

Matematyka dyskretna

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. zimowy 2025/26

inf.ug.edu.pl/~amb/MaD

Andrzej M. Borzyszkowski (Instytut Informatyki) Matematyka dyskretna sem. zimowy 2025/26 1 / 22

Algebra Boole'a

Algebra Boole'a – główny przykład

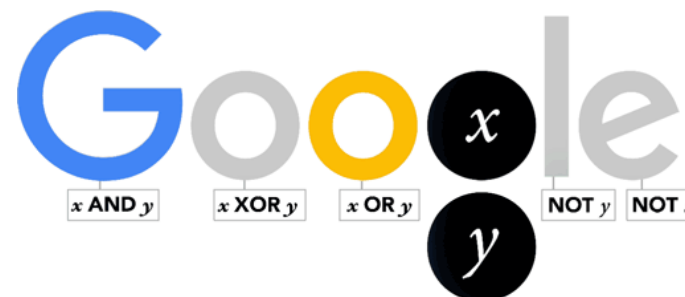
- $B = 2$, tj. $B = \{0, 1\}$, działania $\neg : B \rightarrow B$ oraz $\wedge, \vee : B^2 \rightarrow B$ zdefiniowane tabelkami

p	$\neg p$	p	q	$p \wedge q$	p	q	$p \vee q$
0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	1
1	0	1	0	0	1	0	1
1	0	1	1	1	1	1	1

- interpretujemy jako wartości logiczne:
prawda: 1, fałsz: 0, negacja: $\neg$, koniunkcja: $\wedge$ i dyzjunktja: $\vee$
- logika zdań: $p, q, \dots$ – zdania, które mają wartość logiczną, prawda lub fałsz
operatory $\neg, \wedge, \vee$ służą do budowy dalszych zdań

Andrzej M. Borzyszkowski (Instytut Informatyki) Matematyka dyskretna sem. zimowy 2025/26 3 / 22

Algebra Boole'a



dwusetna rocznica urodzin XI.2015

Andrzej M. Borzyszkowski (Instytut Informatyki) Matematyka dyskretna sem. zimowy 2025/26 2 / 22

Algebra Boole'a Algebra Boole'a – prawa

Algebra Boole'a – prawa

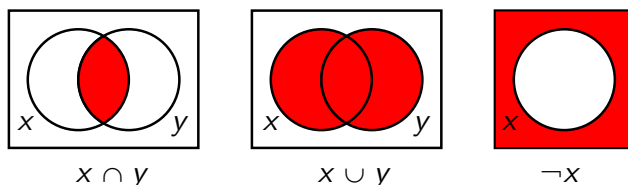
- zero i jedynka: $0 \wedge p = 0$, $0 \vee p = p$, $1 \wedge p = p$, $1 \vee p = 1$, $\neg 0 = 1$, $\neg 1 = 0$
- idempotentność: $\neg(\neg p) = p$, $p \wedge p = p$, $p \vee p = p$
- negacja: $p \wedge \neg p = 0$, $p \vee \neg p = 1$
- przemienność: $p \wedge q = q \wedge p$, i podobnie dla $\vee$
- łączność: $p \wedge (q \wedge r) = (p \wedge q) \wedge r$, i podobnie dla $\vee$
- rozdzielność obu działań:
 $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$
 $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$
- pochłanianie: $(p \wedge q) \vee p = p$, $(p \vee q) \wedge p = p$
- prawa de Morgana: $\neg(p \wedge q) = \neg p \vee \neg q$, $\neg(p \vee q) = \neg p \wedge \neg q$

Andrzej M. Borzyszkowski (Instytut Informatyki) Matematyka dyskretna sem. zimowy 2025/26 4 / 22

Algebra Boole'a – drugi główny przykład

- dany zbiór A , rozpatrujemy jego podzbiory $B = 2^A$
- możemy zinterpretować wszystkie powyższe działania
 - $0 = \emptyset, 1 = A$
 - $\neg X = A \setminus X$
 - $X \wedge Y = X \cap Y, X \vee Y = X \cup Y$
- spełnione są wszystkie prawa algebry Boole'a
- zauważmy, że $X \subseteq Y$ wtedy i tylko wtedy, gdy $X \cap Y = X$ wtedy i tylko wtedy, gdy $X \cup Y = Y$

- diagramy Venna



źródło: [wikimedia.org/wiki/File:Vennandornot.svg](https://commons.wikimedia.org/wiki/File:Vennandornot.svg)

Algebra Boole'a – inne podejście

- dla algebry Boole'a B definiujemy relację porządku na elementach:
 $p \leq q$ wtedy i tylko wtedy, gdy $p \wedge q = p$
- dla algebry $B = 2$ niewiele elementów jest w relacji: $0 \leq 1$
- dla algebry Boole'a podzbiorów ten porządek to $X \subseteq Y$
- wówczas $p \wedge q$ to największe r takie, że $r \leq p, q$
dw.: $p \wedge q \leq p$ bo $(p \wedge q) \wedge p = p \wedge q$, podobnie $p \wedge q \leq q$
niech $r \leq p, q$ czyli $r \wedge p = r \wedge q = r$, wówczas $r \wedge (p \wedge q) = r$
- podobnie $p \vee q$ to najmniejsze r takie, że $p, q \leq r$
 0 to najmniejszy element w B , a 1 jest największy
- można zacząć definicję od zbioru uporządkowanego, żądać istnienia elementów $0, 1, p \wedge q, p \vee q$ j.w.
żądać operacji $\neg$ i spełnienia praw algebry Boole'a

Algebra Boole'a – definicja aksjomatyczna

- definicja: zbiór B , stałe $0, 1 \in B$, działania $\neg : B \rightarrow B$ oraz $\wedge, \vee : B^2 \rightarrow B$, takie, że spełnione są podane wyżej prawa
- można oszczędnie definiować: zero, negacja, koniunkcja,
 $0 \wedge p = 0, p \wedge \neg p = 0, \neg 0 \wedge p = p, \neg \neg p = p, p \wedge p = p$,
przemienność, łączność i rozdzielność
- definiujemy $1 = \neg 0, p \vee q = \neg(\neg p \wedge \neg q)$
- inne prawa można udowodnić, np. pochłanianie:
 $p \vee (p \wedge q) = (p \wedge 1) \vee (p \wedge q) = p \wedge (1 \vee q) = p \wedge 1 = p$
- wiele pomysłów na oszczędne zdefiniowanie działań i aksjomatów,
niewygodnych człowieka, sensownych w innych celach
- twierdzenie Stone'a: każda algebra Boole'a jest izomorficzna z algebrą Boole'a zbiorów, czyli podalgebrą 2^A dla pewnego A
a skończona jest nawet izomorficzna z całym 2^A dla pewnego A

Wyrażenia boolowskie

- wyrażenia buduje się ze zmiennych, stałych, operacji, nawiasów
- stałe: 0 i 1
- operacje $\neg, \wedge, \vee$ i pochodne, np. $\oplus, \Rightarrow, \Leftrightarrow$
- nawiasy można opuszczać przyjmując siłę wiązania, najbardziej wiąże negacja, $\wedge$ bardziej niż $\vee$
- używa się też wielu innych notacji
 - negacja: $\bar{x}, \neg x, !x, \sim x$
 - koniunkcja, tak jak mnożenie: $x \cdot y, xy$
 - dysjunkcja jak dodawanie: $x + y$
- przykłady wyrażeń: $(X + Y)Z, \overline{(X + Y)Z} \cdot \overline{XY + Z}, (W \oplus U) \oplus \neg V$
- wyrażenia można interpretować w różnych algebrach, np. w 2 , albo w 2^A

Funkcje o wartościach boolowskich

- dana algebra Boole'a B i zbiór X
- na zbiorze funkcji $X \rightarrow B$ określamy działania „po współrzędnych”
 - $0(x) = 0, 1(x) = 1$
 - $(\neg p)(x) = \neg(p(x))$
 - $(p \wedge q)(x) = p(x) \wedge q(x), (p \vee q)(x) = p(x) \vee q(x)$
 - podobnie inne operacje, np. $\oplus$
- w ten sposób $X \rightarrow B$ staje się też algebrą Boole'a
- np. $2^A = A \rightarrow 2$, podzbiór jako funkcja charakterystyczna
 - $\chi_\emptyset(a) = 0, \chi_A(a) = 1$
 - $\chi_{\neg Y}(a) = \neg \chi_Y(a)$
 - $\chi_{Y \cup Z}(a) = \chi_Y(a) \vee \chi_Z(a), \chi_{Y \cap Z}(a) = \chi_Y(a) \wedge \chi_Z(a),$

Własności innych operacji

- xor:
 - przemienność: $p \oplus q = q \oplus p$
 - łączność: $(p \oplus q) \oplus r = p \oplus (q \oplus r)$
 - rozdzielność $\wedge$ względem $\oplus$: $p \wedge (q \oplus r) = (p \wedge q) \oplus (p \wedge r)$
 - ale analogiczna własność dla $\vee$ nie zachodzi
 - zero i jedynka: $p \oplus 0 = p, p \oplus 1 = \neg p, p \oplus p = 0, p \oplus \neg p = 1$
- wszystkie operacje można zdefiniować w oparciu o funktor Scheffera (NAND):
 - $\neg p = p|p$
 - $p \wedge q = (p|q)|(p|q)$
 - $p \vee q = (p|p)|(q|q)$
- również NOR ma tę własność
- i żadna inna pojedyncza operacja nie jest wystarczająca

Inne operacje w algebrach Boole'a

- różnica (dopełnienie): $p \setminus q = p \wedge \neg q$
- implikacja: $p \Rightarrow q = \neg p \vee q$
- alternatywa wykluczająca (xor): $p \oplus q = p \wedge \neg q \vee \neg p \wedge q$
- równoważność: $p \Leftrightarrow q = (p \Rightarrow q) \wedge (q \Rightarrow p)$ czyli $\neg(p \oplus q)$
- funktor Scheffera (bramka NAND): $p|q = \neg(p \wedge q)$
- bramka NOR: $\neg(p \vee q)$

p	q	$p \setminus q$	p	q	$p \Rightarrow q$	p	q	$p \oplus q$	p	q	$p q$
0	0	0	0	0	1	0	0	0	0	0	1
0	1	0	0	1	1	0	1	1	0	1	1
1	0	1	1	0	0	1	0	1	1	0	1
1	1	0	1	1	1	1	1	0	1	1	0

- i to właściwie wszystko, pozostają jeszcze rzuty $(p, q) \mapsto p,$
 $(p, q) \mapsto q$, stałe $(p, q) \mapsto 0$ i ich negacje

Równości w algebrze Boole'a

- dane są dwa wyrażenia boolowskie U i W , pytamy czy w algebrze Boole'a $A \rightarrow B$ zachodzi równość $U = W$
- odp.: na pewno zachodzi, jeśli sprawdzimy równość dla algebry B – działania w $A \rightarrow B$ są po współrzędnych
- każda algebra Boole'a jest podalgebrą $A \rightarrow 2$
- wniosek: wystarczy badać wartość wyrażeń boolowskich w 2

Przykłady

czy $p + (p \cdot q) = p$?

p	q	$p \cdot q$	$p + pq$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

równość zachodzi

czy $p + (q \oplus r) = (p + q) \oplus (p + r)$?

p	q	r	$q \oplus r$	Lew	$p+q$	$p+r$	Pr
0	0	0	0	0	0	0	0
0	0	1	1	1	0	1	1
0	1	0	1	1	1	0	1
0	1	1	0	0	1	1	0
1	0	0	0	1	1	1	0
1	0	1	1	1	1	1	0
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	0

nie ma rozdzielności dodawania względem $\oplus$

Zastosowania algebr Boole'a, c.d.

- ciągi bitów = funkcje o wartościach boolowskich, $n \rightarrow 2$
 - operacje „po współrzędnych” (*bitwise*)
 - przesunięcia, z obcięciem, z przeniesieniem, cyklicznie
 - $sh(u)(x) = u(x-1)$ i trzeba wyjaśnić co z $x=0$, n
- grafika komputerowa: piksel jest ciągiem bitów
- operacje na pikselach są wyrażone jako operacje boolowskie
- hasła w wyszukiwarce
 - można żądać wystąpienia wszystkich: koniunkcja
 - można żądać co najmniej jednego: dyzjunkcja
 - można żądać, by te hasła nie wystąpiły (negacja dyzjunkcji)
- „full Boolean logic” – wyszukiwarka pozwala wpisywać dowolne wyrażenia logiczne
- na ogół są pośrednie możliwości

Zastosowania algebr Boole'a

- logika zdań: wartości logiczne i operacje na nich
- teoria zbiorów: zbiory i operacje na nich
- obwody elektryczne: przełącznik, ma wartość logiczną „przewodzi”, „nie przewodzi”
 - bramki logiczne: negacja, podwójna negacja
 - AND, OR, XOR, ich negacje
 - wszystkie dadzą się wyrazić bramką NAND
 - prawa algebry Boole'a – upraszczanie obwodów przełączników

negacja, and, or, xor
oraz ich negacjenegacja negacji jest buforem,
nie zmienia wartości sygnału
ale opóźnia lekko

Wyrażenia boolowskie w językach programowania

- języki oparte na C:
 - wartości logiczne: fałsz – liczba zero, pusty napis
prawda – dowolna liczba czy napis inne niż fałsz
 - negacja: !
 - koniunkcja: &&, dyzjunkcja: ||
 - równość argumentów: if (x==y)
uwaga: if (x=y) ma znaczenie i to całkiem inne niż równość
 - porównania: <=, >=, !=
 - przesunięcia: x<<n pomnożenie przez 2ⁿ
x>>n podzielenie przez 2ⁿ
 - operacje na ciągach bitów: &, |, xor: ^
 - patrz: wikipedia.org/wiki/Bitwise_operations_in_C
 - uwaga: a & b == c będzie rozumiane jako a&(b==c)

Działania na bitach w C

```
int main(){
    printf(" 3 && -6 = %d\n", 3 && -6);
    printf(" 3 || -6 = %d\n", 3 || -6);
    printf(" 3 & -6 = %d\n", 3 & -6);
    printf(" 3 | -6 = %d\n", 3 | -6);
    printf(" 3 ^ -6 = %d\n", 3 ^ -6);
    return 0;
}
```

wypisze na ekranie

```
3 && -6 = 1
3 || -6 = 1
3 & -6 = 2
3 | -6 = -5
3 ^ -6 = -7
```

Przykłady funkcji boolowskich

- każda funkcja boolowska jest jednoznacznie opisana tabelką wartości, np. arbitralna funkcja $f : B^3 \rightarrow B$
- funkcja progowa $T_k^n : B^n \rightarrow B$ odpowiada na pytanie „czy co najmniej k argumentów jest 1?”

p	q	r	$f(p, q, r)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

p	q	r	$T_2^3(p, q, r)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Funkcje boolowskie

- funkcje postaci $f : B^n \rightarrow B$ gdy $B = 2$
- $n = 0$: dwie stałe
- $n = 1$: identyczność, negacja i stałe
- $n = 2$: działania w algebrze Boole'a, jest ich $2^4 = 16$
- każde wyrażenie boolowskie wyznacza funkcję boolowską

- ta sama funkcja ma wiele wyrażen

- np. $p \Rightarrow q$ oraz $\neg p \vee q$ wyznaczają tę samą funkcję f

- każde dwa wyrażenia W i V , których równość udowodnimy wyznaczają tę samą funkcję

p	q	$f(p, q)$
0	0	1
0	1	1
1	0	0
1	1	1

Specjalne postaci funkcji

- warto wyrazić funkcję boolowską za pomocą wyrażenia
- ale nie ma jednoznacznego sposobu
- dla funkcji progowych można rozumować następująco
 - koniunkcja $x_{i1} \wedge x_{i2} \dots \wedge x_{ik}$ zachodzi jeśli te właśnie zmienne mają wartość 1
 - dysjunkcja wszystkich możliwych wyborów k spośród n zmiennych będzie zachodzić, jeśli istnieje k zmiennych o wartości 1
 - $T_1^3 = p + q + r$, $T_2^3 = p \cdot q + p \cdot r + q \cdot r$, $T_3^3 = p \cdot q \cdot r$
- w przypadku ogólnym rozważamy dwa rodzaje znormalizowanych wyrażen dla funkcji:
 - dysjunkcyjną postać normalną, DNF oraz
 - koniunkcyjną postać normalną, CNF

Dyzjunkcyjna postać normalna DNF

- $f = \bigvee_i K_i$ a każde K_i jest koniunkcją zmiennych i ich negacji (w każdym składniku występują wszystkie zmienne)
- algorytm budowy:
 - w tabelce wartości funkcji szukamy wierszy z wartością **1**
 - dla każdego wiersza definiujemy składnik $x_1^\varepsilon \cdot x_2^\varepsilon \cdot \dots \cdot x_n^\varepsilon$, gdzie $x^\varepsilon = \bar{x}$ jeśli wartość zmiennej x jest **0**
 - funkcja jest dyzjunkcją takich koniunkcji
- np. dla wcześniejszej funkcji $f = \bar{p}\bar{q}r + \bar{p}qr + pq\bar{r}$
- wiersze niewymienione we wzorze będą dawać wartość 0, bo w każdym składniku któraś ze zmiennych nie będzie pasować i koniunkcja będzie równa 0

Koniunkcyjna postać normalna CNF

- pojęcie dualne do DNF
- $f = \bigwedge_i D_i$, każde D_i jest dyzjunkcją zmiennych i ich negacji
- algorytm budowy:
 - w tabelce wartości funkcji szukamy wierszy z wartością **0**
 - dla każdego wiersza definiujemy czynnik $x_1^\varepsilon + x_2^\varepsilon + \dots + x_n^\varepsilon$, gdzie $x^\varepsilon = \bar{x}$ jeśli wartość x w tym wierszu jest **1**
 - funkcja jest koniunkcją takich dyzjunkcji
- np. dla wcześniejszej funkcji

$$f = (p + q + r) \cdot (p + \bar{q} + r) \cdot (\bar{p} + q + r) \cdot (\bar{p} + q + \bar{r}) \cdot (\bar{p} + \bar{q} + \bar{r})$$
- wiersze niewymienione we wzorze będą dawać wartość 1, bo w każdym czynniku któraś ze zmiennych będzie równa 1 i dyzjunkcja będzie też równa 1