

# Komponenty PC, dvojková soustava, Booleova algebra

## Převody čísel mezi pozičními soustavami

Ne vždy máme k dispozici obvyklých 10 číslic: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Někdy jenom 2 (tj. 0 a 1), jindy 8 (tj.  $2^3$ ) nebo 16 (tj.  $2^4$ ) nebo dokonce 36 (10 číslic + 26 písmen). Každá číslice v poziční soustavě má svoji

- **jmenovitou hodnotu** – hodnota samotné číslice bez ohledu na pozici v čísle
- **pozici** – pořadové číslo zprava (...3, 2, 1, 0)
- **váhu** – mocnina základu na pozici (...8, 4, 2, 1)
- **hodnotu** – součin váhy a jmenovité hodnoty číslice (např.  $0 * 8 = 0$ ,  $1 * 8 = 8$ ); součet všech těchto hodnot dává hodnotu celého čísla

Při převodech mezi soustavami používáme dva algoritmy (volíme podle toho, ve které soustavě se nám lépe počítá, příp. přepočítáváme přes desítkovou):

- násobíme jmenovité hodnoty číslic váhami a sečteme součiny:  $(101)_2 = (1*(2*2*1) + 0*(2*1) + 1*(1))_{10} = 5$
- naopak: postupně dělíme celou hodnotu čísla základem soustavy a sepisujeme zbytky:  
 $5 =$   
 $4 + 1 =$   
 $(2 + 0) * 2 + 1 =$   
 $((1 * 2) + 0) * 2 + 1)_{10} =$   
 $(101)_2$

Jde to ovšem i takhle (z desítkové do dvojkové prvním způsobem):

$$(107 = 1 * 100 + 0 * 10 + 7)_{10} = (1 * 1100100 + 0 * 1010 + 111 = 1100100 + 111 = 1101011)_2$$

Přepočty názorně viz [tabulka](#)

## Úkoly

- Ukažte dvě vstupní a jedno výstupní zařízení počítače.
- Ukažte v počítači procesor, paměť RAM, základní desku a zdroj.
- Převedte číslo 83 z desítkové do dvojkové soustavy a ukažte postup výpočtu.
- Převedte číslo 1100011 z dvojkové do desítkové soustavy a ukažte postup výpočtu.

**Pracujte ve dvojicích. Za každý splněný úkol si dejte oba zapsat čárku do hodnocení!**

---

## Výroky

Výroky jsou věty buď pravdivé, nebo nepravdivé. Např.:

- Jedenáct je prvočíslo.
- Jedna a jedna jsou tři.
- $3 + 4 = 7$
- Žížala je obratlovec.

Není-li věta ani pravdivá, ani nepravdivá, není to výrok. Např.:

- Umyj si ruce.

- Kolik je hodin?

Potíže jsou s nejednoznačnými větami, které jsou buď nejednoznačné (třeba o krvavém západu), přibližné (hodně, málo), subjektivní (ošklivý / hezký), závislé na okolnostech nebo souvislostech (venku je chladno -- kde venku? komu je chladno?)

Výroky můžeme spojovat výrokovými spojkami (též operacemi nebo funkcemi). Tak vznikají **výroky složené**. Jejich pravdivost a nepravdivost určujeme pomocí

- pravdiel (příp. zákonů výrokové logiky)
- pravdivostních tabulek (nejsou nekonečné, ba ani příliš rozsáhlé)

Viz [logika](#) nebo [Výroková logika I](#) nebo [Propositional Logic](#).

Na pokusy (právě tak jako na programování umělé inteligence nebo expertních systémů) se hodí jazyk *Prolog*, zjednodušenou verzi najdete [zde](#).

## Úkoly

### 1. Zadejte program:

```
prsi.
mrzne.
jeChladno <= prsi.
jeVlhko <= prsi.
jeNaledi <= prsi and mrzne.
```

Vyzkoušejte dotazy:

- ? jeVlhko.
- ? jeNaledi.

*Pozn.: každý dotaz se musí vyhodnotit zvlášť, zvláštním spuštěním prologu.*

### 2. Zadejte program:

```
jeRodic(adam,abel).
jeRodic(adam,kain).
jeRodic(adam,jarmila).
jeRodic(kain,enocho).
jePredek(X,Y) <= jeRodic(X,Y).
jePredek(X,Y) <= jeRodic(X,Z) and jePredek(Z,Y).
```

Vyzkoušejte dotazy:

- ? jeRodic(adam,X).
- ? jePredek(adam,X).

### 3. Doplňte další tvrzení a pravidla:

- Adam je muž.
- Eva je žena.
- Eva je rodič Abela.
- Eva je rodič Kaina.
- Otec osoby X je rodič osoby X a zároveň muž.

- Matka osoby X je rodič osoby X a zároveň žena.

Vyzkoušejte dotazy:

- Je Adam otcem Kaina?
- Je Adam otcem Enocha?
- Je Eva matkou Abela?

#### 4. Vymyslete pravidlo, které určí, že dvě osoby jsou bratři

jsouBratri(X,Y) <= ...

Vyzkoušejte, zda

- Kain a Abel jsou bratři
- Enoch a Adam jsou bratři
- Kain a Jarmila jsou bratři

**Pracujte ve dvojicích. Za každý splněný úkol si dejte oba zapsat čárku do hodnocení!**

---

### Booleova logika, Booleova algebra

Nápad [George Boolea](#) z poloviny 19. století – tenkrát se mu každý smál. Po 100 letech využili jeho nápad konstruktéři prvních počítačů ...a využíváme jej stále (a Booleovi se už neposmíváme).

- logický součin (konjunkce, "a"):  $0 \cdot 0 = 0$ ,  $0 \cdot 1 = 0$ ,  $1 \cdot 0 = 0$ ,  $1 \cdot 1 = 1$
- logický součet (disjunkce, "nebo"):  $0 + 0 = 0$ ,  $0 + 1 = 1$ ,  $1 + 0 = 1$ ,  **$1 + 1 = 1$**
- negace (doplněk do 1 – z 0 dělá 1 a naopak):  $1 - 0 = 1$ ,  $1 - 1 = 0$

Je celkem 16 logických funkcí 2 argumentů.

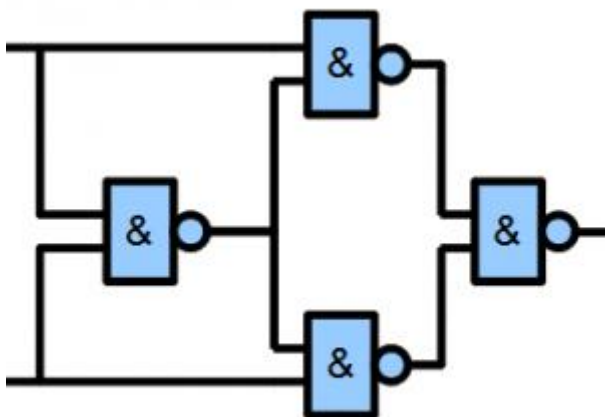
| x | 0101 |  | název funkce                         |
|---|------|--|--------------------------------------|
| y | 0011 |  |                                      |
| f | 0000 |  | 0                                    |
|   | 0001 |  | konjunkce                            |
|   | 0010 |  | negovaná implikace $y \rightarrow x$ |
|   | 0011 |  | y                                    |
|   | 0100 |  | negovaná implikace $x \rightarrow y$ |
|   | 0101 |  | x                                    |
|   | 0110 |  | xor (nonekvivalence, výlučné nebo)   |
|   | 0111 |  | disjunkce                            |
|   | 1000 |  | nor (negovaná disjunkce)             |
|   | 1001 |  | ekvivalence $x=y$                    |
|   | 1010 |  | negace x                             |
|   | 1011 |  | implikace $x \rightarrow y$          |
|   | 1100 |  | negace y                             |
|   | 1101 |  | implikace $y \rightarrow x$          |
|   | 1110 |  | nand (negovaná konjunkce)            |

Některé funkce se dají vyjádřit pomocí jiných, např.  $(a \vee b)$  se dá zapsat pomocí konjunkce a negací jako  $(a' \wedge b')$ . Soubor funkcí, pomocí nichž se dají vyjádřit všechny ostatní, se nazývá **úplný soubor**. V logice se obvykle vystačí se 3 funkcemi: konjunkce, disjunkce a negace. V [elektronice](#) stačí jediná, typicky NAND (negovaná konjunkce) – integrovaný obvod *Texas Instruments SN7400* z roku 1966 obsahuje 4 nandy a dá se z něj zapojit cokoli (složité logické funkce mnoha proměnných, aritmetické jednotky, paměti... třeba celý počítač). Tato unifikace umožnila vyrábět velké množství velmi laciných součástí pro konstrukci počítačů.

Viz [Booleova algebra](#), [Booleova logika](#) nebo [tato tabulka](#).

## Úkoly

- V tabulkovém kalkulátoru vytvořte vzorce na výpočet operací AND, OR a NOT.
- V tabulkovém kalkulátoru vytvořte vzorec na výpočet operace NAND (negovaná konjunkce).
- Napište tabulku pravdivostních hodnot operace NAND.
- Napište tabulku pravdivostních hodnot funkce  $f(x,y)$ , kterou realizuje schema na obrázku. Co je to za funkci? K čemu může sloužit?



| x | y | f   |
|---|---|-----|
| 0 | 0 | ??? |
| 0 | 1 | ??? |
| 1 | 0 | ??? |
| 1 | 1 | ??? |

**Pracujte ve dvojicích. Za každý splněný úkol si dejte oba zapsat čárku do hodnocení!**