

METODY FORMALNE INFORMATYKI

Marek Zaionc

<https://marekzaionc.staff.tcs.uj.edu.pl/>



Wykłady 1 i 2 w dniu 2 i 8 października 2024

Formuły:

Zmienne zdaniowe $Z = \{x, y, p, q, \dots\}$

- każda zmienna jest formułą
- jeżeli A i B są formułami to $A \Rightarrow B$, $A \vee B$, $A \wedge B$, $\neg A$ są formułami
- nic innego nie jest formułą

Wartościowanie to funkcja $v : Z \rightarrow \mathbb{B}$ gdzie $\mathbb{B}$ jest matrycą Boolowską, to znaczy $\mathbb{B} = \{0, 1\}$.

Wartościowanie określone na zmiennych zdaniowych rozszerzamy jednoznacznie na zbiór formuł:

- $v(A \vee B) = \max(v(A), v(B))$
- $v(A \wedge B) = \min(v(A), v(B))$
- $v(\neg A) = 1 - v(A)$
- $v(A \Rightarrow B) = \max(1 - v(A), v(B))$

Równoważność, tautologie, formuły spełnialne

Definicja

Dwie formuły A i B nazywamy równoważnymi gdy dla każdego wartościowania v jest tak że: $v(A) = v(B)$. Formuły równoważne oznaczamy $A \equiv B$.

Definicja

Formułę A nazywamy tautologią gdy dla każdego wartościowania v jest tak że $v(A) = 1$.

Definicja

Formułę A nazywamy spełnialną gdy dla pewnego wartościowania v jest tak że $v(A) = 1$.

Przykładowe równoważności:

- $\neg(A \vee B) \equiv \neg A \wedge \neg B$
- $\neg(A \Rightarrow B) \equiv A \wedge \neg B$
- $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

Przykładowe tautologie

- $A \Rightarrow A$
- $A \Rightarrow (B \Rightarrow A)$
- $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$
- $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$

Funkcje Boolowskie

Rozważamy wszystkie funkcje $\mathbb{B}^k \rightarrow \mathbb{B}$ dla wszystkich $k \in \mathbb{N}$. Funkcje takie nazywamy Boolowskimi. Dla $k = 1$ mamy:

	<i>numer</i>		
		0	1
	0	0	0
	1	0	1
	2	1	0
	3	1	1

Systemy funkcjonalnie pełne

numer funkcji	$x = 0$ $y = 0$	$x = 0$ $y = 1$	$x = 1$ $y = 0$	$x = 1$ $y = 1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Systemy funkcjonalnie pełne

numer funkcji	$x = 0$ $y = 0$	$x = 0$ $y = 1$	$x = 1$ $y = 0$	$x = 1$ $y = 1$	
0	0	0	0	0	$\mathbb{F}$
1	0	0	0	1	$\wedge$
2	0	0	1	0	$\neg(x \Rightarrow y)$
3	0	0	1	1	x
4	0	1	0	0	$\neg(y \Rightarrow x)$
5	0	1	0	1	y
6	0	1	1	0	$\neg \Leftrightarrow$
7	0	1	1	1	$\vee$
8	1	0	0	0	NOR
9	1	0	0	1	$\Leftrightarrow$
10	1	0	1	0	$\neg y$
11	1	0	1	1	$y \Rightarrow x$
12	1	1	0	0	$\neg x$
13	1	1	0	1	$x \Rightarrow y$
14	1	1	1	0	NAND
15	1	1	1	1	$\mathbb{T}$

Definicja

Skończony zbiór funkcji Boolowskich nazywamy funkcjonalnie pełnym gdy każda funkcja Boolowska daje się przedstawić jako złożenie funkcji z tego zbioru.

Systemy funkcjonalnie pełne, formuła Shannona

Lemat

Niech $f : \mathbb{B}^{n+1} \rightarrow \mathbb{B}$ będzie funkcją Boolowską. Zachodzi tak zwana: Shannon's expansion formula

$$f(x, y_1, \dots, y_n) = (x \wedge f(1, y_1, \dots, y_n)) \vee (\neg x \wedge f(0, y_1, \dots, y_n))$$

Twierdzenie

Zbiór $\{\vee, \wedge, \neg\}$ jest funkcjonalnie pełny.

Twierdzenie

Następujące systemy są funkcjonalnie pełne.

- $\{\vee, \neg\}$
- $\{\wedge, \neg\}$
- $\{\Rightarrow, \neg\}$
- $\{NOR\}$
- $\{NAND\}$

Formy normalne

Rozważmy język zbudowany z formuł atomowych przy pomocy spójników $\{\vee, \wedge, \neg\}$.

Definicja

- *Literałem nazywamy każdą zmienną lub jej zaprzeczenie.*
- *Klauzulą nazywamy alternatywę literałów.*
- *Formuła jest w koniunkcyjnej postaci normalnej gdy jest koniunkcją klauzul.*
- *Oznaczamy przez $\square$ klauzulę pustą.*

koniunktywna postać normalna

Twierdzenie

Dla każdej formuły istnieje (efektywnie) równoważna formuła w postaci normalnej

Definicja

Zbiór klauzul K jest spełnialny przez wartościowanie v gdy każda klauzula tego zbioru jest spełnialna przez v .

Twierdzenie

Dla każdej formuły istnieje (efektywnie) skończony zbiór klauzul, który jest spełniony przez te i tylko te wartościowania które spełniają formułę.

Sprawdzanie niespełnialności formuł.

Definicja

Rezolwentą dwóch klauzul

$$C_1 = x \vee y_1 \vee \dots \vee y_n$$

$$C_2 = \neg x \vee z_1 \vee \dots \vee z_k$$

nazywamy klauzulę $y_1 \vee \dots \vee y_n \vee z_1 \vee \dots \vee z_k$

Fakt

Jeżeli klauzule C_1, C_2 są spełnione przez wartościowanie v to każda rezolwenta klauzul C_1, C_2 też.

Definicja

Dowodem dla zbioru klauzul K jest ciąg klauzul $C_1, \dots, C_n$ taki, że każda klauzula tego dowodu jest bądź elementem zbioru klauzul K bądź jest otrzymana jako rezolwenta z klauzul wcześniejszych w dowodzie. Ostatnią klauzulą C_n dowodu jest $\square$.

Twierdzenie o niesprzeczności

Jeżeli istnieje dowód dla zbioru klauzul K to zbiór ten jest niespełnialny.

Pełność metody rezolucji

Twierdzenie o pełności

Jeżeli zbioru klauzul jest niespełnialny to ma dowód.

Algorytm poszukiwania dowodu

algorytm badający tautologiczność

Input: formuła ϕ

Output: Ocena czy ϕ jest tautologią.

Algorytm:

Problem SAT i 3SAT

Problem SAT

Wejście: Dana jest formuła.

Problem: Czy jest ona spełnialna ?

Status problemu: Nie wiadomo czy istnieje algorytm, który rozwiązuje SAT w czasie wielomianowym.

Status problemu SAT

SAT jest pierwszym znanym problemem NP- zupełnym. Dowód tego pochodzi od Stephen Cook, University of Toronto in 1971 i niezależnie przez Leonid Levin w 1973.



Stephen Cook, University of Toronto, zdjęcie z 1968 roku

Problem czy $P = NP$ jest pierwszym problemem milenijnym 2000 rok .

Problemy Hilberta 1900 rok .



Aksjomatyzacja rachunku zdań

Aksjomatyzacja fragmentu z implikacją i negacją

Rozważamy formuły zbudowane ze zmiennych zdaniowych przy pomocy implikacji i negacji Schematy aksjomatów: Dla dowolnych formuł A, B, C aksjomatami są:

Aksjomaty

- 1 $A \Rightarrow (B \Rightarrow A)$
- 2 $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$
- 3 $(\neg A \Rightarrow B) \Rightarrow ((\neg A \Rightarrow \neg B) \Rightarrow A)$

Reguła dowodzenia

$$\frac{A, A \Rightarrow B}{B} \quad (\textit{modus ponens})$$

Definicja

Dowodem formuły A jest skończony ciąg formuł $A_1, \dots, A_n$ o takiej własności że:

- *A_n jest formułą A*
- *dla każdego $i \leq n$ formuła A_i jest aksjomatem lub daje się oderwać z wcześniejszych wyrazów tego ciągu to znaczy istnieją takie $j, k < i$ że formuła A_k jest postaci $A_j \Rightarrow A_i$.*

Lemat

Przykładowy dowód aksjomatyczny formuły $A \Rightarrow A$.

Twierdzenie

(O niesprzeczności) Każda formuła dowodliwa jest tautologią.

Twierdzenie

(O pełności Emil Leon Post (1921)) Każda tautologia ma dowód.

Logika minimalna

Język: rachunek zdań jedynie z implikacją $\rightarrow$.

Definicja

Udowodnić implikację $A \rightarrow B$ to znaczy znaleźć funkcję f (obliczalną), która przekształci każdy dowód formuły A na dowód formuły B .

Przykładowe formuły dowodliwe w logice intuicjonistycznej

- $A \rightarrow A$
- $A \rightarrow (B \rightarrow A)$
- $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
- $((A \rightarrow A) \rightarrow A) \rightarrow A$

Przykładowa formuła będąca tautologią klasyczną i nie mająca dowodu w logice intuicjonistycznej

$$((A \rightarrow B) \rightarrow A) \rightarrow A$$