdej liczby rzeczywistej $x \geq 0$ istnieje liczba $y > 0$ taka, że $y^2 = x$".

Formalnie:

$$\bigwedge_{x \geq 0} \bigvee_{y \geq 0} y^2 = x.$$

Tutaj $D = \{x : x \geq 0\} = [0, \infty)$.

Dziedzina kwantyfikatora – ogólnie

Dziedzina D jest pewien zbiór D taki, że $x \in D$.

$$\bigwedge_{x \in D} P(x)$$

oznacza:
„dla każdego $x \in D$ prawdziwe jest $P(x)$”,

$$\bigvee_{x \in D} P(x)$$

oznacza:
„istnieje $x \in D$ takie, że prawdziwe jest $P(x)$”.

Kwantyfikatory – przykłady

Załóżmy, że x i y są liczbami rzeczywistymi.

1. $\bigwedge_{x\in\mathbb{R}} x^2 \geq 0$ – jest zdaniem prawdziwym.
2. $\bigvee_{x\in\mathbb{R}} \sqrt{x} < 0$ – jest zdaniem fałszywym.
3. $\bigwedge_{x\in\mathbb{R}} \bigvee_{y\in\mathbb{R}} y > x$ – jest zdaniem prawdziwym.

Rachunek kwantyfikatorów – implikacje

Oczywiste jest , że

$$\bigwedge_{x \in D} P(x) \Longrightarrow \bigvee_{x \in D} P(x).$$

Rachunek kwantyfikatorów – implikacje

Oczywiste jest , że

$$\bigwedge_{x\in D} P(x) \Longrightarrow \bigvee_{x\in D} P(x).$$

Równie oczywiste jest, że

$$\sim\left(\bigvee_{x\in D} P(x) \Longrightarrow \bigwedge_{x\in D} P(x)\right).$$

Prawa de Morgana

$$\sim \left(\bigvee_{x \in D} P(x) \right) \iff \bigwedge_{x \in D} (\sim P(x))$$

$$\sim \left(\bigwedge_{x \in D} P(x) \right) \iff \bigvee_{x \in D} (\sim P(x))$$

Prawa de Morgana

$$\sim\left(\bigvee_{x\in D} P(x)\right) \iff \bigwedge_{x\in D} (\sim P(x))$$

$$\sim\left(\bigwedge_{x\in D} P(x)\right) \iff \bigvee_{x\in D} (\sim P(x))$$

Wniosek:

$$\bigvee_{x\in D} P(x) \iff \sim \bigwedge_{x\in D} (\sim P(x))$$

$$\bigwedge_{x\in D} P(x) \iff \sim \bigvee_{x\in D} (\sim P(x))$$

Prawa de Morgana – dla optymisty i pesymisty

- x – dzień, D – wszystkie dni,
- $P(x)$: x jest szczęśliwym dniem,
- $$\{x \in D : P(x)\}$$
  – wszystkie szczęśliwe dni.

Prawa de Morgana – dla optymisty i pesymisty

- x – dzień, D – wszystkie dni,
- $P(x)$: x jest szczęśliwym dniem,
- $$\{x \in D : P(x)\}$$
  – wszystkie szczęśliwe dni.

„Wszystkie dni są szczęśliwe” $\iff$ „Nie istnieją nieszczęśliwe dni”.
„Wszystkie dni są nieszczęśliwe” $\iff$ „Nie istnieją szczęśliwe dni”.
„Istnieją szczęśliwe dni” $\iff$ „Nie wszystkie dni są nieszczęśliwe”.
„Istnieją nieszczęśliwe dni” $\iff$ „Nie wszystkie dni są szczęśliwe”.

Prawa de Morgana – dla optymisty i pesymisty

- x – dzień, D – wszystkie dni,
- $P(x)$: x jest szczęśliwym dniem,
- $$\{x \in D : P(x)\}$$
  – wszystkie szczęśliwe dni.

„Wszystkie dni są szczęśliwe” $\iff$ „Nie istnieją nieszczęśliwe dni”.
„Wszystkie dni są nieszczęśliwe” $\iff$ „Nie istnieją szczęśliwe dni”.
„Istnieją szczęśliwe dni” $\iff$ „Nie wszystkie dni są nieszczęśliwe”.
„Istnieją nieszczęśliwe dni” $\iff$ „Nie wszystkie dni są szczęśliwe”.