STU FIIT
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Dátové štruktúry a algoritmy

Binárne Rozhodovacie Diagramy

28. 03. 2023

letný semester 2022/2023

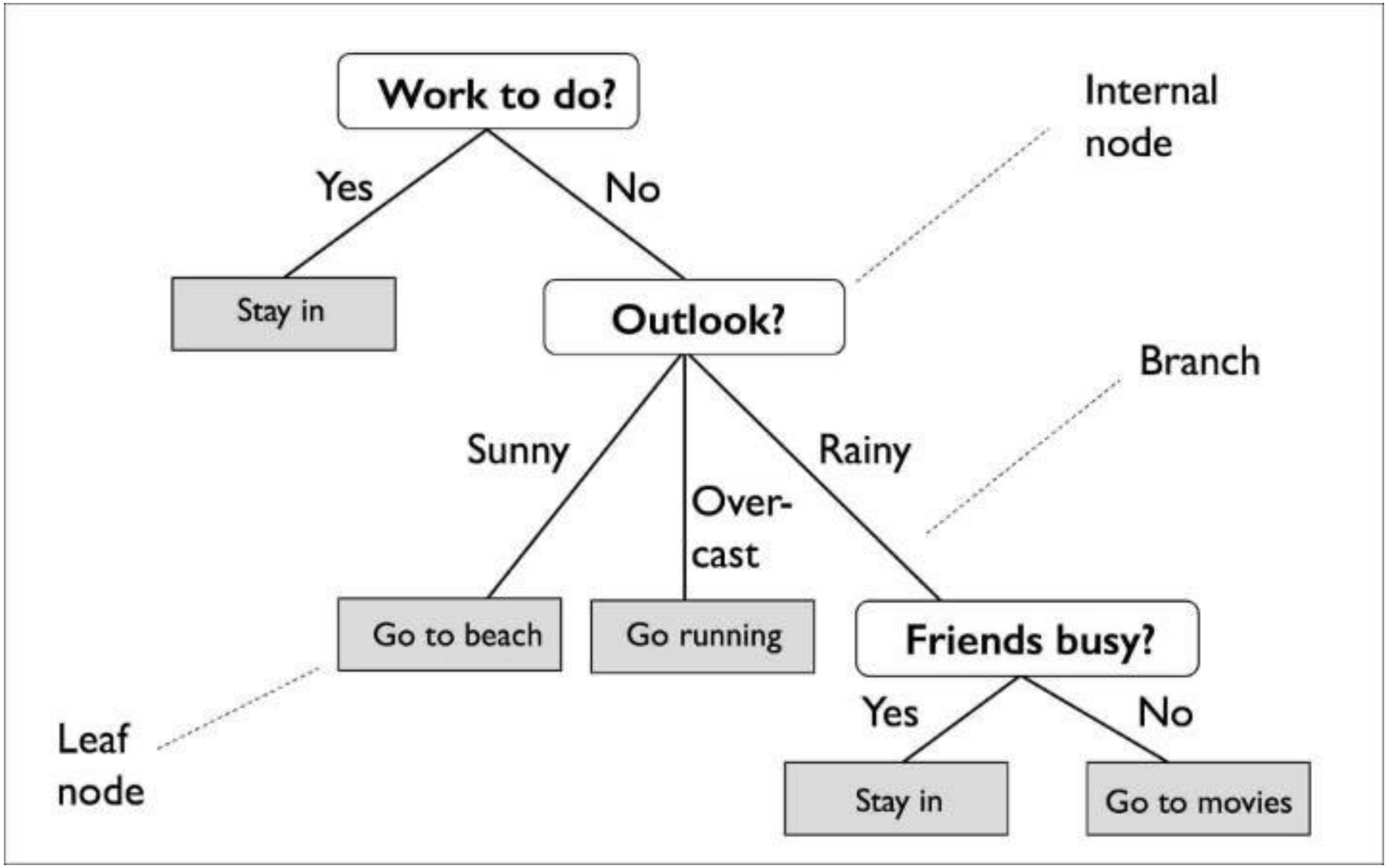
Lukáš Kohútka

Binárny rozhodovací diagram

- V úplnom tvare je to binárny strom
- Neslúži ako ADT slovník/dynamická množina
- Slúži na rozhodovanie
  - Rozhodovací strom (decision tree)
- Rozhodovanie prebieha prechodom od začiatku stromu (koreň) do jeho konca (list)
- Každý vnútorný uzol predstavuje jedno čiastkové rozhodnutie, list = výsledné rozhodnutie
- Celkové rozhodnutie = celá cesta koreň → list

Rozhodovací strom

- Môže byť binárny, ale nemusí



Binárny rozhodovací diagram

- Binary Decision Diagram (skratka BDD)
- Dátová štruktúra
- Má tvar ako binárny strom
- Slúži na rozhodovanie (rôzne aplikácie)
- Jednou z najpoužívanejších aplikácií je: Reprezentácia ľubovoľnej Booleovskej funkcie

Reprezentácie Booleovských funkcií

- Pravdivostná tabuľka
- Vektor
  - to isté ako pravdivostná tabuľka, len jej výstupy
- Karnaughova mapa
- Výraz
  - t.j. rovnica (napr. Y = A . B + C)
  - Normálové formy
    - DNF – súčet súčinov
    - KNF – súčin súčtov
- BDD

XOR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Booleovská algebra (Boolean Algebra)

- Booleovská algebra (B-algebra) je taká množina **B**, ktorá obsahuje nezávisle premenné **a, b, c, ...** s dvoma diskrétnymi hodnotami **0** a **1**, v ktorej sú definované operácie **logického súčtu**, **súčinu** a **inverzie** tak, že platí nasledovný súbor axióm:

1. **Neutrálnosť konštanty:** Existujú prvky **0, 1 ∈ B**, že pre ľubovoľnú premennú **a ∈ B** platí:

   $$\mathbf{a + 0 = a}$$

   $$\mathbf{a \,.\, 1 = a}$$

2. **Vylúčenie tretieho:** Pre každý prvok **a ∈ B** existuje jemu zodpovedajúci negovaný prvok taký, že platí:

   $$\mathbf{a + \overline{a} = 1}$$

   $$\mathbf{a \,.\, \overline{a} = 0}$$

Booleovská algebra (Boolean Algebra)

3. **Komutatívnosť:** Pre každé dva prvky **a, b ∈ B** platí:

   a + b = b + a

   a . b = b . a

4. **Distributívnosť:** Pre každé tri prvky **a, b, c ∈ B** platí:

   a + b . c = (a + b) . (a + c)

   a . (b + c) = a . b + a . c

Pravidlá v Booleovskej algebre

Pravidlá agresívnosti konštanty

$$a \,.\, 0 = 0$$

$$a + 1 = 1$$

Pravidlá pohltenia inverzie:

$$a + \overline{a}\, b = a + b$$

$$a \,.\, (\overline{a} + b) = a \,.\, b$$

Asociatívne pravidlá:

$$a + (b + c) = (a + b) + c$$

$$a \,.\, (b \,.\, c) = (a \,.\, b) \,.\, c$$

Pravidlo dvojnásobnej inverzie:

$$\overline{\overline{a}} = a$$

Pravidlá pohltenia (absorbcie).

$$a + a = a$$

$$a \,.\, a = a$$

De Morganove pravidlá:

$$\overline{a + b} = \overline{a} \,.\, \overline{b}$$

$$\overline{a \,.\, b} = \overline{a} + \overline{b}$$

De Morganove pravidlá sa môžu rozšíriť aj na funkcie o **n** premenných. Platí:

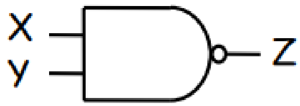
$$\overline{a_1 + a_2 + \ldots + a_n} = \overline{a}_1 \,.\, \overline{a}_2 \,.\, \ldots \,.\, \overline{a}_n$$

$$\overline{a_1 . a_2 \ldots a_n} = \overline{a}_1 + \overline{a}_2 + \ldots + \overline{a}_n$$

Základné logické hradlá

Gate	Symbol	Truth-Table	Expression

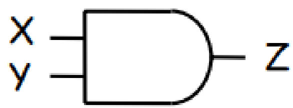
NAND



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$$Z = \overline{X \bullet Y}$$

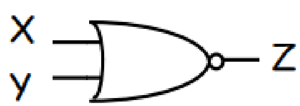
AND



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

$$Z = X \bullet Y$$

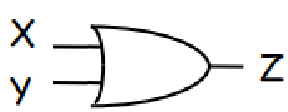
NOR



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

$$Z = \overline{X + Y}$$

OR



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

$$Z = X + Y$$

Základné logické hradlá

Gate	Symbol	Truth-Table	Expression

XOR
$(X \oplus Y)$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

$Z = X\,\overline{Y} + \overline{X}\,Y$
X or Y but not both
("inequality", "difference")

XNOR
$\overline{(X \oplus Y)}$

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

$Z = \overline{X}\,\overline{Y} + X\,Y$
X and Y the same
("equality")

Widely used in arithmetic structures such as adders and multipliers

Redukcia výrazu

A	B	C	Y
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**

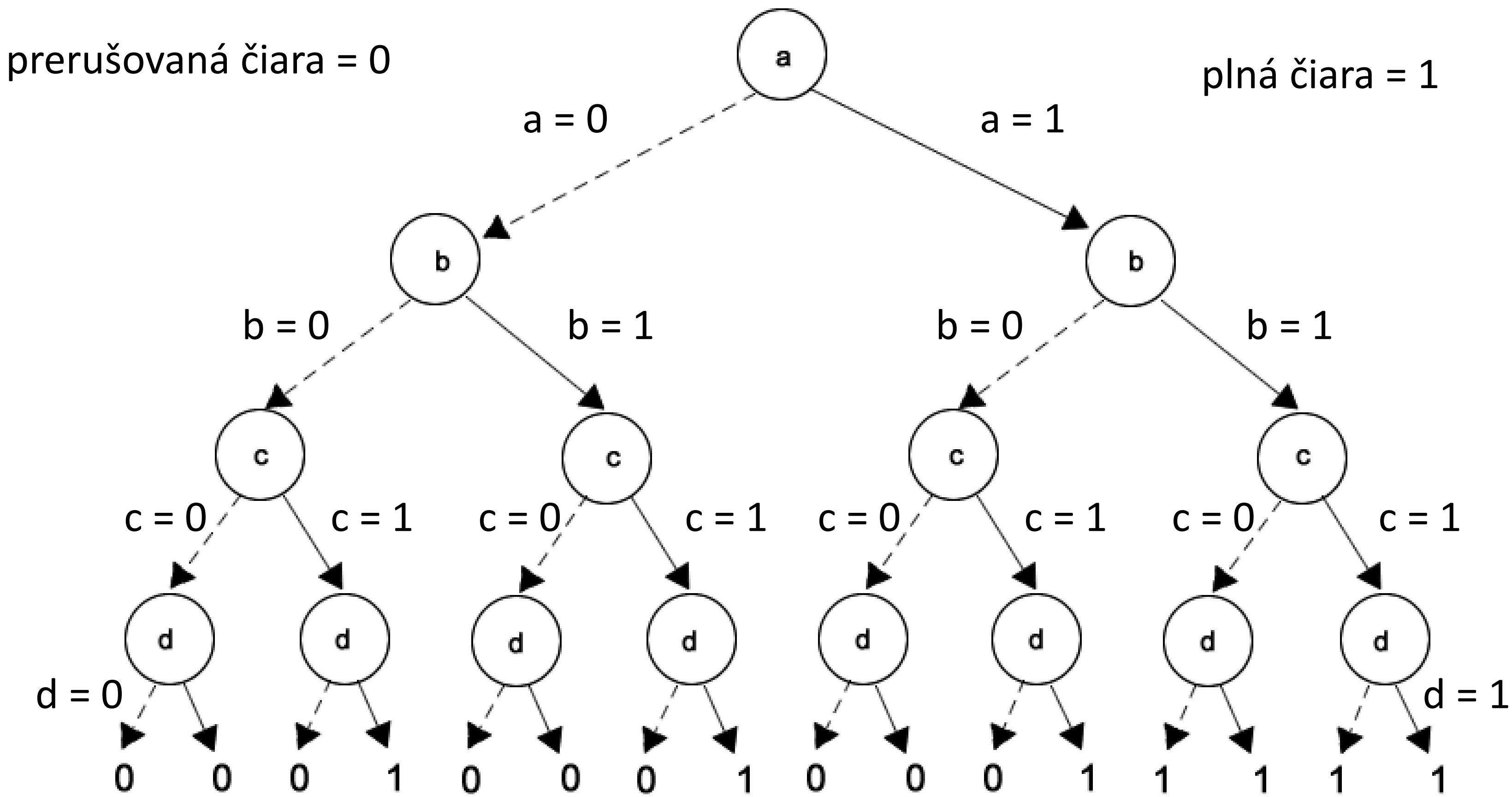
$$Y = \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$$

$$Y = (\overline{A} + A).B.C + A.\overline{B}.C + A.B.\overline{C}$$

$$Y = A.B + A.C + B.C$$

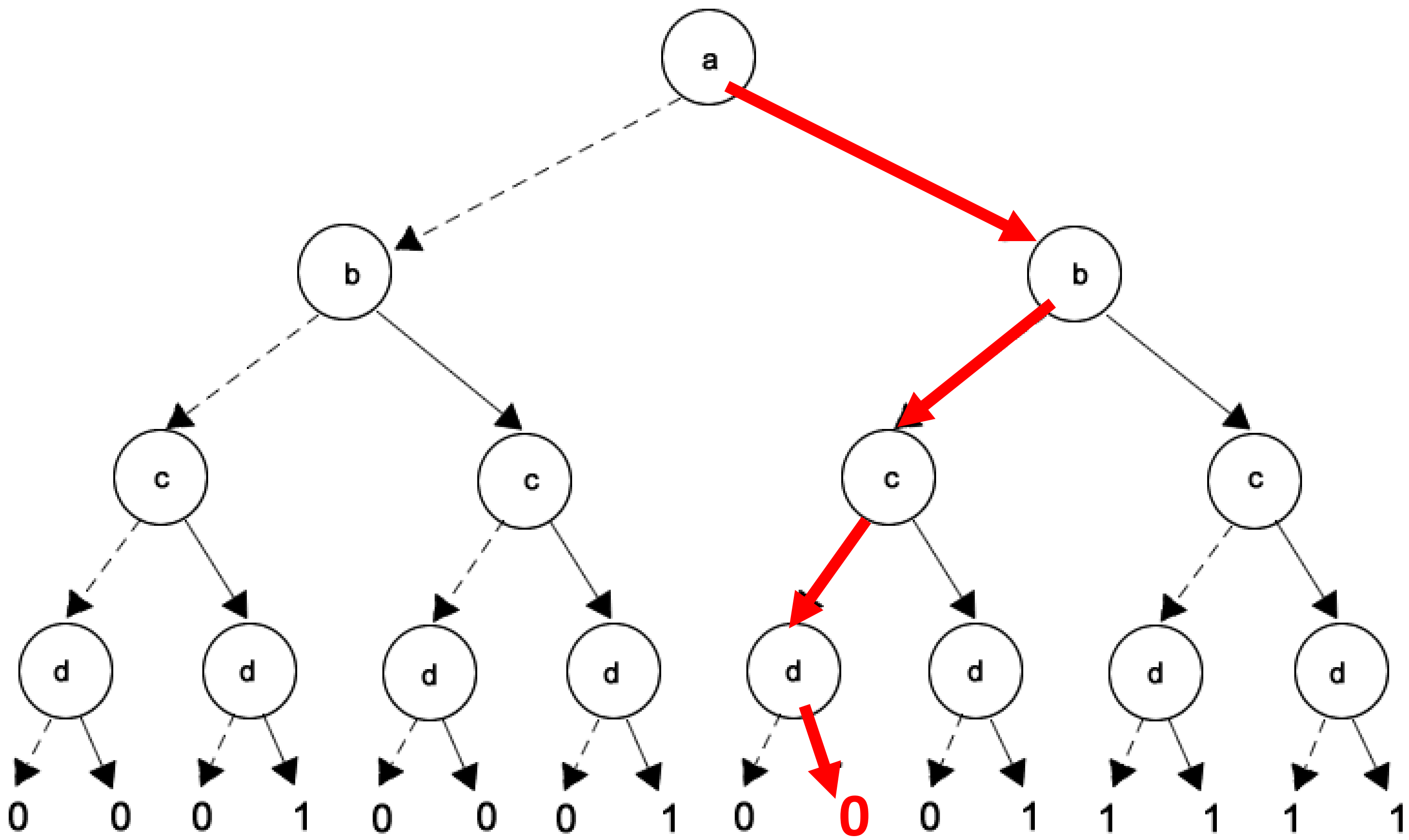
Stačí nám teda 3x 2-vstupové hradlo AND a 1x 3-vstupový OR

Binárny rozhodovací diagram



Binárny rozhodovací diagram

Aký je výsledok pre a = 1, b = 0, c = 0, d = 1 ?

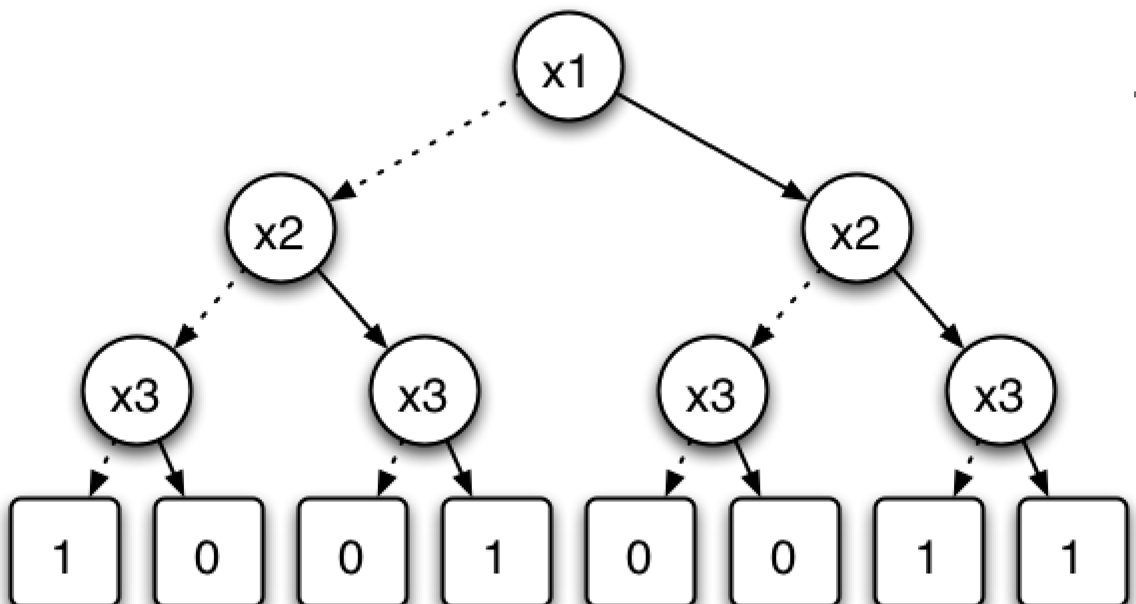


Binary Decision Diagram (BDD)

- Dá sa ňou reprezentovať ľubovoľná Booleovská funkcia
  - Alternatíva pre pravdivostnú tabuľku, vektor alebo Karnaughovu mapu

$$f(x_1, x_2, x_3) = (\neg x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2) \vee (x_2 \wedge x_3)$$

$$\bar{x}_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 + x_2 x_3$$ (alternatívny zápis)



x1	x2	x3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

vektor

10010011

Zostrojenie nového BDD

- Nový BDD sa môže zostrojiť relatívne jednoducho z ostatných spôsobov opisu Booleovskej funkcie
  - Bez ohľadu na to, akú konkrétnu Booleovskú funkciu opisujeme
- Dva prístupy:
  - Zhora nadol
    - postupnou dekompozíciou (rozkladom) podľa jednotlivých premenných
    - každá premenná predstavuje jednu úroveň v BDD
  - Zdola nahor
    - Postupným skladaním výstupov
    - Tiež každá premenná je jedna úroveň v BDD

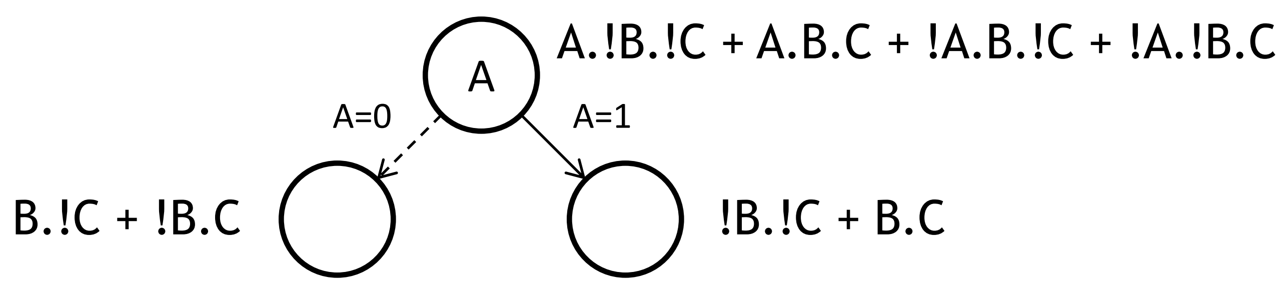
Zostrojenie nového BDD – zhora nadol

- Shannonova dekompozícia (Shannon decomposition alebo Shannon extension)

$$f(X_1, X_2, \ldots, X_n) = X_1 \cdot f(1, X_2, \ldots, X_n) + X_1' \cdot f(0, X_2, \ldots, X_n)$$

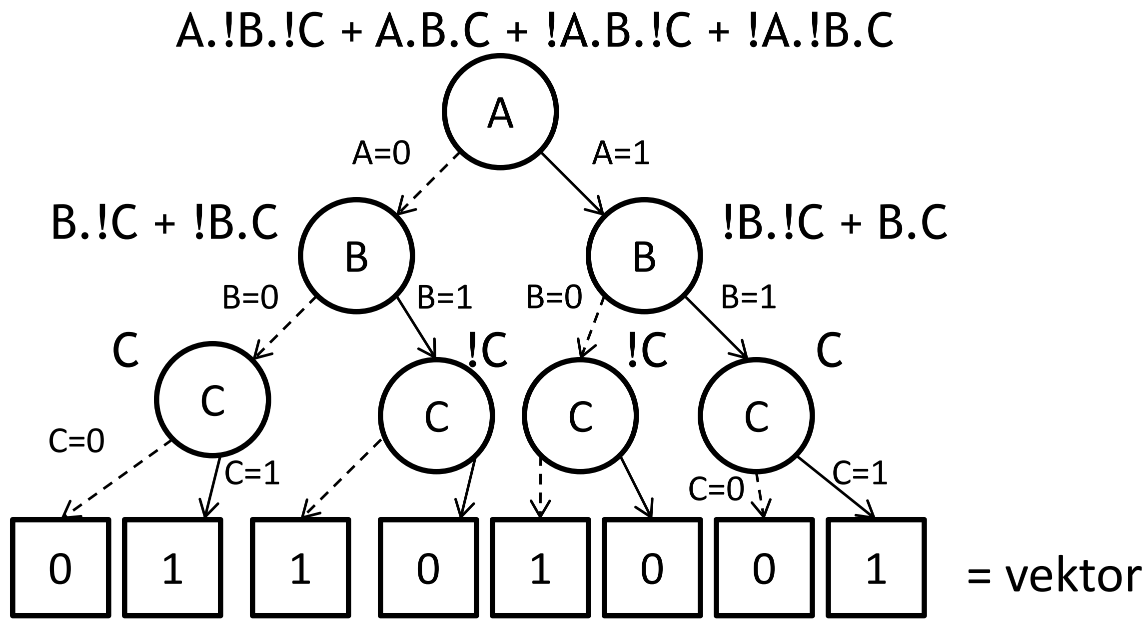
- Ak vytvárame BDD z výrazu v tvare DNF, napr.:
  - $Y = A.\overline{B}.\overline{C} + A.B.C + \overline{A}.B.\overline{C} + \overline{A}.\overline{B}.C$

1. Vytvoríme koreň, ktorý reprezentuje celý výraz
2. Vyberieme si jednu premennú, napr. A (tým pádom koreň je riadený premennou A)
3. $Y = A\,.(\overline{B}.\overline{C} + B.C) + \overline{A}\,.(B.\overline{C} + \overline{B}.C)$
4. Vytvoríme potomkov a vložíme zostatkové časti do nich
5. Opakujeme kroky 2 až 4 pre každú premennú (t.j. N-krát, kde N je počet vstupných premenných Booleovskej funkcie)



Zostrojenie nového BDD – zhora nadol

- Takže vyberieme zase nejakú premennú (napr. B)
- A pre všetky uzly v novej úrovni urobíme opäť Shannonovu dekompozíciu podľa premennej B
- B.!C + !B.C sa rozdelí na B.(!C) + !B.(C)
- !B.!C + B.C sa rozdelí na B.(C) + !B.(!C)
- Nakoniec predstavuje uzol len poslednú premennú alebo jej negovaný tvar



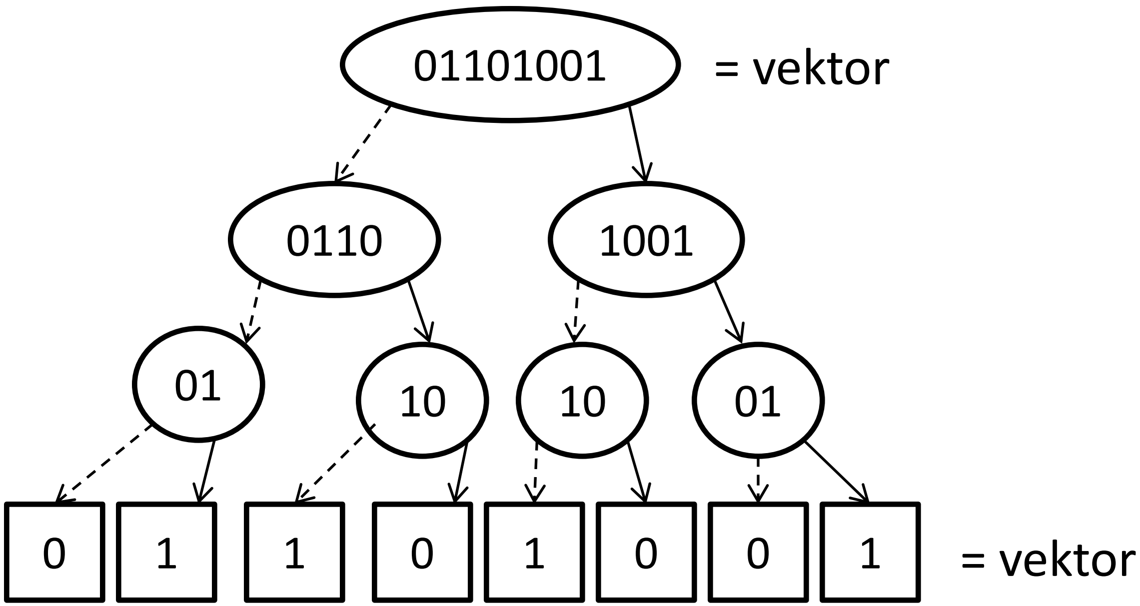
Zostrojenie nového BDD – zdola nahor

- Videli sme, že výsledný riadok s nulami a jednotkami je totožný s vektorom
- Preto ak chceme zostrojiť BDD z Booleovskej funkcie opísanej formou vektora (alebo pravdivostnej tabuľky), môžeme tak urobiť prístupom zdola nahor
- Majme napr. vektor 01101001
- Najprv vytvoríme koncové uzly (listy) s týmito hodnotami v rovnakom poradí
  - Vytvoríme ako pole (koncových) uzlov

0	1	1	0	1	0	0	1	= vektor

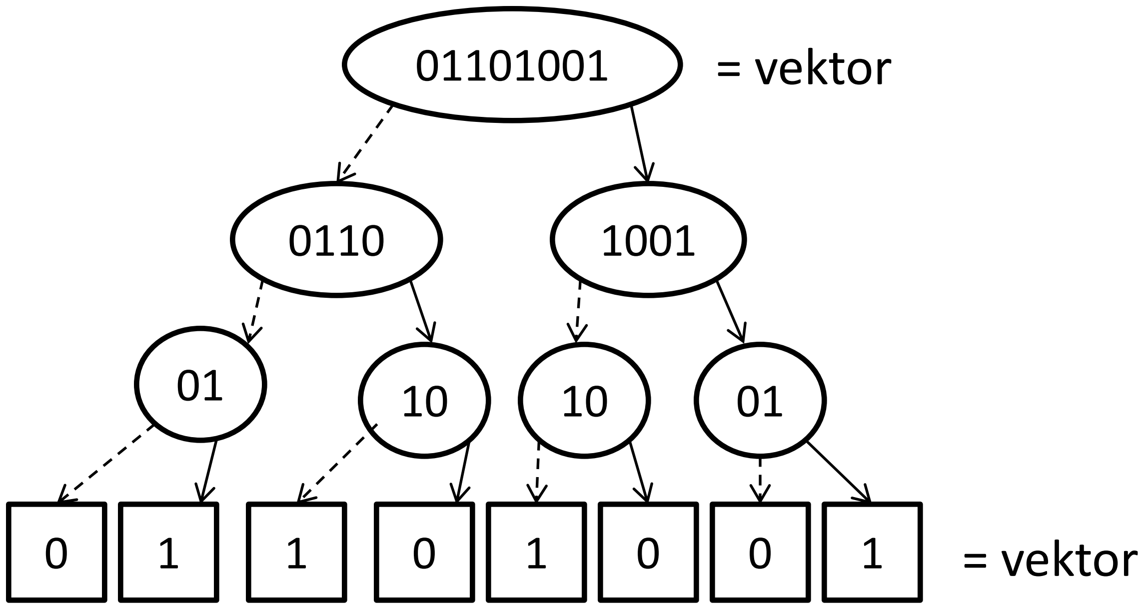
Zostrojenie nového BDD – zdola nahor

- Vezmeme vždy dvojicu susediacich uzlov a spojíme ich dokopy pridaním rodiča
- Rodič reprezentuje zlúčenú kombináciu hodnôt potomkov
- Opakujeme
- Poradie premenných je fixne dané podľa poradia vektora



Zostrojenie nového BDD - zhora nadol

- Vektor → BDD
  - možné tiež realizovať prístupom zhora nadol
- Vektor delíme vždy na polovice
- Poradie je fixne dané, tak ako vo vektore
- Napr. vektor 01101001

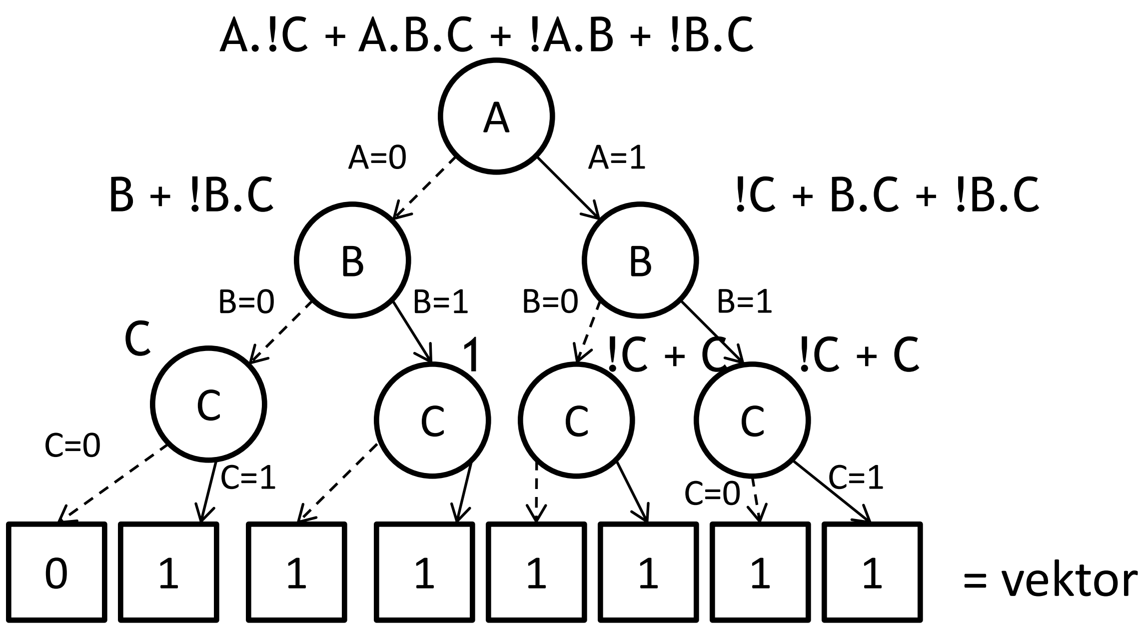


Shannonova dekompozícia výrazu

- Rozdelíme výraz na 2 časti

$$f(X_1, X_2, \ldots, X_n) = X_1 \cdot f(1, X_2, \ldots, X_n) + X_1' \cdot f(0, X_2, \ldots, X_n)$$

- Ak nám niekde premenná chýba (nie je prítomná ani v normálnom tvare ani negovanom), tak tá časť výrazu sa použije pre obidva nasledovníky
- Napr.:



Ďakujem za pozornosť

NABUDÚCE PREZENČNE (veľká aula)