Obsah

0. Úvod.

Stejně jako první vydání skript, i toto jejich druhé, upravené vydání je určeno studentům oboru matematické inženýrství na Fakultě strojního inženýrství Vysokého učení technického v Brně. Má sloužit jako učební text jednosemestrálního předmětu *Metody diskrétní matematiky*, který je v tomto oboru vyučován. Předmět si klade za cíl seznámit studenty se základními metodami diskrétní matematiky, které nacházejí bohaté uplatnění v technické praxi, především v informatice. Význam diskrétní matematiky rapidně vzrůstá s pronikáním počítačů do všech oblastí lidské činnosti, takže v současné době je diskrétní matematika z hlediska praktických aplikací neméně důležitá než matematika spojitá.

Protože skripta mají sloužit studentům inženýrské a nikoliv matematické specializace, jsou psána se snahou o co největší srozumitelnost a vyhýbají se náročnějším matematickým úvahám a konstrukcím. Vyžadují jen znalosti středoškolské matematiky a vědomosti získané v kurzu *Lineární algebra* v prvním ročníku studia matematického inženýrství - viz skripta [8], ze kterých je převzato základní označení (zde jen připomeňme, že symboly $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Z}^+$, $\mathbb{Q}$, $\mathbb{R}$ a $\mathbb{R}^+$ po řadě označují množiny přirozených, celých, celých nezáporných, racionálních, reálných a reálných nezáporných čísel).

V předmětu *Metody diskrétní matematiky*, tedy ani v předkládaných skriptech, není zahrnuta teorie grafů, která tvoří významnou součást diskrétní matematiky. S touto teorií budou studenti matematického inženýrství seznámeni zvlášť, a to v kurzu *Grafy a algoritmy*.

První kapitola skript pojednává o jednom z nejobecnějších matematických pojmů, o (binárních) relacích mezi množinami. Ve druhé kapitole jsou diskutována zobrazení, která jsou chápána jako speciální relace mezi množinami. Třetí kapitola se zabývá relacemi na množině, přičemž nejdůležitější z těchto relací jsou studovány v následujících dvou kapitolách: tolerance a ekvivalence v kapitole 4. a uspořádané množiny v kapitole 5. Kapitola 6. pojednává o svazech a kapitola 7. o Booleových svazech. Booleovy funkce jsou pak diskutovány v kapitole 8. Devátá kapitola se zabývá logickými a spínačovými obvody jako konkrétními aplikacemi Booleových svazů a funkcí. V kapitole 10. jsou diskutovány formální jazyky s důrazem na jazyky regulární a v navazující 11. a 12. kapitole jsou studovány konečné automaty (deterministické akceptory) a gramatiky. Závěrečná 13. kapitola je věnována teorii kódování, přesněji teorii samoopravných (lineárních) kódů.

Výklad každé kapitoly je demonstrován na příkladech a pro kontrolu pochopení probírané látky slouží uvedená cvičení. S dalšími významnými partiemi diskrétní matematiky, které nebylo možno zařadit do těchto skript z důvodu jejich omezeného rozsahu, se může čtenář seznámit v některé z učebnic uvedených v citované literatuře.

V Brně 12. listopadu 2004

1. Relace mezi množinami

V běžném životě se často setkáváme se situacemi, kdy existuje jistý vztah mezi dvojicemi nějakých objektů. K popsání takovýchto vztahů slouží v matematice pojem relace.

1.1. Definice. Buďte A, B množiny. *Relací* z A do B rozumíme libovolnou podmnožinu $R \subseteq A \times B$.

Připomeňme, že symbolem $A \times B$ značíme kartézský součin množin A a B, tedy množinu, jejímiž prvky jsou právě všechny dvojice (a, b), kde $a \in A$ a $b \in B$ (přičemž prvky množiny A jsou vždy na prvním místě v těchto dvojicích a prvky množiny B jsou vždy na místě druhém). Relace R z A do B je pak množina, která je tvořena některými z těchto dvojic. Přitom připouštíme i krajní situace $R = \emptyset$ (tzv. *prázdná* relace) a $R = A \times B$ (tzv. *univerzální* relace). Jak je obvyklé, místo $(a, b) \in R$ budeme psát aRb a místo $(a, b) \notin R$ budeme psát $a\bar{R}b$. Poznamenejme, že místo relace se v literatuře také používá pojmu korespondence.

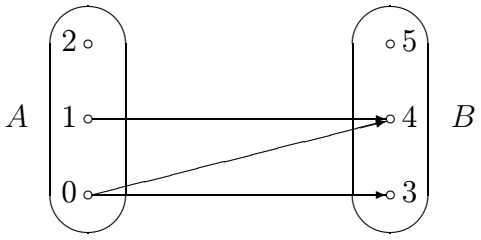
1.2. Příklady (1) Buď T množina všech učitelů na jisté škole a S množina všech předmětů vyučovaných na této škole v daném školním roce. Pak $R = \{(t, s) \in T \times S;$ učitel t vyučuje předmět $s\}$ je relace z T do S.

(2) Buď P množina všech bodů v rovině a L množina všech přímek v rovině. Jestliže pro libovolný bod $p \in P$ a libovolnou přímku $l \in L$ položíme $pRl \Leftrightarrow p$ leží na l, pak R je relace z P do L.

Buď R relace z konečné množiny $A = \{a_1, a_2, ..., a_m\}$ do konečné množiny $B = \{b_1, b_2, ..., b_n\}$. Pak relaci R je možno reprezentovat její maticí, což je matice $M = (m_{ij})$ typu $m \times n$ tvořená jen nulami a jedničkami, přičemž $m_{ij} = 0$, právě když $a_i\bar{R}b_j$, a $m_{ij} = 1$, právě když a_iRb_j $(i = 1, ..., m;\ j = 1, ..., n)$. Matice M relace R se někdy zapisuje ve formě tabulky, která obsahuje navíc nultý řádek tvořený prvky $b_1, ..., b_n$ a nultý sloupec tvořený prvky $a_1, ..., a_m$.

Jiným způsobem reprezentace relace R je její *graf*. Obdržíme jej tak, že prvky množin A a B znázorníme jako body a z prvku a_i vedeme orientovanou hranu (šipku) do prvku b_j, právě když a_iRb_j $(i = 1, ..., m;\ j = 1, ..., n)$.

1.3. Příklad. Nechť $A = \{0, 1, 2\}$, $B = \{3, 4, 5\}$ a $R = \{(0, 3), (0, 4), (1, 4)\}$. Pak R je relace z A do B, jejíž maticí je $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$. Graf relace R má tvar



1.4. Definice. Buď R relace z A do B. *Inverzní relací* k relaci R rozumíme relaci R^{-1} z B do A danou vztahem $R^{-1} = \{(b, a) \in B \times A;\ aRb\}$ (tj. vztahem $bR^{-1}a \Leftrightarrow aRb$).

Z matice relace R obdržíme matici relace R^{-1} transponováním, z grafu relace R obdržíme graf relace R^{-1} obrácením orientace všech hran.

1.5. Příklad. Buď R relace z $\mathbb{R}$ do $\mathbb{R}^+$ daná vztahem $mRn \Leftrightarrow n = m^2$. Pak $nR^{-1}m$, právě když $m = \pm\sqrt{n}$.

1.6. Cvičení. Dokažte, že pro libovolné relace R a S z A do B platí $(R^{-1})^{-1} = R$, $R \subseteq S \Rightarrow R^{-1} \subseteq S^{-1}$, $(R \cup S)^{-1} = R^{-1} \cup S^{-1}$, $(R \cap S)^{-1} = R^{-1} \cap S^{-1}$, $(R - S)^{-1} = R^{-1} - S^{-1}$.

1.7. Definice. Buďte $R \subseteq A \times B$, $S \subseteq B \times C$ relace. Relaci $RS \subseteq A \times C$ definovanou vztahem $RS = \{(a, c) \in A \times C$; existuje $b \in B$ tak, že aRb a $bSc\}$ nazýváme *složením* relací R a S.

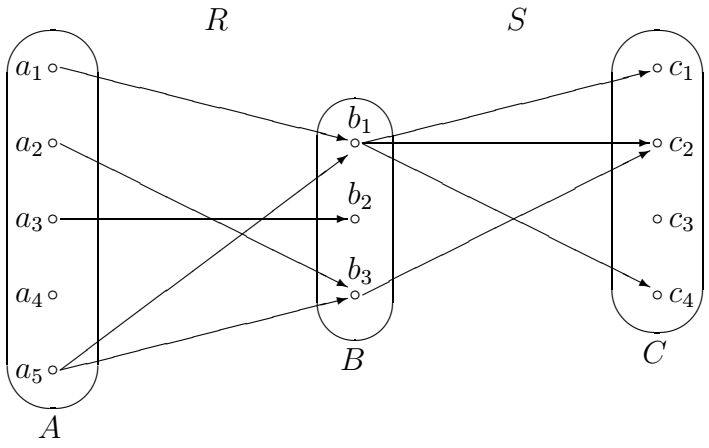
Jestliže M_R je matice relace R a M_S je matice relace S, pak matici relace RS obdržíme z matice $M_R \cdot M_S$ (kde $\cdot$ značí maticový součin) nahrazením všech prvků větších než 1 jedničkami. Graf relace RS obdržíme z grafů relací R a S tak, že z prvku $a \in A$ vedeme orientovanou hranu do prvku $c \in C$ právě tehdy, když existuje prvek $b \in B$ takový, že v grafu relace R vede orientovaná hrana z a do b a v grafu relace S vede orientovaná hrana z b do c.

1.8. Příklad. Nechť $A = \{a_1, a_2, a_3, a_4, a_5\}$, $B = \{b_1, b_2, b_3\}$, $C = \{c_1, c_2, c_3, c_4\}$ a nechť $R = \{(a_1, b_1), (a_2, b_3), (a_3, b_2), (a_5, b_1), (a_5, b_3)\} \subseteq A \times B$, $S = \{(b_1, c_1), (b_1, c_2), (b_1, c_4), (b_3, c_2)\} \subseteq B \times C$. Pak $RS = \{(a_1, c_1), (a_1, c_2), (a_1, c_4), (a_2, c_2), (a_5, c_1), (a_5, c_2), (a_5, c_4)\}$.

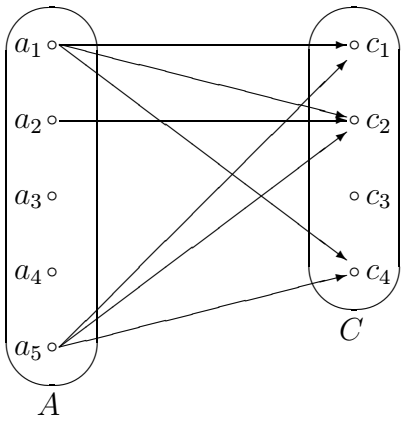
Zřejmě platí $M_R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ a $M_S = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$. Odtud dostáváme

$$M_R \cdot M_S = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix}, \text{ takže } M_{RS} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}.$$

Grafy relací R a S mají následující tvar:



Tedy graf relace RS má tvar:



1.9. Věta Nechť $R \subseteq A \times B$, $S \subseteq B \times C$, $T \subseteq C \times D$ jsou relace. Pak platí $(RS)T = R(ST)$ (tj. skládání relací je asociativní).

Důkaz. Nechť $(a, d) \in A \times D$ je libovolný prvek. Pak $(a, d) \in (RS)T \Leftrightarrow$ existuje prvek $c \in C$ tak, že $(a, c) \in RS$ a $(c, d) \in T \Leftrightarrow$ existují prvky $c \in C$ a $b \in B$ tak, že $(a, b) \in R$, $(b, c) \in S$ a $(c, d) \in T \Leftrightarrow$ existuje prvek $b \in B$ tak, že $(a, b) \in R$ a $(b, d) \in ST \Leftrightarrow (a, d) \in R(ST)$. Tedy $(RS)T = R(ST)$.

V důsledku věty 1.9 místo $(RS)T$ a $R(ST)$ píšeme stručně RST.

1.10. Věta. Nechť $R \subseteq A \times B$ a $S \subseteq B \times C$ jsou relace. Pak platí $(RS)^{-1} = S^{-1}R^{-1}$.

Důkaz. Buď $(c, a) \in C \times A$ libovolný prvek. Pak platí $(c, a) \in (RS)^{-1} \Leftrightarrow (a, c) \in RS \Leftrightarrow$ existuje prvek $b \in B$ tak, že $(a, b) \in R$ a $(b, c) \in S \Leftrightarrow$ existuje prvek $b \in B$ tak, že $(c, b) \in S^{-1}$ a $(b, a) \in R^{-1} \Leftrightarrow (c, a) \in S^{-1}R^{-1}$. Tedy $(RS)^{-1} = S^{-1}R^{-1}$.

1.11. Příklad. Uvažujme relace R a S z příkladu 1.8. Pak $(RS)^{-1} = \{(c_1, a_1), (c_2, a_1), (c_4, a_1), (c_2, a_2), (c_1, a_5), (c_2, a_5), (c_4, a_5)\}$. Protože $R^{-1} = \{(b_1, a_1), (b_3, a_2), (b_2, a_3), (b_1, a_5), (b_3, a_5)\}$ a protože $S^{-1} = \{(c_1, b_1), (c_2, b_1), (c_4, b_1), (c_2, b_3)\}$, dostáváme $S^{-1}R^{-1} = \{(c_1, a_1), (c_1, a_5), (c_2, a_1), (c_2, a_5), (c_4, a_1), (c_4, a_5), (c_2, a_2)\}$. Skutečně tedy platí $(RS)^{-1} = S^{-1}R^{-1}$.

1.12. Cvičení. Dokažte, že pro relace R, S, T, U a příslušné složené relace (pokud jsou definovány)

a) z podmínek $R \subseteq S$ a $T \subseteq U$ plyne $RT \subseteq SU$,

b) platí distributivní zákony vzhledem ke sjednocení, tj.

$(R \cup S)T = RT \cup ST$ a $R(S \cup T) = RS \cup RT$,

c) obecně neplatí distributivní zákon vzhledem k průniku ani vzhledem k množinovému rozdílu, nýbrž platí jen inkluze

$(R \cap S)T \subseteq RT \cap ST$, $R(S \cap T) \subseteq RS \cap RT$,

$(R - S)T \supseteq RT - ST$ a $R(S - T) \supseteq RS - RT$.

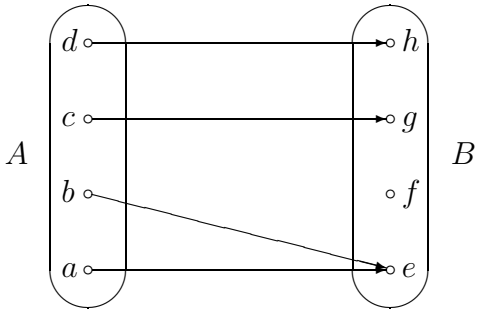
2. Zobrazení

Zobrazení množiny A do množiny B se často definuje jako předpis, který každému prvku množiny A přiřazuje jediný prvek množiny B. Pomocí pojmu relace lze definovat zobrazení přesněji takto:

2.1. Definice. Buďte A, B mnnožiny a R relace z A do B. Relace R se nazývá *zobrazení* množiny A do B, jestliže jsou splněny následující dvě podmínky:

(a) Ke každému $a \in A$ existuje $b \in B$ tak, že aRb.

(b) Pro libovolné prvky $a \in A$ a $b, c \in B$ z podmínek aRb a aRc plyne $b = c$.



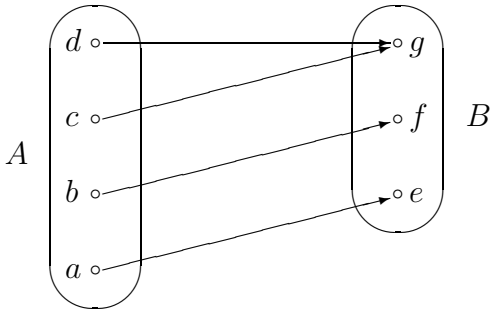
Jsou-li množiny A, B konečné, pak relace R z A do B je zobrazení, právě když každý řádek matice relace R obsahuje právě jednu jedničku, tj. právě když v grafu relace R vede z každého prvku $a \in A$ právě jedna orientovaná hrana.

Místo zobrazení se často používá také pojmu *funkce*. Abychom odlišili zobrazení od relací, budeme je označovat malými písmeny. Skutečnost, že f je zobrazení množiny A do B, se obvykle zapisuje ve tvaru $f : A \to B$. Místo afb se pak píše $b = f(a)$. V souladu s předchozí kapitolou f^{-1} značí inverzní relaci k zobrazení f a fg značí složení zobrazení, tj. relací f a g.

2.2. Příklady. (1) Buď U konečná množina a nechť $\mathcal{P}(U)$ značí množinu všech podmnožin množiny U. Buď $c : \mathcal{P}(U) \to \mathbb{Z}^+$ funkce definovaná vztahem $m = c(S)$ $\Leftrightarrow$ m je počet prvků množiny $S \subseteq U$. Pak číslo $m = c(S)$ se nazývá *mohutnost* množiny S a značí se card S.

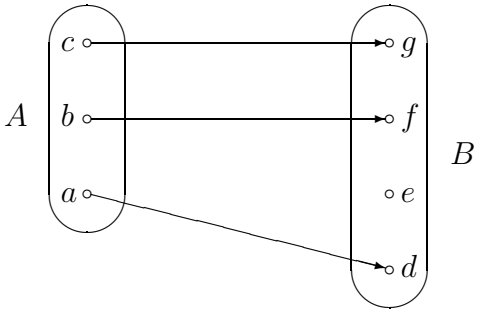
(2) Zobrazení $f : A \to \mathbb{R}$, kde $A \subseteq \mathbb{R}$, jsou předmětem studia matematické analýzy a nazývají se obvykle funkcemi. Např. funkce $f(x) = \sqrt{x}$ je zobrazením $f : \langle 0, \infty) \to \mathbb{R}$.

2.3. Definice. Zobrazení $f : A \to B$ se nazývá zobrazením A *na* B nebo *surjekcí*, jestliže ke každému $b \in B$ existuje $a \in A$ tak, že $b = f(a)$.



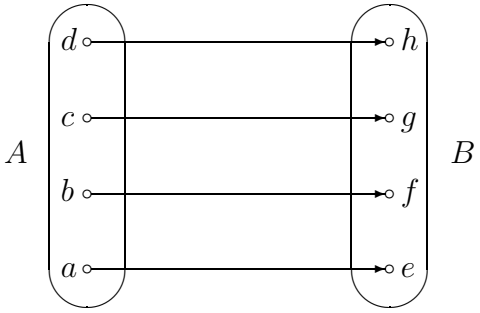
2.4. Příklad. Zřejmě funkce tg x je surjekcí množiny $\mathbb{R} - \{(2k+1)\frac{\pi}{2};\ k \in \mathbb{Z}\}$ na $\mathbb{R}$.

2.5. Definice Zobrazení $f : A \to B$ se nazývá *prosté* nebo *injekce*, jestliže pro libovolné $a_1, a_2 \in A$ z podmínky $f(a_1) = f(a_2)$ plyne $a_1 = a_2$.



2.6. Příklad. Zřejmě funkce arcsin x a arccos x jsou injekce intervalu $\langle -1, 1\rangle$ do $\mathbb{R}$, zatímco arctg x a arccotg x jsou injekce $\mathbb{R}$ do $\mathbb{R}$.

2.7. Definice. Zobrazení $f : A \to B$ se nazývá *bijekcí*, je-li surjekcí i injekcí.



2.8. Příklad. Buď $f : \mathbb{N} \to \mathbb{Z}$ zobrazení dané předpisem

$$f(n) = \begin{cases} \frac{1-n}{2}, \text{ jestliže n} \in \mathbb{N} \text{ je liché,} \\ \frac{n}{2}, \text{ jestliže n} \in \mathbb{N} \text{ je sudé.} \end{cases}$$

Pak je zřejmě f bijekce.

2.9. Definice. Zobrazení $f : A \to A$ se nazývá *identita* na A, jestliže $f(a) = a$ pro každé $a \in A$.

Zřejmě je tedy každá identita bijekcí. Identitu na množině A budeme označovat symbolem id_A.

2.10. Definice. Buď $f : A \to B$ zobrazení. Je-li inverzní relace f^{-1} k f zobrazením B do A, pak se f^{-1} nazývá *inverzním zobrazením* k f.

2.11. Věta. Buď $f : A \to B$ zobrazení. Pak jsou následující podmínky ekvivalentní:
(i) K zobrazení f existuje inverzní zobrazení,
(ii) $ff^{-1} = id_A$, $f^{-1}f = id_B$,
(iii) f je bijekce.

Důkaz. Předpokládejme nejprve, že k zobrazení f existuje inverzní zobrazení, tj., že inverzní relace f^{-1} k f je zobrazení B do A. Nechť $x_1, x_2 \in A$ jsou libovolné prvky. Pak platí $x_1 f f^{-1} x_2 \Leftrightarrow$ existuje $x \in A$ tak, že $x = f(x_1)$ a $x_2 = f^{-1}(x) \Leftrightarrow$ existuje $x \in A$ tak, že $x_1 = f^{-1}(x)$ a $x_2 = f^{-1}(x) \Leftrightarrow x_1 = x_2$ (protože f^{-1} je zobrazení). Tedy $ff^{-1} = id_A$. Analogicky se dokáže druhá rovnost podmínky (ii). Tím máme dokázánu implikaci (i)⇒(ii).
Nechť je splněna podmínka (ii) a nechť $x_1, x_2 \in A$ jsou prvky takové, že $f(x_1) = f(x_2)$. Položme $y = f(x_1)$. Protože $x_1 = ff^{-1}(x_1)$ a $x_2 = ff^{-1}(x_2)$, máme $x_1 f y$, $y f^{-1} x_1$, $x_2 f y$ a $y f^{-1} x_2$. Proto $x_2 = ff^{-1}(x_1)$, takže $x_1 = x_2$. Tedy f je injekce. Buď nyní $z \in B$ libovolný prvek. Podle (ii) platí $z = f^{-1}f(z)$, tedy existuje $x \in A$ tak, že $z = f(x)$ (a $z f^{-1} x$). Proto je f surjekce. Ukázali jsme, že f je bijekce. Tedy platí implikace (ii)⇒ (iii).
Předpokládejme nakonec, že f je bijekce. Ukážeme, že inverzní relace f^{-1} k f je zobrazení. Buď $z \in B$ opět libovolný prvek. Protože je f surjekce, existuje prvek $x \in A$ tak, že $z = f(x)$, tj. tak, že $z f^{-1} x$. Buďte nyní $x_1, x_2 \in A$ prvky takové, že $z f^{-1} x_1$ a $z f^{-1} x_2$. Pak platí $z = f(x_1)$ a $z = f(x_2)$, takže $x_1 = x_2$, neboť f je injekce. Tedy f^{-1} je zobrazení. Proto (iii)⇒ (i) a důkaz je hotov.

2.12. Příklad. Funkce $y = \ln x$ je bijekce intervalu $(0, \infty)$ na $\mathbb{R}$. K ní inverzní funkcí je zřejmě funkce $x = \mathrm{e}^y$.

2.13. Věta. Nechť R je relace z A do B a S je relace z B do C. Jsou-li R i S zobrazení, pak také jejich složení je zobrazení (A do C).

Důkaz. Nechť $x \in A$ je libovolný prvek. Protože je R zobrazení, existuje prvek $y \in B$ takový, že xRy. Protože je S zobrazení, existuje prvek $z \in C$ takový, že ySz. Tedy $xRSz$. Buďte nyní $z_1, z_2 \in C$ prvky takové, že $xRSz_1$ a $xRSz_2$. Pak existují prvky $y_1, y_2 \in B$ takové, že xRy_1, y_1Sz_1, xRy_2 a y_2Sz_2. Protože je R zobrazení, platí $y_1 = y_2$, a protože je S zobrazení, platí $z_1 = z_2$. Tedy je RS zobrazení a důkaz je hotov.

2.14. Poznámka. Složení fg zobrazení f a g se často zapisuje ve tvaru $g \circ f$. Platí tedy $c = fg(a) \Leftrightarrow c = g(f(a)) \Leftrightarrow c = g \circ f(a)$.

2.15. Příklad. Jestliže $f : (0, \infty) \to \mathbb{R}$ je funkce $y = \ln x$ a $g : \mathbb{R} \to \mathbb{R}$ je funkce $z = \sin y$, pak složená funkce $fg = g \circ f : (0, \infty) \to \mathbb{R}$ je funkce $z = \sin(\ln x)$.

2.16. Cvičení. a) Jestliže množina A má m prvků a množina B má n prvků, kolik prvků má množina všech zobrazení A do B?

b) Dokažte, že pro libovolné zobrazení $f : A \to B$ existuje množina C, injekce $h : C \to B$ a surjekce $g : A \to C$ tak, že $f = gh$.

c) Buďte $f : A \to B$ a $g : B \to C$ zobrazení. Dokažte, že platí:
(α) jsou-li f, g injekce (surjekce, bijekce), pak také fg je injekce (surjekce, bijekce),
(β) je-li fg je injekce, pak také f je injekce,
(γ) je-li fg surjekce, pak také g je surjekce.

3. Relace na množině

3.1. Definice. Buď A množina. Pak relace R z A do A (tj. $R \subseteq A \times A = A^2$) se nazývá relace na množině A.

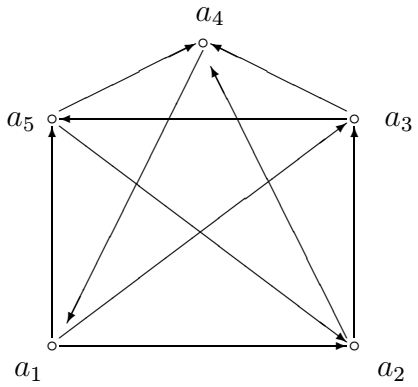
Zřejmě matice relace R na konečné množině A je čtvercová. Graf relace R na konečné množině $A = \{a_1, ...a_n\}$ obdržíme tak, že prvky množiny A znázorníme jako body a z bodu a_i vedeme orientovanou hranu do bodu a_j, právě když $a_i R a_j$. Tedy graf relace R na A je odlišný od grafu relace z A do A, který byl definován v 1. kapitole.

3.2. Příklady. (1) Buď P množina všech žijících lidí a pro libovolné $a, b \in P$ položme $aRb \Leftrightarrow a$ je rodičem b. Pak R je relace na P.

(2) Na tenisovém turnaji, kterého se zúčastnilo 5 hráčů, hrál každý s každým (právě jednou). Nechť $T = \{a_1, a_2, a_3, a_4, a_5\}$ je množina zúčastněných hráčů a definujme relaci R na T vztahem $a_i R a_j \Leftrightarrow a_i$ porazil a_j. Nechť $R = \{(a_1, a_2), (a_1, a_3), (a_1, a_5), (a_2, a_3), (a_2, a_4), (a_3, a_4), (a_3, a_5), (a_4, a_1), (a_5, a_2), (a_5, a_4)\}$. Pak maticí relace R je

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

a její graf má tvar



(3) Nechť $A = \{2, 3, 7, 9, 14\}$ a pro libovolné prvky $a, b \in A$ položme $aRb \Leftrightarrow a, b$ jsou soudělná čísla (tj. existuje společný dělitel čísel a, b, který je větší než 1). Pak R je relace na A, pro kterou platí $R = \{(2, 2), (2, 14), (3, 3), (3, 9), (7, 7), (7, 14), (9, 9), (9, 3), (14, 14), (14, 2), (14, 7)\}$.

Připomeňme, že identita na dané množině A je zobrazení $id_A : A \to A$ dané vztahem $id_A(a) = a$ pro všechna $a \in A$. Identitu na A budeme jako relaci označovat symbolem E_A, takže E_A je relace na A daná vztahem $aE_Ab \Leftrightarrow a = b$ (pro libovolné $a, b \in A$). Protože bude z kontextu vždy zřejmé, na jaké množině identitu uvažujeme, budeme pro ni používat stručnější označení E.

Matice identity je zřejmě jednotková matice (tj. čtvercová matice, jejíž prvky na hlavní diagonále jsou rovny jedné a všechny ostatní prvky jsou nulové).

Uvědomme si také, že k libovolným dvěma relacím na dané množině je definováno jejich složení.

3.3. Definice. Buď R relace na A. Pak klademe

$$R^n = \begin{cases} R \text{ pro } n = 1, \\ R^{n-1}R \text{ pro libovolné } n \in \mathbb{N},\ n > 1. \end{cases}$$

Tím máme rekurentně definovánu relaci R^n na A, tzv. $n-tou\ mocninu$ relace R, pro libovolné $n \in \mathbb{N}$. Definici této mocniny rozšíříme na celou množinu $\mathbb{Z}$ následovně:

$$R^n = \begin{cases} E \text{ pro } n = 0, \\ (R^{-n})^{-1} \text{ pro libovolné } n \in \mathbb{Z},\ n < 0. \end{cases}$$

Je-li M matice relace R, pak matici relace R^n pro libovolné $n \in \mathbb{N}$ dostaneme z matice M^n nahrazením všech prvků větších než 1 jedničkami. Matice relace R^{-n} je pak transponovaná k matici M^n.

3.4. Cvičení. a) Dokažte, že pro libovolnou relaci R na A platí :
(i) $RE = ER = R$,
(ii) $R^mR^n = R^{m+n}$ pro libovolné $m, n \in \mathbb{N}$,
(iii) $(R^m)^n = R^{mn}$ pro libovolné $m, n \in \mathbb{Z}$.

b) Dále dokažte, že pro libovolné relace $R,\ S$ na A a libovolné $n \in \mathbb{Z}$ z podmínky $R \subseteq S$ plyne $R^n \subseteq S^n$.

3.5. Příklad. V příkladu 3.2(1) platí $aR^2b \Leftrightarrow a$ je prarodič b, $aR^3b \Leftrightarrow a$ je praprarodič b, atd. Dále platí $aR^{-1}b \Leftrightarrow a$ je syn nebo dcera B, $aR^{-2}b \Leftrightarrow a$ je vnuk nebo vnučka b, atd.

3.6. Definice. Relace R na množině A se nazývá
a) *reflexivní*, jestliže $E \subseteq R$,
b) *ireflexivní*, jestliže $E \cap R = \emptyset$,
c) *symetrická*, jestliže $R^{-1} \subseteq R$,
d) *asymetrická*, jestliže $R \cap R^{-1} = \emptyset$,
e) *antisymetrická*, jestliže $R \cap R^{-1} \subseteq E$,

f) *tranzitivní*, jestliže $R^2 \subseteq R$.

K bodu c) předchozí definice poznamenejme, že zřejmě platí $R^{-1} \subseteq R \Leftrightarrow R^{-1} = R$, a k bodu e), že platí $R^2 \subseteq R \Leftrightarrow R^n \subseteq R$ pro libovolné $n \in \mathbb{N}$.

3.7. Poznámka. Relace R na A je tedy
a) reflexivní, právě když aRa pro libovolné $a \in A$,
b) ireflexivní, právě když $a\bar{R}a$ pro libovolné $a \in A$,
c) symetrická, právě když $aRb \Rightarrow bRa$ pro libovolné $a, b \in A$,
d) asymetrická, právě když $aRb \Rightarrow b\bar{R}a$ pro libovolné $a, b \in A$,
e) antisymetrická, právě když z aRb a bRa plyne $a = b$,
f) tranzitivní, právě když z aRb a bRc plyne aRc.

Buď R relace na konečné množině o n prvcích a $M = (a_{ij})$, $i, j = 1, ..., n$, její matice. Pak platí:
a) R je reflexivní, právě když $a_{ii} = 1$ pro všechna $i = 1, ..., n$,
b) R je ireflexivní, právě kdy $a_{ii} = 0$ pro všechna $i = 1, ..., n$,
c) R je symetrická, právě když matice M je symetrická (tj. $a_{ij} = a_{ji}$ pro všechna $i, j = 1, ..., n$),
d) R je asymetrická, právě když $a_{ij} + a_{ji} \leq 1$ pro všechna $i, j = 1, ..., n$.

Graf reflexivní relace má smyčky u všech svých prvků (smyčkou rozumíme hranu jdoucí z prvku do téhož prvku), graf symetrické relace má s každou hranou i hranu opačně orientovanou a v grafu antisymetrické relace vede mezi libovolnými dvěma různými prvky nejvýše jedna hrana.

3.8. Poznámka. Různé druhy grafů jsou předmětem studia *teorie grafů*, která tvoří významnou součást diskrétní matematiky s bohatými aplikacemi. Přitom základními typy grafů jsou tzv. *orientované grafy* a *neorientované grafy*. Orientované grafy nejsou nic jiného než grafy relací na množinách a neorientované grafy lze chápat jako grafy symetrických relací na množinách (přičemž v obou případech se prvky uvažovaných množin nazývají *uzly*).

3.9. Příklady (1) Relace R z příkladu 3.2(2) je ireflexivní a asymetrická.
(2) Relace R z příkladu 3.2(3) je reflexivní a symetrická.
(3) Buď R relace na $\mathbb{Z}$ daná vztahem $aRb \Leftrightarrow a = b^2$. Pak R je antisymetrická: Nechť aRb a bRa. Pak $a = b^2$ a $b = a^2$, taže $a = a^4$. Odtud plyne, že $a = 0$, tedy i $b = 0$, nebo $a = 1$, tedy i $b = 1$. Proto $a = b$.
(4) Buď P množina všech žijících lidí a definujme relaci R na P takto: $aRb \Leftrightarrow b$ je potomkem a. Pak relace R je tranzitivní.

3.10. Cvičení. Buďte R, S relace ne množině A. Dokažte, že platí:
a) Je-li R asymetrická, pak je také ireflexivní.

b) Jsou-li R, S reflexivní, pak jsou také $R \cup S$, $R \cap S$, R^{-1} a RS reflexivní.

c) Jsou-li R, S symetrické (ireflexivní), pak jsou také $R \cup S$, $R \cap S$ a R^{-1} symetrické (ireflexivní).

d) Jsou-li R, S asymetrické (antisymetrické, tranzitivní), pak jsou také $R \cap S$ a R^{-1} asymetrické (antisymetrické, tranzitivní).

3.11. Věta. Buďte R, S symetrické relace na množině A. Pak relace RS je symetrická, právě když platí $RS = SR$.

Důkaz. Předpokládejme nejprve, že relace RS je symetrická. Nechť $a, c \in A$ jsou libovolné prvky. Pak platí $aRSc \Leftrightarrow cRSa \Leftrightarrow$ existuje $b \in A$ tak, že cRb a $bSa \Leftrightarrow$ existuje prvek $b \in A$ tak, že aSb a $bRc \Leftrightarrow aSRc$. Tedy $RS = SR$.

Naopak, nechť $RS = SR$. Pak máme $(RS)^{-1} = S^{-1}R^{-1} = SR = RS$. Takže relace RS je symetrická.

Jestliže relace R není reflexivní, vždy ji lze doplnit jistými prvky (dvojicemi) tak, aby reflexivní byla. Snahou přitom je, abychom doplňovali co nejméně těchto prvků. Totéž platí pro symetrii a tranzitivitu. Proto definujeme:

3.12. Definice. Buď R relace na množině A. Jejím *reflexivním (symetrickým, tranzitivním) obalem* rozumíme reflexivní (symetrickou, tranzitivní) relaci S na A takovou, že $R \subseteq S$ a pro každou reflexivní (symetrickou, tranzitivní) relaci T na A s vlastností $R \subseteq T$ platí $S \subseteq T$.

3.13. Cvičení. Buď R relace na množině A. Ukažte, že platí:

a) Reflexivním obalem relace R je relace $R \cup E$.

b) Symetrickým obalem relace R je relace $R \cup R^{-1}$.

Matici reflexivního obalu relace R (na konečné množině) obdržíme z matice relace R tak, že všechny prvky ležící na hlavní diagonále nahradíme jedničkami; její graf získáme z grafu relace R doplněním smyček ke všem prvkům množiny A. Matici symetrického obalu relace R obdržíme z matice relace R tak, že pro každý prvek a_{ij}, který je roven 1, napíšeme jedničku také na místo prvku a_{ji}; její graf získáme z grafu relace R tak, že ke každé hraně přidáme hranu opačně orientovanou.

3.14. Příklady. (1) Buď X množina a $\mathcal{P}(X)$ množina všech jejích podmnožin. Pak relace vlastní inkluze $\subset$ je relace na $\mathcal{P}(X)$ a relace inkluze $\subseteq$ je její reflexivní obal.

(2) Buď P množina všech (žijících) lidí a definujme relaci R na P vztahem $aRb \Leftrightarrow a$ je manželem b. Pak symetrickým obalem relace R je relace S na P daná vztahem $aSb \Leftrightarrow a, b$ jsou manželé.

3.15. Věta. Tranzitivním obalem relace R na množině A je relace $\widehat{R} = \bigcup_{n=1}^{\infty} R^n$.

Důkaz. Zřejmě platí $R \subseteq \widehat{R}$. Budťe $x, y, z \in A$ prvky takové, že $x\widehat{R}y$ a $y\widehat{R}z$. Pak existují čísla $m, n \in \mathbb{N}$ tak, že $xR^m y$ a $yR^n z$, tedy $xR^{m+n}z$. Proto je $\widehat{R}$ tranzitivní. Buď nyní S libovolná tranzitivní relace na A taková, že $R \subseteq S$. Pak pro libovolné $n \in N$ máme $R^n \subseteq S^n$ (viz cvičení 3.4b)) a $S^n \subseteq S$ (protože je S tranzitivní). Takže $R^n \subseteq S$ pro libovolné $n \in \mathbb{N}$, tj. $\widehat{R} \subseteq S$. Tím je věta dokázána.

3.16. Příklad Relace R z příkladu 3.9(4) je tranzitivní obal relace R z příkladu 3.2(1).

3.17. Poznámka. Pro libovolné přirozené číslo n se definuje tzv. *n-ární relace* na množině A jako libovolná podmnožina kartézského součinu $A^n = \underbrace{A \times A \times ... \times A}_{n-\text{krát}}$. Místo 1-ární (2-ární, 3-ární atd.) relace říkáme *unární* (*binární*, *ternární* atd.) relace. Tedy unární relace na množině A jsou právě podnmožiny množiny A. Binární relace na A splývají s relacemi na A studovanými v této kapitole. Také n-ární relace mají bohaté aplikace v nejrůznějších oborech, např. v informatice jsou užívány pro tvorbu tzv. relačních databází.

4. Tolerance a ekvivalence

4.1. Definice. Reflexivní a symetrická relace (na dané množině) se nazývá *tolerance*.

V grafu tolerance R na A vynecháváme smyčky (které bychom jinak museli kreslit u všech prvků množiny A) a prvky $a, b \in A$ spojujeme neorientovanou hranou, právě když aRb (jinak bychom s každou hranou museli kreslit i hranu opačně orientovanou). Tím dostáváme tzv. neorientovaný graf - viz poznámku 3.8.

4.2. Příklady. (1) Buď $X \neq \emptyset$ množina a $\mathcal{P}^+(X)$ množina všech jejích neprázdných podmnožin. Jestliže pro libovolné množiny $A, B \in \mathcal{P}^+(X)$ položíme $ARB \Leftrightarrow A \cap B \neq \emptyset$, pak R je tolerance na $\mathcal{P}^+(X)$.

(2) Buď G množina cestujících v letadle na mezikontinentální lince. Pro libovolné $a, b \in G$ položme $aRb \Leftrightarrow$ osoby $a,\ b$ ovládají tentýž jazyk. Pak R je tolerance na G.

(3) Buď F_D množina všech (reálných) funkcí definovaných na množině $D \subseteq \mathbb{R}$ a nechť $\varepsilon > 0$ je libovolné reálné číslo. Pro libovolné dvě funkce $f, g \in F_D$ položme $fRg \Leftrightarrow |f(x) - g(x)| < \varepsilon$ pro každé $x \in D$. Pak R je tolerance na F_D.

4.3. Definice. Buď R tolerance na množině A. Neprázdná podmnožina $C \subseteq A$ se nazývá *třída* tolerance R, jestliže jsou splněny následující dvě podmínky:

(i) Pro libovolné prvky $x, y \in C$ platí xRy.

(ii) Jestliže $x \in A$ je prvek takový, že xRy pro každé $y \in C$, pak $x \in C$.

4.4. Příklady. (1) Buď R relace z příkladu 4.2(1) a nechť $x \in X$ je libovolný prvek. Pak množina $\{A \in \mathcal{P}(X);\ x \in A\}$ je třída tolerance R.

(2) V příkladu 4.2(2) předpokládejme, že existuje cestující, který ovládá jen angličtinu. Pak množina $\{a \in G;\ a$ ovládá angličtinu$\}$ je třída tolerance R.

(3) Buď R relace z příkladu 4.2(3) a nechť $f \in F_D$ je libovolná funkce. Pak množina $\{g \in F_D;\ f(x) \leq g(x) < f(x) + \varepsilon$ pro všechna $x \in D\}$ je třída tolerance R.

4.5. Lemma. Nechť R je tolerance na množině A a $x, y \in A$ jsou libovolné prvky s vlastností xRy. Pak existuje třída C tolerance R tak, že $x, y \in C$.

Důkaz. Důkaz provedeme jen pro konečnou množinu $A = \{a_1, ..., a_n\}$. Buď $C_0 \subset C_1 \subset ... \subset C_k$ konečná posloupnost podmnožin množiny A definovaná takto: $C_0 = \{x, y\}$; je-li množina C_i známa pro nějaké $i \in \{0, 1, ..., k-1\}$, pak $C_{i+1} = C_i \cup \{a_j\}$, kde $j \in \{1, ..., n\}$ je nejmenší číslo takové, že $a_j \notin C_i$ a $a_j Rz$ pro všechna $z \in C_i$. Jestliže takové j neexistuje, pak C_i je poslední množinou v naší posloupnosti, tj. $C_i = C_k$. Množina C_k je zřejmě třídou tolerance R takovou, že $x, y \in C$.

4.6. Důsledek. Tolerance R na množině A je jednoznačně určena systémem všech svých tříd.

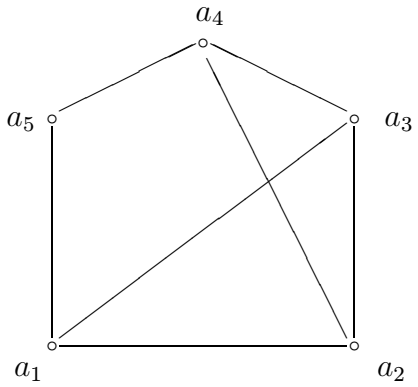
Důkaz. Z definice 4.3 a z lemmatu 4.5 plyne, že pro libovolné prvky $x, y \in A$ platí $xRy \Leftrightarrow$ existuje třída C tolerance R taková, že $x, y \in C$.

4.7. Definice. Buď A množina, $\mathcal{A} = \{A_i;\ i \in I\}$ nějaký systém jejích neprázdných podmnožin. Pak $\mathcal{A}$ se nazývá *pokrytí* množiny A, jestliže $A = \bigcup_{i \in I} A_i$. Jestliže navíc platí $A_i \cap A_j = \emptyset$ pro libovolné $i, j \in I$, $i \neq j$, pak se $\mathcal{A}$ nazývá *rozklad* množiny A.

4.8. Věta. Systém všech tříd libovolné tolerance na množině A je pokrytí A. Naopak, jestliže $\mathcal{A}$ je pokrytí množiny A, pak existuje nejvýše jedna tolerance na množině A taková, že $\mathcal{A}$ je systém všech jejích tříd.

Důkaz. Buď R tolerance na A a $x \in A$ libovolný prvek. Pak xRx, tedy podle lemmatu 4.5 existuje třída C tolerance R s vlastností $x \in C$. Takže systém všech tříd tolerance R tvoří pokrytí množiny A.
Naopak, nechť $\mathcal{A} = \{A_i;\ i \in I\}$ je pokrytí množiny A a připusťme, že existují dvě různé tolerance R a S na A takové, že $\mathcal{A}$ je systém všech tříd tolerance R i S. Protože tolerance R a S jsou různé, existuje prvek $(x, y) \in A^2$ patřící pouze do jedné z nich. Bez újmy na obecnosti můžeme předpokládat, že $(x, y) \in R$, avšak $(x, y) \notin S$. Podle lemmatu 4.5 z $(x, y) \in R$ vyplývá, že existuje $i_0 \in I$ tak, že $x, y \in A_{i_0}$. Na druhé straně ovšem $\{x, y\} \not\subseteq A_i$ pro každé $i \in I$, jelikož $(x, y) \notin S$. Tím dostáváme spor.

4.9. Příklad. Nechť $A = \{a_1, a_2, a_3, a_4, a_5\}$ a $\mathcal{A} = \{\{a_1, a_5\}, \{a_4, a_5\}, \{a_1, a_2, a_3\}, \{a_2, a_3, a_4\}\}$. Pak $\mathcal{A}$ je pokrytí množiny A, které je systémem tříd tolerance R, jejíž graf má následující tvar:



4.10. Věta. Relace T na množině A je tolerance, právě když existuje relace R z A do nějaké množiny B tak, že jsou splněny následující dvě podmínky:
(i) Pro každý prvek $x \in A$ existuje $y \in B$ tak, že xRy,
(ii) $T = RR^{-1}$.

Důkaz. Buď T tolerance a nechť $\mathcal{A}$ je systém jejích tříd. Pro libovolný prvek $x \in A$ a libovolnou množinu $C \in \mathcal{A}$ položme $xRC \Leftrightarrow x \in C$. Podle věty 4.8 existuje pro libovolný prvek $x \in A$ množina $C \in \mathcal{A}$ s vlastností $x \in C$, tj. s vlastností xRC. Tedy je splněna podmínka (i). Dále, nechť $x, y \in A$ jsou libovolné prvky. Pak platí $xTy \Leftrightarrow$ existuje $C \in \mathcal{A}$ tak, že $x, y \in C \Leftrightarrow xRC$ a $yRC \Leftrightarrow xRC$ a $CR^{-1}y \Leftrightarrow xRR^{-1}y$. Proto $T = RR^{-1}$, což je podmínka (ii).
Naopak, nechť R je relace z A do nějaké množiny B taková, že jsou splněny podmínky (i) a (ii). Buď $x \in A$ libovolný prvek. Pak existuje $y \in B$ tak, že xRy, tj. $yR^{-1}x$. Máme tedy $xRR^{-1}x$, takže xTx. Proto je T reflexivní. Dále, buďte $x, y \in A$ libovolné prvky a nechť platí xTy. Pak existuje $z \in B$ tak, že xRz a $zR^{-1}y$. Odtud dostáváme $zR^{-1}x$ a yRz, tudíž $yRR^{-1}x$, tj. yTx. Proto je T symetrická. Takže T je tolerance.

4.11. Definice. Tranzitivní tolerance se nazývá ekvivalence.

4.12. Příklad. Buď $n \in \mathbb{N}$ a nechť $\equiv$ je relace na $\mathbb{Z}$ daná vztahem $x \equiv y \Leftrightarrow$ existuje $k \in \mathbb{Z}$ tak, že $kn = (x - y)$ (tj. n dělí $(x - y)$). Pak $\equiv$ je ekvivalence na $\mathbb{Z}$, která se nazývá *kongruence modulo* n.

4.13. Věta. Buď R tolerance na množině A a $\mathcal{A} = \{C_i;\ i \in I\}$ systém jejích tříd. Pak R je ekvivalence, právě když třídy C_i, $i \in I$, jsou po dvou disjunktní (tj. $C_i \cap C_j = \emptyset$ pro libovolné $i, j \in I$, $i \neq j$).

Důkaz. Buď R ekvivalence a nechť $C_i, C_j \in \mathcal{A}$, $C_i \cap C_j \neq \emptyset$. Pak existuje prvek $x \in C_i \cap C_j$. Buď $y \in C_i$ libovolný prvek. Pak platí xRy, tedy yRx. Ovšem xRz platí pro každý prvek $z \in C_j$, takže máme yR^2z pro každý prvek $z \in C_j$. Odtud plyne (podle 4.3(ii)), že $y \in C_j$. Tím jsme dokázali, že $C_i \subseteq C_j$. Analogicky se dokáže opačná inkluze, takže $C_i = C_j$. Tedy $i = j$ a odtud plyne, že množiny C_i, $i \in I$, jsou po dvou disjunktní.
Naopak, nechť množiny C_i, $i \in I$, jsou po dvou disjunktní a nechť $x, y \in A$, xR^2y. Pak existuje $z \in A$ tak, že xRz a zRy. Buď C_i třída tolerance R obsahující prvky x, z a C_j třída tolerance R obsahující prvky z, y. Pak musí platit $C_i = C_j$, takže xRy. Proto $R^2 \subseteq R$, tj. R je tranzitivní. Tedy R je ekvivalence.

4.14. Důsledek. Buď A množina. Je-li R ekvivalence na A, pak systém jejích tříd je rozklad množint A. Naopak, ke každému rozkladu množiny A existuje (jediná) ekvivalence, pro niž je tento rozklad systémem tříd.

Důkaz. První část tvrzení plyne ihned z vět 4.8 a 4.13. Buď $\mathcal{A} = \{A_i;\ i \in I\}$ libovolný rozklad množiny A a pro libovolné prvky $x, y \in A$ položme $xRy \Leftrightarrow$ existuje $i_0 \in I$ tak, že $x, y \in A_{i_0}$. Je snadno vidět, že R je ekvivalence na A a $\mathcal{A}$ je systém všech jejích tříd. Podle věty 4.8 je R jediná ekvivalence, pro niž je $\mathcal{A}$ systémem všech tříd.

4.15. Cvičení. Dokažte, že podmnožina $C \subseteq A$ je třídou dané ekvivalence R na A, právě když existuje prvek $x \in A$ tak, že $C = \{y \in A; xRy\}$. Pak zřejmě $x \in C$ a prvek x se nazývá *reprezentant* třídy C.

4.16. Věta. Relace S na množině A je ekvivalencí, právě když existuje zobrazení R množiny A do nějaké množiny B tak, že platí $S = RR^{-1}$.

Důkaz. Stačí dokázat, že ve větě 4.10 je tolerance T na A ekvivalencí, právě když pro libovolné prvky $x \in A$ a $y, z \in B$ z podmínek xRy a xRz plyne $y = z$. Pokračujme tedy v první části důkazu věty 4.10 tak, že předpokládáme, že T je ekvivalence, a uvažujeme prvky $x \in A$, $C_1, C_2 \in \mathcal{A}$ takové, že xRC_1 a xRC_2. Odtud dostáváme $x \in C_1$ a $x \in C_2$, takže $C_1 = C_2$ (neboť třídy ekvivalence T jsou podle věty 4.14 po dvou disjunktní). Dále pokračujme v druhé části důkazu věty 4.10 tak, že budeme naopak předpokládat, že pro libovolné prvky $x \in A$ a $y, z \in B$ z podmínek xRy a xRz plyne $y = z$. Buďte $u, v, w \in A$ prvky takové, že platí uTv a vTw. Pak existují prvky $p, q \in B$ s vlastností uRp, $pR^{-1}v$ (tj. vRp), vRq a $qR^{-1}w$. Tedy máme $p = q$, takže uRp a $pR^{-1}w$. Odtud vyplývá, že $uRR^{-1}w$, tj. uTw. Proto je tolerance T tranzitivní, tedy ekvivalence. Tím je důkaz hotov.

5. Uspořádané množiny

5.1. Definice. Reflexivní, antisymetrická a tranzitivní relace R na množině A se nazývá *uspořádání* a dvojice (A, R) se nazývá *uspořádaná množina*.

5.2. Příklady. (1) Relace $\leq$ je uspořádání na $\mathbb{N}$ (resp. $\mathbb{Z}$, resp. $\mathbb{Q}$, resp. $\mathbb{R}$).

(2) Buď A množina. Pak množinová inkluze $\subseteq$ je uspořádání na množině $\mathcal{P}(A)$ všech podmnožin množiny A.

(3) Relace "dělí", kterou budeme označovat symbolem $|$, je uspořádání na $\mathbb{N}$.

5.3. Definice. Ireflexivní a tranzitivní relace R na množině A se nazývá *ostré uspořádání* a dvojice (A, R) se nazývá *ostře uspořádaná množina*.

5.4. Příklady. (1) Relace $<$ je ostré uspořádání na $\mathbb{N}$ (resp. $\mathbb{Z}$, resp. $\mathbb{Q}$, resp. $\mathbb{R}$).

(2) Buď A množina. Pak vlastní množinová inkluze $\subset$ je ostré uspořádání na množině $\mathcal{P}(A)$ všech podmnožin množiny A.

(3) Buď D množina vojenských hodností v armádě nějakého státu a pro libovolné $a, b \in D$ položme $aRb \Leftrightarrow$ hodnost a je nižší než hodnost b. Pak R je ostré uspořádání na D.

Zřejmě platí (viz cvičení 3.10):

5.5. Věta. Inverzní relace k uspořádání (ostrému uspořádání) je uspořádání (ostré uspořádání).

Relaci uspořádání, resp. ostrého uspořádání na dané množině budeme obvykle značit symbolem $\leq$, resp.$<$, i když se nebude jednat o číselnou množinu a uspořádání, resp. ostré uspořádání podle velikosti. Inverzní uspořádání, resp. inverzní ostré uspořádání k danému uspořádání $\leq$, resp. ostrému uspořádání $<$ budeme značit symbolem $\geq$, resp $>$. Jinými slovy, klademe $\leq^{-1} = \geq$ a $<^{-1} = >$.

5.6. Věta. Relace R je ostré uspořádání, právě když R je asymetrická a tranzitivní.

Důkaz. Nechť R je ostré uspořádání na A. Pak relace R je tranzitivní a ukážeme sporem, že R je také asymetrická. Předpokládejme tedy, že relace R není asymetrická. Pak existují prvky $a, b \in A$ tak, že aRb a bRa. Z tranzitivity relace R nyní plyne aRa. To je ovšem spor, neboť R je ireflexivní.
Je-li naopak R je asymetrická a tranzitivní, pak je R ostré uspořádání, neboť asymetrie vždy implikuje ireflexivitu.

Následující tvrzení je evidentní:

5.7. Věta. Buď A množina. Pak existuje bijekce f mezi množinami všech uspořádání a všech ostrých uspořádání na A, která je dána vztahem $f(\leq) =\leq -E$ (takže pro inverzní bijekci platí $f^{-1}(<) =< \cup E$).

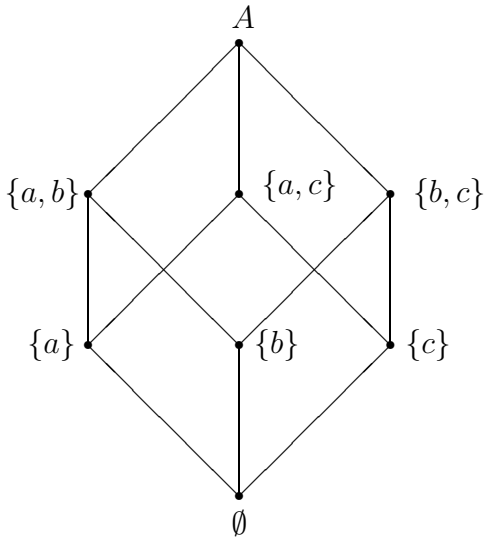
V souladu s předchozí větou užíváme následující značení: Je-li $\leq$ uspořádání na A, pak píšeme $a < b$, právě když $a \leq b$ a $a \neq b$. Je-li $<$ ostré uspořádání na A, pak píšeme $a \leq b$, právě když $a < b$ nebo $a = b$.

5.8. Definice. Buď $(A, \leq)$ uspořádaná množina, $a, b \in A$ nějaké její prvky. Pak říkáme, že prvek b *pokrývá* prvek a, právě když $a < b$ a neexistuje prvek $c \in A$ takový, že $a < c$ a $c < b$.

Zřejmě "pokrývá" je asymetrická (a tedy ireflexivní) relace na A.

Pro grafickou reprezentaci uspořádaných množin $(A, \leq)$ se užívají tzv. Hasseovy diagramy: Prvky množiny A znázorníme jako body, přičemž prvek a umístíme níž než prvek b a oba prvky spojíme úsečkou, právě když prvek b pokrývá prvek a. Pro libovolné dva prvky $x, y \in A$ pak platí $x < y$, právě když a je níž než b a z a do b je možno se dostat po úsečkách směrem vzhůru.

5.9. Příklad. Nechť $A = \{a, b, c\}$ a nechť $\mathcal{P}(A)$ je množina všech podmnožin množiny A. Pak uspořádaná množina $(\mathcal{P}(A), \subseteq)$ má následující Hasseův diagram:



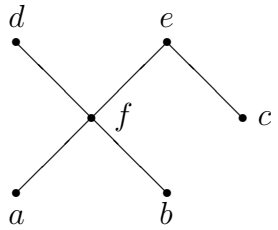
5.10. Definice. Uspořádání $\leq$ na množině A se nazývá *lineární* a dvojice $(A, \leq)$ se nazývá *lineárně uspořádaná množina* (nebo *řetězec*), jestliže pro libovolné dva prvky $a, b \in A$ platí $a \leq b$ nebo $b \leq a$.

5.11. Příklady. Uspořádání z příklad 5.2(1) jsou lineární.

5.12. Definice. Buď $(A, \leq)$ uspořádaná množina. Prvek $a \in A$ se nazývá
(1) *nejmenší (největší)* prvek uspořádané množiny $(A, \leq)$, jestliže platí $a \leq x$ $(a \geq x)$ pro každý prvek $x \in A$;
(2) *minimální (maximální)* prvek uspořádané množiny $(A, \leq)$, jestliže pro libovolný prvek $x \in A$ z podmínky $x \leq a$ $(x \geq a)$ plyne $x = a$.

Zřejmě je každý nejmenší (největší) prvek uspořádané množiny jejím minimálním (maximálním) prvkem a pro lineárně uspořádané množiny platí toto tvrzení i obráceně. Každá konečná uspořádaná množina má alespoň jeden minimální a alespoň jeden maximální prvek; má-li jediný minimální (maximální) prvek, pak je tento prvek nejmenším (největším).

5.13. Příklady. (1) Je-li A množina a $\mathcal{P}(A)$ množina jejích podmnožin, pak $\emptyset$ je nejmenší a A největší prvek uspořádané množiny $(\mathcal{P}(A), \subseteq)$.
(2) V uspořádané množině s následujícím Hasseovým diagramem



jsou prvky a, b, c minimální a prvky d, e maximální. Prvek f není minimální ani maximální.

5.14. Definice. Je-li $(A, \leq)$ uspořádaná množina a $B \subseteq A$ libovolná podmnožina, pak také množina B je uspořádaná, a to relací $\leq \cap B^2$, kterou obvykle také značíme symbolem $\leq$. Dvojice $(B, \leq)$ se pak nazývá *uspořádaná podmnožina* uspořádané množiny $(A, \leq)$.

5.15. Příklad. (1) $(\mathbb{N}, \leq)$, resp. $(\mathbb{Z}, \leq)$, resp. $(\mathbb{Q}, \leq)$ je uspořádaná podmnožina uspořádané množiny $(\mathbb{R}, \leq)$.
(2) Buďte A, B množiny, $A \subseteq B$. Nechť $\mathcal{P}(A)$, resp. $\mathcal{P}(B)$ je množina všech podmnožin množiny A, resp. B. Pak $(\mathcal{P}(A), \subseteq)$ je uspořádaná podmnožina uspořádané množiny $(\mathcal{P}(B), \subseteq)$.

5.16. Cvičení. Nechť $(A, \leq)$ je uspořádaná podmnožina uspořádané množiny $(B, \leq)$ a nechť $a \in A$. Dokažte, že platí:
Jestliže a je nejmenším (největším, minimálním, maximálním) prvkem uspořádané množiny $(B, \leq)$, pak je také nejmenším (největším, minimálním, maximálním) prvkem uspořádané množiny $(A, \leq)$.

5.17. Definice. Buď $(A, \leq)$ uspořádaná množina, $x, y \in A$. Prvky $x, y \in A$ se nazývají *srovnatelné*, jestliže platí $x \leq y$ nebo $y \leq x$. Pokud neplatí ani $x \leq y$ ani $y \leq x$, pak se prvky x, y nazývají *nesrovnatelné* a píšeme $x \parallel y$.

Kapitolu zakončíme následující významnou větou:

5.18. Věta. Buď $\leq$ uspořádání na množině A. Pak existuje lineární uspořádání $\preceq$ na A takové, že platí inkluze $\leq \subseteq \preceq$ (tj. každé uspořádání lze linearizovat).

Důkaz. Důkaz provedeme jen pro případ konečné množiny $A = \{a_1, a_2, ..., a_n\}$, a to konstruktivně, což znamená, že popíšeme algoritmus pro konstrukci lineárního uspořádání $\preceq$. Pro každé $i \in \{1, ..., n\}$ definujme rekurentně lineární uspořádání $\preceq_i$ na $A_i = \{a_1, ..., a_i\}$ tak, že položíme $\preceq_1 = \leq \cap A_1^2 (= E_{A_1})$ a je-li $\preceq_i$ definováno pro nějaké $i \in \{1, ..., n-1\}$, přičemž $a_{k_1} \preceq_i a_{k_2} \preceq_i ... \preceq_i a_{k_i}$ ($\{k_1, ..., k_i\} = \{1, ..., i\}$), pak $\preceq_{i+1}$ obdržíme následovně: Jestliže prvek a_{i+1} není srovnatelný s žádným prvkem množiny A_i (v uspořádané množině $(A, \leq)$), pak pro libovolné $x, y \in A_{i+1}$ položíme $x \preceq_{i+1} y \Leftrightarrow x \preceq_i y$ nebo $y = a_{i+1}$. Naopak, nechť existuje prvek $z_0 \in \{a_1, ..., a_i\}$ srovnatelný s a_{i+1} v uspořádané množině $(A, \leq)$. To znamená, že $z_0 < a_{i+1}$ nebo $a_{i+1} < z_0$.
Předpokládejme nejprve, že $z_0 < a_{i+1}$, a položme $B_i = \{z \in A_i;\ z < a_{i+1}\}$. Pak $(B_i, \preceq_i)$ je konečný neprázdný řetězec, tedy má největší prvek, řekněme a_{k_j}. Pro libovolné $x, y \in A_{i+1}$ položíme $x \preceq_{i+1} y$ tehdy a jen tehdy, jestliže platí jedna z následujících čtyř podmínek:
(a) $x \preceq_i y$,
(b) $x \preceq_i a_{k_j}$ a $y = a_{i+1}$,
(c) $x = a_{i+1}$ a $a_{k_{j+1}} \preceq_i y$,
(d) $x = y = a_{i+1}$.
Konečně předpokládejme, že $a_{i+1} < z_0$, a položme $C_i = \{z \in A_i;\ a_{i+1} < z\}$. Pak $(C_i, \preceq_i)$ je konečný neprázdný řetězec, tedy má nejmenší prvek, řekněme a_{k_l}. Pro libovolné $x, y \in A_{i+1}$ položíme $x \preceq_{i+1} y$ tehdy a jen tehdy, jestliže platí jedna z následujících čtyř podmínek:
(a') $x \preceq_i y$,
(b') $x = a_{i+1}$ a $a_{k_l} \preceq_i y$,
(c') $x \preceq_i a_{k_{l-1}}$ a $y = a_{i+1}$,
(d') $x = y = a_{i+1}$.
Zřejmě $\leq \cap A_1^2 \subseteq \preceq_1$ a platí-li $\leq \cap A_i^2 \subseteq \preceq_i$ pro nějaké $i \in \{1, ..., n-1\}$, pak také $\leq \cap A_{i+1}^2 \subseteq \preceq_{i+1}$. Tedy podle principu matematické indukce máme $\leq \cap A_i^2 \subseteq \preceq_i$ pro každé $i \in \{1, ..., n\}$. Nyní jen položíme $\preceq = \preceq_n$ a $\preceq$ je pak lineární uspořádání na A takové, že platí $\leq \subseteq \preceq$, neboť $\leq = \leq \cap A^n$ (jelikož $A_n = A$).

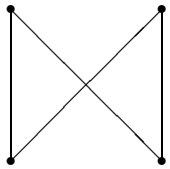
6. Svazy

6.1. Definice Buď $(A, \leq)$ uspřádaná množina a $B \subseteq A$ její podmnožina. Prvek $a \in A$ se nazývá *dolní (horní)* závora množiny B, jestliže pro každé $x \in B$ platí $a \leq x$ ($a \geq x$). Nechť Z značí množinu všech dolních (horních) závor množiny B. Pak největší (nejmenší) prvek množiny Z - pokud existuje - se nazývá *infimum (supremum)* množiny B a označuje se inf B (sup B).

6.2. Příklady. (1) Uvažujme na $\mathbb{R}$ (lineární) uspořádání podle velikosti a nechť $(a, b) \subseteq \mathbb{R}$ je otevřený interval. Pak platí $\inf(a, b) = a$ a $\sup(a, b) = b$.

(2) Buď A množina a uvažujme uspořádanou množinu $(\mathcal{P}(A), \subseteq)$, kde $\mathcal{P}(A)$ značí množinu všech podmnožin množiny A. Pak pro libovolnou podmnožinu $\mathcal{B} \subseteq \mathcal{P}(A)$ platí $\inf \mathcal{B} = \bigcap_{B \in \mathcal{B}} B$ a $\sup \mathcal{B} = \bigcup_{B \in \mathcal{B}} B$.

(3) Uvažujme uspořádanou množinu A s následujícím Hasseovým diagramem:



Pak inf A ani sup A neexistuje.

6.3. Poznámka. Buď $(A, \leq)$ uspořádaná množina. Pak inf A (sup A) je nejmenším (největším) prvkem množiny A. Naopak, inf $\emptyset$ (sup $\emptyset$) je největším (nejmenším) prvkem množiny A.

6.4. Definice. Uspořádaná množina se nazývá *svaz*, jestliže každá její dvouprvková podmnožina $\{a, b\}$ má infimum i supremum. Pak místo $\inf\{a, b\}$ píšeme $a \wedge b$ a místo $\sup\{a, b\}$ píšeme $a \vee b$ - takto definované binární operace operace $\wedge$ a $\vee$ se (po řadě) nazývají *průsek* a *spojení*. Uspořádaná množina se nazývá *úplný svaz*, jestliže každá její podmnožina má infimum a supremum.

Každý úplný svaz je tedy svazem (majícím nejmenší a největší prvek) a v libovolném svazu existuje inf A a sup A pro každou konečnou neprázdnou podmnožinu A (neboť $\inf\{a_1, a_2, ..., a_n\} = a_1 \wedge (a_2 \wedge (... \wedge a_n)...)$ a analogicky pro supremum). Je-li na dané množině S dáno uspořádání $\leq$ takové, že $(S, \leq)$ je svaz, pak budeme stručně říkat, že S je svaz.

6.5. Příklady. (1) Každá lineárně uspořádaná množina je svaz, v němž platí $a \wedge b = \min(a, b)$ a $a \vee b = \max(a, b)$ (kde $\min(a, b)$, resp. $\max(a, b)$ je takový prvek $x \in \{a, b\}$, pro nějž platí $x \leq a$ i $x \leq b$, resp. $x \geq a$ i $x \geq b$).

(2) Značí-li $\mathcal{P}(A)$ množinu všech podmnožin dané množiny A, pak $(\mathcal{P}(A), \subseteq)$ je úplný svaz. V tomto svazu jsou infima průniky a suprema sjednocení (viz příklad 6.2(2)).

(3) Uvažujme uspořádanou množinu $(\mathbb{N}, |)$, kde symbol $|$ značí relaci "dělí". Pak $(\mathbb{N}, |)$ je svaz, v němž pro libovoná přirozená čísla $m, n \in \mathbb{N}$ je $m \wedge n$ největší společný dělitel a $m \vee n$ nejmenší společný násobek čísel m a n.

6.6. Věta. Je-li S svaz, pak pro libovolnou trojici prvků $x, y, z \in S$ platí:
(i) $x \wedge x = x$ a $x \vee x = x$ (idempotence),
(ii) $x \wedge y = y \wedge x$ a $x \vee y = y \vee x$ (komutativita),
(iii) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ a $x \vee (y \vee z) = (x \vee y) \vee z$ (asociativita),
(iv) $x \wedge (x \vee y) = x$ a $x \vee (x \wedge y) = x$ (absorpce),
(v) $x \wedge y = x \Leftrightarrow x \leq y$ a $x \vee y = y \Leftrightarrow x \leq y$ (konzistence).

Důkaz. Platnost vztahů (i),(ii) a (v) plyne ihned z definice 6.4.
Vztah (iii): Zřejmě $(x \wedge y) \wedge z \leq x \wedge y \leq x$ a také $(x \wedge y) \wedge z \leq x \wedge y \leq y$, dále $(x \wedge y) \wedge z \leq z$. Tedy $(x \wedge y) \wedge z \leq y \wedge z$, proto $(x \wedge y) \wedge z \leq x \wedge (y \wedge z)$. Podobně se ukáže opačná nerovnost, takže $x \wedge (y \wedge z) = (x \wedge y) \wedge z$. Pro spojení je důkaz analogický.
Vztah (iv): Zřejmě $x \wedge (y \vee x) \leq x$. Protože $x \leq x$ a $x \leq x \vee y$, máme $x \leq x \wedge (x \vee y)$. Tedy $x \wedge (x \vee y) = x$. Pro spojení je důkaz analogický.

6.7. Poznámka a) V důsledku vztahu (iii) z předchozí věty můžeme ve svazu pro libovolnou neprázdnou konečnou množinu $A = \{a_1, a_2, ..., a_n\}$ psát $\inf A = a_1 \wedge a_2 \wedge ... \wedge a_n$ (a analogicky pro supremum).

b) Každý svaz můžeme chápat jako množinu S se dvěma binárními operacemi $\wedge$ a $\vee$, které splňují vztahy (i)-(iv) z předchozí věty. Příslušné uspořádání na S je pak dáno vztahem (v).

c) Má-li svaz S nejmenší prvek 0 (největší prvek 1), pak pro libovolný prvek $x \in S$ platí $x \wedge 0 = 0$ a $x \vee 0 = x$ ($x \wedge 1 = x$ a $x \vee 1 = 1$).

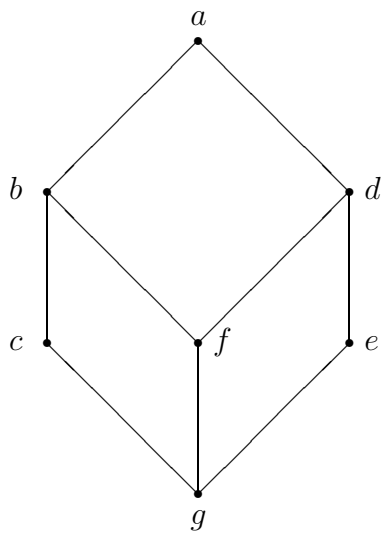
6.8. Definice. Buď $(S, \leq)$ svaz a $(S', \leq)$ jeho uspořádaná podmnožina. Pak $(S', \leq)$ se nazývá *podsvaz* svazu $(S, \leq)$, jestliže pro libovolné prvky $x, y \in S'$ platí $x \wedge y \in S'$ i $x \vee y \in S'$.

Zřejmě tedy každý podsvaz daného svazu $(S, \leq)$ je sám také svaz (v němž spojení, resp. průsek libovolné dvojice prvků se rovná jejich spojení, resp. průseku v $(S, \leq)$). Ovšem uspořádaná podmnožina daného svazu $(S, \leq)$ může být sama svazem, i když není podsvazem svazu $(S, \leq)$. Je-li $(S', \leq)$ podsvaz svazu $(S, \leq)$, budeme stručně říkat, že S' je podsvaz svazu S.

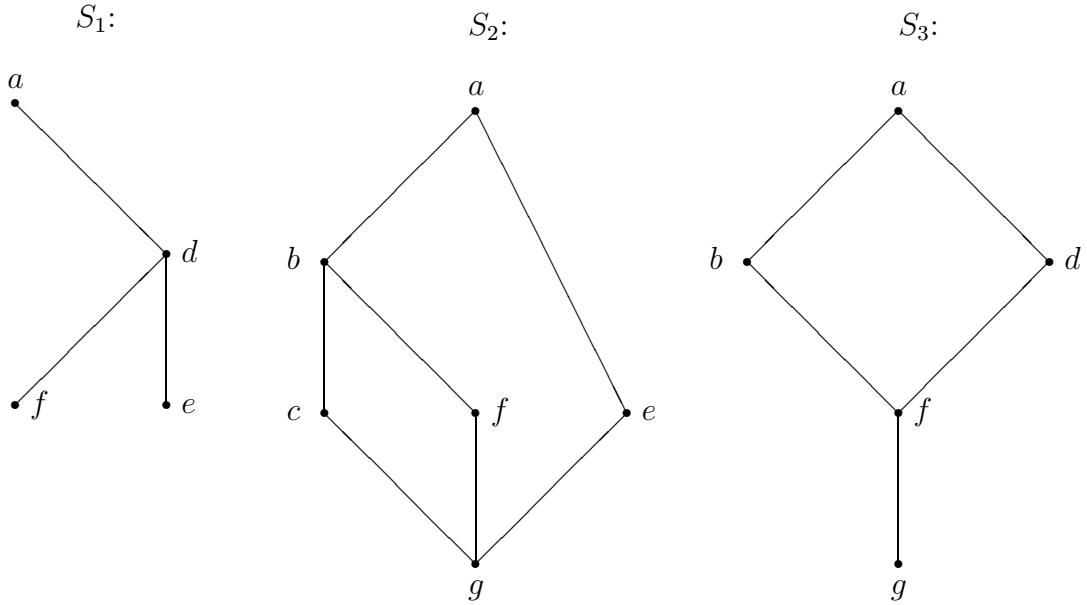
6.9. Příklady. (1) Uvažujme na každé z množin $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ a $\mathbb{R}$ lineární uspořádání podle velikosti. Pak jsou tyto množiny svazy, přičemž $\mathbb{N}$ je podsvaz $\mathbb{Z}$, $\mathbb{Z}$ je podsvaz $\mathbb{Q}$ a $\mathbb{Q}$ je podsvaz $\mathbb{R}$ (viz 5.15(1)).

(2) Nechť A, B jsou množiny, $A \subseteq B$. Značí-li $\mathcal{P}(A)$ množinu všech podmnožin množiny A a $\mathcal{P}(B)$ množinu všech podmnožin množiny B, pak $(\mathcal{P}(A), \subseteq)$ je podsvaz svazu $(\mathcal{P}(B), \subseteq)$ (viz 5.15(2)).

(3)Buď S svaz daný následujícím Hasseovým diagramem:



Uvažujme následující tři uspořádané podmnožiny svazu S:



Pak S_1 zřejmě není svaz. S_2 je svaz, ale není podsvaz svazu S (neboť $f \vee e = d \notin S_2$). S_3 je podsvaz svazu S.

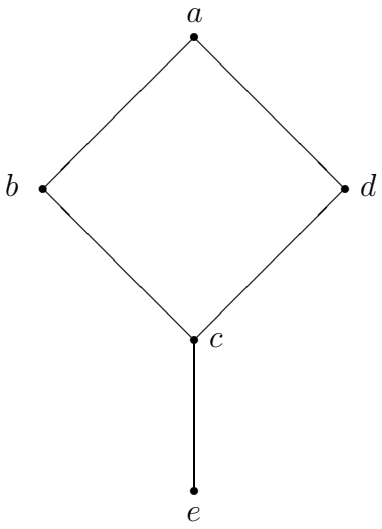
6.10. Definice. Svaz S se nazývá *distributivní*, jestliže pro libovolnou trojici prvků $x, y, z \in S$ platí

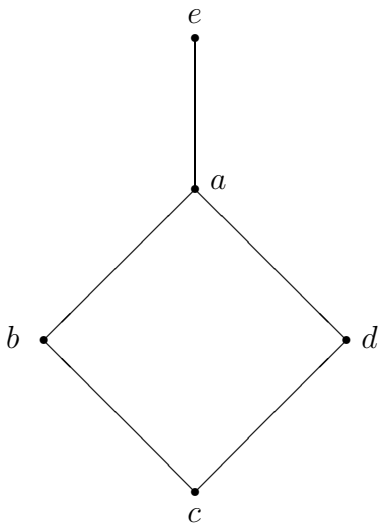
$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$$

6.11. Příklady. (1) Pro libovolnou množinu A je svaz $(\mathcal{P}(A), \subseteq)$ distributivní (symbolem $\mathcal{P}(A)$ značíme množinu všech podmnožin množiny A). Připomeňme, že podle příkladu 6.2.(2) ve svazu $(\mathcal{P}(A), \subseteq)$ pro libovolné dvě podmnožiny $B, C \subseteq A$ platí $B \wedge C = B \cap C$ a $B \vee C = B \cup C$.

(2) Každá lineárně uspořádaná množina je distributivní svaz.

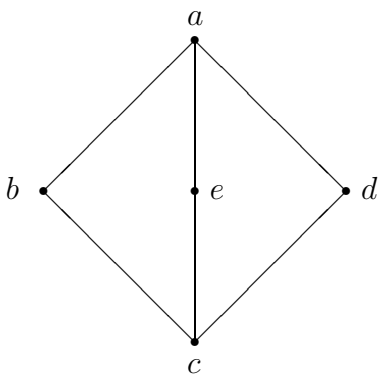
(3) Svazy s následujícími Hasseovými diagramy jsou distributivní:



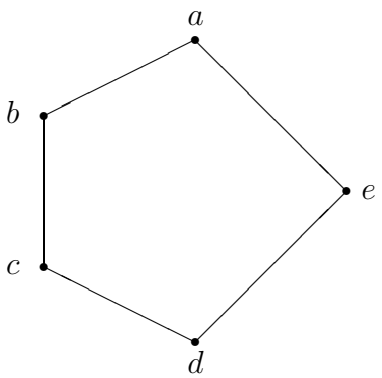


(4) Svazy P a D s následujícími Hasseovými diagramy nejsou distributivní:

D:



P:



Ve svazu D totiž platí $e \vee d = a$ a $b \wedge e = b \wedge d = c$, takže $b \wedge (e \vee d) = b$, zatímco $(b \wedge e) \vee (b \wedge d) = c$. Ve svazu P pak platí $e \vee c = a$, $b \wedge e = d$ a $b \wedge c = c$, takže $b \wedge (e \vee c) = b$, zatímco $(b \wedge e) \vee (b \wedge c) = c$.

6.12. Cvičení. a) Dokažte, že libovolný svaz S je distributivní, právě když pro libovolnou trojici prvků $x, y, z \in S$ platí podmínka

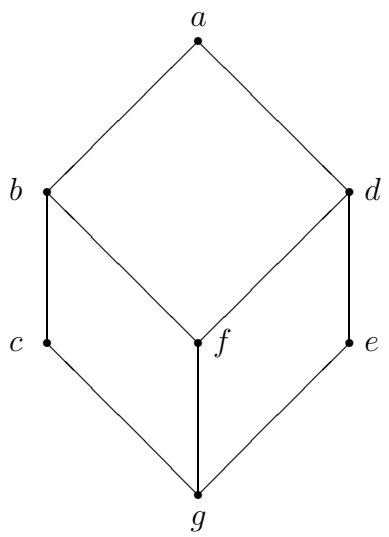
$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z),$$

tj., že tato podmínka je ekvivalentní podmínce z definice 6.10.

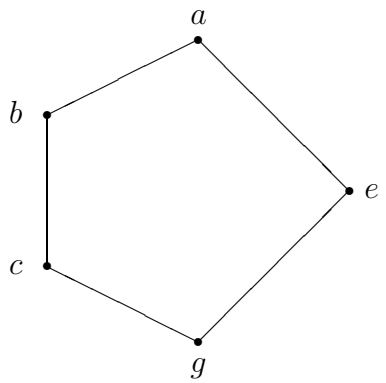
b) Dokažte, že svaz $(\mathbb{N}, |)$ (kde symbol $|$ značí relaci "dělí") je distributivní - viz příklad 6.5(3).

6.13. Poznámka. Je zřejmé, že každý podsvaz distributivního svazu je distributivní. Tedy žádný distributivní svaz nemůže obsahovat podsvaz tvaru D ani P z příkladu 6.11(4). Je známo, že platí i opačné tvrzení, takže libovolný svaz je distributivní, právě když neobsahuje podsvaz tvaru D ani P z příkladu 6.11(4).

6.14. Příklad. Svaz S s následujícím Hasseovým diagramem



není distributivní, neboť obsahuje následující podsvaz tvaru P z příkladu 6.11.(4):



6.15. Věta. Ke každému distributivnímu svazu S existuje množina A, distributivní podsvaz L svazu $(\mathcal{P}(A), \subseteq)$ a bijekce $f : S \to L$ taková, že pro libovolné prvky $a, b \in S$ platí $f(a \vee b) = f(a) \cup f(b)$ a $f(a \wedge b) = f(a) \cap f(b)$.

Důkaz předchozí věty přesahuje rámec tohoto učebního textu, proto jej vynecháváme. Tato věta říká, že každý distributivní svaz lze nalézt jako podsvaz (až na označení prvků) svazu $(\mathcal{P}(A), \subseteq)$, kde A je jistá množina.

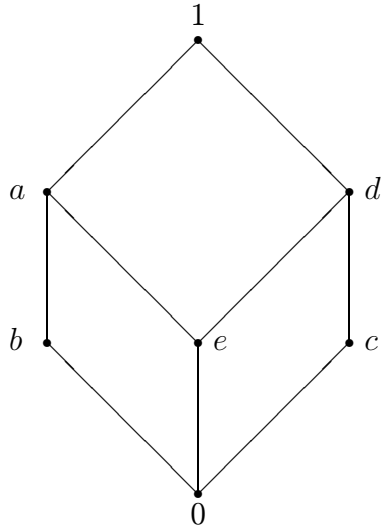
6.16. Definice. (a) Má-li svaz nejmenší i největší prvek, pak se tento svaz nazývá *ohraničený*.

(b) Buď S ohraničený svaz, jehož nejmenší prvek je označen symbolem 0 a největší prvek symbolem 1. *Komplementem* prvku $a \in S$ rozumíme každý takový prvek $\bar{a} \in S$, pro nějž platí $a \wedge \bar{a} = 0$ a $a \vee \bar{a} = 1$.

(c) Ohraničený svaz, v němž ke každému prvku existuje komplement, se nazývá *komplementární*.

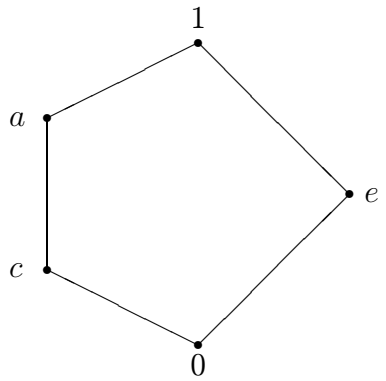
I v dalším textu budeme symboly 0 a 1 označovat nejmenší a největší prvek daného (ohraničeného) svazu. Zřejmě vždy platí $\bar{0} = 1$ a $\bar{1} = 0$. Poznamenejme také, že v ohraničeném svazu daný prvek obecně nemusí mít žádný komplement, nebo může mít jeden či více komplementů.

6.17. Příklady. (1) Uvažujme opět svaz S s následujícím Hasseovým diagramem:



Pak prvek e nemá žádný komplement, prvky $a, d, 0, 1$ mají po jednom komplementu a prvky b, c po dvou komplementech (např. prvky c i d jsou komplementy prvku b). Tento svaz tedy není komplementární.

(2) I v komplementárním svazu může mít daný prvek více než jeden komplement. Např. ve svazu s následujícím Hasseovým diagramem



má prvek c dva komplementy - prvky a a b.

6.18. Věta. V libovolném distributivním komplementárním svazu má každý prvek právě jeden komplement.

Důkaz. Buď S distributivní komplementární svaz a $x \in S$ jeho libovolný prvek. Stačí ukázat, že x nemá více než jeden komplement. Předpokládejme tedy, že prvky $y_1, y_2 \in S$ jsou komplementy prvku x. Pak $x \wedge y_1 = 0 = x \wedge y_2$ a $x \vee y_1 = 1 = x \vee y_2$. Máme tedy $y_1 = y_1 \vee 0 = y_1 \vee (x \wedge y_2) = (y_1 \vee x) \wedge (y_1 \vee y_2) = 1 \wedge (y_1 \vee y_2) = y_1 \vee y_2$.

Odtud dostáváme $y_2 \leq y_1$. Podobně (vzájemnou záměnou y_1 a y_2) se ukáže, že také $y_1 \leq y_2$. Proto $y_1 = y_2$, čímž je důkaz hotov.

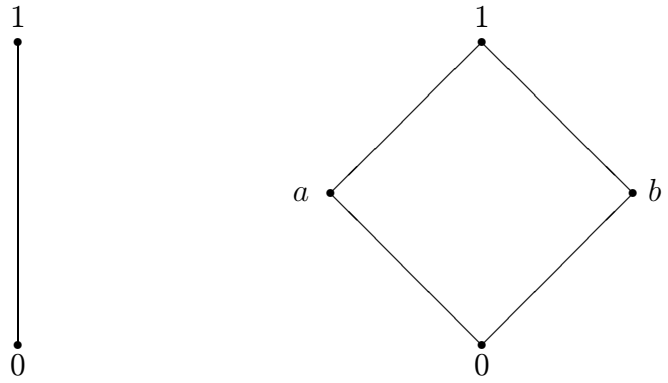
6.19. Příklad. Pro libovolnou množinu A je $(\mathcal{P}(A), \subseteq)$ distributivní a komplementární svaz. Je-li $B \subseteq A$ nějaká podmnožina, pak komplementem množiny B v tomto svazu je množina $A - B$ (tzv. *doplněk* množiny B v množině A).

7. Booleovy svazy

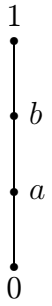
7.1. Definice. Distributivní a komplementární svaz B se nazývá *Booleův svaz*. Binární operace $\wedge$ a $\vee$ a unární operace $^-$ na B se nazývají *základními Booleovými operacemi*.

7.2. Příklady. (1) Pro libovolnou množinu A je $(\mathcal{P}(A), \subseteq)$ Booleův svaz.
(2) Svazy S a L s následujícími Hasseovými diagramy jsou Booleovy:

S : L :

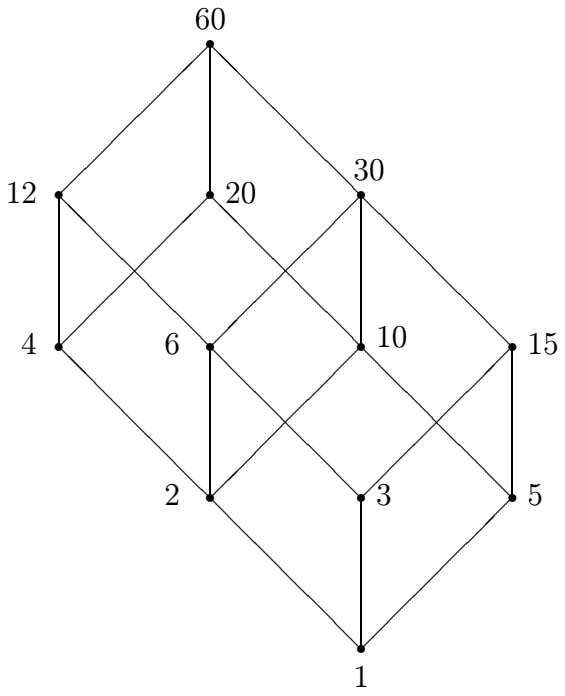


(3) Svazy D a P z příkladu 6.11(4) nejsou Booleovy, neboť nejsou distributivní.
(4) Buď S svaz s následujícím Hasseovým diagramem:



Pak S není Booleův, neboť není komplementární (prvky a ani b nemají komplement).

(5) Buď L množina všech kladných dělitelů čísla 60. Pak svaz $(L, |)$ je distributivní, neboť je podsvazem distributivního svazu z cvičení 6.12b). Jeho Hasseův diagram má následující tvar:



Ovšem $(L, |)$ není Booleův svaz, neboť není komplementární: čísla 2,6,10,30 nemají komplementy (připomeňme, že infima jsou největší společní dělitelé a suprema jsou nejmenší společné násobky).

7.3. Věta. Buď B Booleův svaz. Pak
(i) pro libovolný prvek $x \in B$ platí $\bar{\bar{x}} = x$,
(ii) pro libovolné prvky $x, y, z \in B$ platí $\overline{x \vee y} = \bar{x} \wedge \bar{y}$ a $\overline{x \wedge y} = \bar{x} \vee \bar{y}$ (de Morganova pravidla).

Důkaz. (i) Prvek x je zřejmě komplementem prvku $\bar{x}$ a protože $\bar{x}$ má podle věty 6.18 jediný komplement, máme $\bar{\bar{x}} = x$.
(ii) Z distributivity plyne $(x \vee y) \wedge (\bar{x} \wedge \bar{y}) = (\bar{x} \wedge \bar{y} \wedge x) \vee (\bar{x} \wedge \bar{y} \wedge y) = (\bar{y} \wedge 0) \vee (\bar{x} \wedge 0) = 0 \wedge 0 = 0$ a $(x \vee y) \vee (\bar{x} \wedge \bar{y}) = (x \vee y \vee \bar{x}) \wedge (x \vee y \vee \bar{y}) = (y \vee 1) \wedge (x \vee 1) = 1 \wedge 1 = 1$ (viz cvičení 6.12). Tedy $\bar{x} \wedge \bar{y}$ je komplementem prvku $x \vee y$. Analogicky se dokáže druhá rovnost.

7.4. Příklady. (1) V Booleově svazu $(\mathcal{P}(A), \subseteq)$, kde A je libovolná množina, mají de Morganova pravidla známý množinový tvar: Pro libovolné množiny $X, Y, Z \in \mathcal{P}(A)$ platí
$A - (X \cup Y) = (A - X) \cap (A - Y)$,
$A - (X \cap Y) = (A - X) \cup (A - Y)$.
(2) Uvažujme svaz z příkladu 7.2(5). Tento svaz není Booleův (jelikož není komplementární), avšak pro některé jeho prvky platí de Morganova pravidla. Ověřme, že tato pravidla platí pro prvky 3 a 4, tj., že platí $\overline{\text{n.s.n.}(3,4)} = \text{n.s.d.}(\bar{3}, \bar{4})$ a $\overline{\text{n.s.d.}(3,4)} = \text{n.s.n.}(\bar{3}, \bar{4})$ (samozřejmě, n.s.n. značí nejmenší společný násobek a n.s.d. největší společný dělitel). Skutečně, máme:

$\overline{\text{n.s.n.}(3,4)} = \overline{12} = 5 = \text{n.s.d.}(20,15) = \text{n.s.d.}(\bar{3},\bar{4})$ a
$\overline{\text{n.s.d.}(3,4)} = \bar{1} = 60 = \text{n.s.n.}(20,15) = \text{n.s.n.}(\bar{3},\bar{4})$.

7.5. Poznámka (princip duality). Je-li $(B,\leq)$ Booleův svaz, pak $(B,\leq^{-1}) = (B,\geq)$ je také Booleův svaz, v němž infima (suprema, nejmenší prvek, největší prvek) jsou suprema (infima, největší prvek, nejmenší prvek) svazu $(B,\leq)$. Odtud plyne tzv. princip duality: Pokud platí pro Booleovy svazy nějaké tvrzení, v němž se vyskytují symboly $\leq,\ \wedge,\ \vee,\ 0,\ 1$, pak platí také tvrzení, které dostaneme záměnou těchto symbolů po řadě za $\geq,\ \vee,\ \wedge,\ 1,\ 0$.

7.6. Věta. Je-li B Booleův svaz, pak pro libovolné prvky $x, y \in B$ platí:
$x \wedge y = x \wedge (\bar{x} \vee y)$ a
$x \vee y = x \vee (\bar{x} \wedge y)$.

Důkaz. Zřejmě platí $x \wedge (\bar{x} \vee y) = (x \wedge \bar{x}) \vee (x \wedge y) = 0 \vee (x \wedge y) = x \wedge y$. Druhá rovnost plyne z principu duality.

7.7. Cvičení. Dokažte, že pro libovolné prvky $x,\ y$ Booleova svazu B platí:
(1) $x = y \Leftrightarrow (x \vee \bar{y}) \wedge (\bar{x} \vee y) = 1$,
(2) $x \leq y \Leftrightarrow \bar{y} \leq \bar{x} \Leftrightarrow \bar{x} \vee y = 1 \Leftrightarrow x \wedge \bar{y} = 0$,

7.8. Definice. Buď B Booleův svaz a B' jeho podsvaz mající alespoň jeden prvek. Pak B' se nazývá *Booleův podsvaz* Booleova svazu B, právě když pro libovolný prvek $x \in B'$ platí $\bar{x} \in B'$.

7.9. Věta. Booleův podsvaz Booleova svazu B je Booleovým svazem, který má stejné prvky 0 a 1 jako B.

Důkaz. Buď B' Booleův podsvaz Booleova svazu B. Z definice 7.8. ihned plyne, že B' je distributivní a že komplementy prvků v B' jsou stejné jako komplementy těchto prvků v B. Stačí tedy ukázat, že prvky 0 a 1 svazu B leží v B'. To je ovšem snadné, neboť pro libovolný prvek $x \in B'$ máme $\bar{x} \in B'$, takže $x \wedge \bar{x} = 0 \in B'$ a $x \vee \bar{x} = 1 \in B'$.

7.10. Příklad. Buď $A = \{a,b,c\}$ množina a uvažujme Booleův svaz $B = (\mathcal{P}(A),\subseteq)$, jehož Hasseův diagram je uveden v příkladu 5.9. Pak jeho podsvaz $B' = (\{\emptyset,\{a\},\{b,c\},\{a,b,c\}\},\subseteq)$ je jeho Booleovým podsvazem. To však neplatí např. pro podsvaz $C = (\{c\},\{a,c\},\{b,c\},\{a,b,c\}\},\subseteq)$, neboť v tomto případě máme $\{b,c\} \in C$, avšak $\overline{\{b,c\}} = \{a\} \notin C$.

Užitím věty 6.15 lze dokázat následující tvrzení:

7.11. Věta. Buď B Booleův svaz. Pak existuje množina A, Booleův podsvaz L Booleova svazu $(\mathcal{P}(A),\subseteq)$ a bijekce $f : B \to L$ taková, že pro libovolné prvky

$a, b \in B$ platí $f(a \wedge b) = f(a) \cap f(b)$, $f(a \vee b) = f(a) \cup f(b)$ a $f(\bar{a}) = A - f(a)$. Je-li svaz B konečný, lze volit $L = (\mathcal{P}(A), \subseteq)$.

7.12. Poznámka. Podobně jako můžeme každý svaz chápat jako množinu se dvěma binárními operacemi $\wedge$ a $\vee$ splňujícími dané axiomy (viz poznámku 6.7b)), také každý Booleův svaz můžeme chápat jako množinu se dvěma binárními operacemi $\wedge$ a $\vee$, jednou unární operací $^-$ a prvky 0 a 1 splňujícími dané axiomy (tj. podmínky (i)-(iv) z věty 6.6, rovnosti z poznámky 6.7c), rovnost z definice 6.10 a rovnosti z definice 6.16(b)). Místo o Booleových svazech pak hovoříme o *Booleových algebrách* (příslušné uspořádání je opět dáno vztahem (v) z věty 6.6).

8. Booleovy funkce

Připomeňme (viz poznámku 3.17), že pro libovolnou množinu A a libovolné $n \in \mathbb{Z}^+$ klademe $A^n = \underbrace{A \times A \times ... \times A}_{n-\text{krát}}$, tj., A^n je množina všech n-tic tvořených prvky množiny A s ohledem na pořadí těchto prvků.

8.1. Definice. Buď B Booleův svaz a n přirozené číslo. Booleovou funkcí n proměnných v Booleově svazu B rozumíme libovolnou funkci $F : B^n \to B$, kterou je možno zapsat (v explicitním tvaru) užitím pouze symbolů označujících proměnné, závorek a symbolů označujících základní Booleovy operace, tedy symbolů $\wedge$, $\vee$ a $^-$.

8.2. Příklad. V libovolném Booleově svazu jsou
$F(x, y) = x \wedge y$, $F(x, y) = x \vee y$, $F(x, y) = (x \wedge \bar{y}) \vee (\bar{x} \wedge y)$
Booleovy funkce dvou proměnných a
$F(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee ((x_3 \vee x_4) \wedge \overline{(x_1 \wedge x_2)})) \wedge (x_2 \vee (x_2 \wedge x_3))$
je Booleova funkce čtyř proměnných.

Snadno se dokáže následující tvrzení:

8.3. Věta. Buď $B = (B, \leq)$ Booleův svaz a n přirozené číslo. Označme symbolem $\mathcal{F}_n$ množinu všech Booleových funkcí n proměnných v Booleově svazu B a pro libovolné $F, G \in \mathcal{F}_n$ položme $F \preceq G \Leftrightarrow F(x_1, ..., x_n) \leq G(x_1, ..., x_n)$ pro všechna $x_1, ..., x_n \in B$. Pak $(\mathcal{F}_n, \preceq)$ je Booleův svaz, ve kterém pro libovolné $F, G \in \mathcal{F}_n$) a libovolné $x_1, ..., x_n$ platí:
$(F \wedge G)(x_1, ..., x_n) = F(x_1, ..., x_n) \wedge G(x_1, ..., x_n)$,
$(F \vee G)(x_1, ..., x_n) = F(x_1, ..., x_n) \vee G(x_1, ..., x_n)$,
$\bar{F}(x_1, ..., x_n) = \overline{F(x_1, ..., x_n)}$.
Booleova funkce H n proměnných v B daná vztahem $H(x_1, ..., x_n) = 0$, resp. $H(x_1, ..., x_n) = 1$ pro všechna $x_1, ..., x_n \in B$ je nejmenším, resp. největším prvkem Booleova svazu $\mathcal{F}_n$.

8.4. Příklad. Buď $B = (\{0, 1\}, \leq)$ Booleův svaz (mající právě dva prvky: nejmenší a největší). Je dobře známo, že pro libovolné přirozené číslo n je každá funkce $F : \{0, 1\}^n \to \{0, 1\}$ Booleovou funkcí n-proměnných v B. Tedy množina $\mathcal{F}_2$ všech Booleových funkcí dvou proměnných v B je dána následující tabulkou:

x	y	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

V $(\mathcal{F}_2, \preceq)$ např. platí $F_{10} = (\bar{a} \vee b) \wedge (a \vee \bar{b})$, $F_6 \wedge F_7 = F_5$, $F_6 \vee F_7 = F_8$, $F_3 \preceq F_8$, prvky F_4 a F_{15} jsou nesrovnatelné, F_1 je nejmenší prvek a F_{16} největší prvek.

8.5. Cvičení. a) Dokažte, že platí: Má-li Booleův svaz B m prvků, pak příslušný Booleův svaz $\mathcal{F}_n$ má nejvýše m^{m^n} prvků.

b) Určete explicitní tvary všech šestnácti Booleových funkcí z příkladu 8.4.

8.6. Definice. Buď B Booleův svaz a $F(x_1, ..., x_n)$ Booleova funkce n proměnných v B. Řekneme, že $F(x_1, ..., x_n)$ má *úplný disjunktní normální tvar*, jestliže platí $F(x_1, ..., x_n) = (G_1 \vee ... \vee G_m)(x_1, ..., x_n)$, kde m je přirozené číslo a $G_1, ..., G_m$ jsou navzájem různé Booleovy funkce n-proměnných v B takové, že pro libovolné $i \in \{1, ..., n\}$ platí $G_i(x_1, ..., x_n) = x_1^* \wedge x_2^* \wedge ... \wedge x_n^*$, přičemž $x_j^* = x_j$ nebo $x_j^* = \bar{x}_j$ pro každé $j \in \{1, ..., n\}$.

8.7. Příklad. Následující Booleovy funkce jsou zapsány v úplném disjunktním normálním tvaru:

$F(x, y) = (x \wedge y) \vee (\bar{x} \wedge y) \vee (\bar{x} \wedge \bar{y})$,

$G(x, y, z) = (\bar{x} \wedge y \wedge z) \vee (\bar{x} \wedge y \wedge \bar{z})$.

8.8. Věta. Každou Booleovu funkci lze jednoznačně (až na pořadí funkcí G_i) vyjádřit v úplném disjunktním normálním tvaru.

Důkaz. Důkaz provedeme konstruktivně, tj. ukážeme postup, jakým lze převést danou Booleovu funkci do úplného disjunktního normálního tvaru. Tento postup se skládá z následujících čtyř kroků:

1. Pomocí de Morganových pravidel docílíme toho, že znaménko komplementu se bude vyskytovat jen u (některých) proměnných.

2. Opakovaným užitím distributivního zákona a základních vlastností průseku, spojení a komplementu převedeme funkci do tzv. *disjunktního normálního tvaru*, který se od úplného disjunktního normálního tvaru liší jen tím, že funkce G_i nemusí obsahovat x_j^* pro všechna $j \in \{1, ..., n\}$.

3. Každou funkci G_i, která neobsahuje x_j^* pro některá $j \in \{1, ..., n\}$, nahradíme průsekem této funkce a výrazů $(x_j \vee \bar{x}_j)$ pro všechna takováto j.

4. Užitím distributivního zákona již obdržíme úplný disjunktní normální tvar dané funkce (který je určen jednoznačně až na pořadí funkcí G_i a výrazů x_j^*).

8.9. Příklad. Převeďme následující funkce do úplného disjunktního normálního tvaru:

a) $F(x, y, z) = (x \vee y) \wedge (x \vee z) \wedge \overline{(\bar{x} \vee \bar{y})}$,

b) $G(x, y, z) = \bar{y} \vee [(\bar{z} \vee x \vee \overline{(y \wedge z)}) \wedge (z \vee (\bar{x} \wedge y))]$.

Užitím postupu z důkazu věty 8.8 dostaneme:

a) $F(x, y, z) = [x \vee (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)] \wedge x \wedge y = (x \wedge y) \vee (x \wedge y \wedge z) = x \wedge y \wedge (z \vee \bar{z}) \vee (x \wedge y \wedge z) = (x \wedge y \wedge z) \vee (x \wedge y \wedge \bar{z})$,

b) $G(x,y,z) = \bar{y} \vee [(\bar{z} \vee x \vee (\bar{y} \vee \bar{z})) \wedge (z \vee (\bar{x} \wedge y))] = \bar{y} \vee [(\bar{z} \vee x \vee \bar{y}) \wedge (z \vee (\bar{x} \wedge y))] = \bar{y} \vee [0 \vee (x \wedge z) \vee (\bar{y} \wedge z) \vee (\bar{z} \wedge \bar{x} \wedge y) \vee 0 \vee 0] = \bar{y} \vee (x \wedge z) \vee (\bar{y} \wedge z) \vee (\bar{x} \wedge y \wedge \bar{z}) = [\bar{y} \wedge (x \vee \bar{x}) \wedge (z \vee \bar{z})] \vee [(x \wedge z) \wedge (y \vee \bar{y})] \vee [(\bar{y} \wedge z) \wedge (x \vee \bar{x})] \vee (\bar{x} \wedge y \wedge \bar{z}) = [(x \wedge \bar{y} \wedge z) \vee (\bar{x} \wedge \bar{y} \wedge z) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (\bar{x} \wedge \bar{y} \wedge \bar{z})] \vee [(x \wedge y \wedge z) \vee (x \wedge \bar{y} \wedge z)] \vee [(x \wedge \bar{y} \wedge z) \vee (\bar{x} \wedge \bar{y} \wedge z)] \vee (\bar{x} \wedge y \wedge \bar{z}) = (x \wedge \bar{y} \wedge z) \vee (\bar{x} \wedge \bar{y} \wedge z) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (\bar{x} \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge y \wedge z) \vee (\bar{x} \wedge y \wedge \bar{z})$.

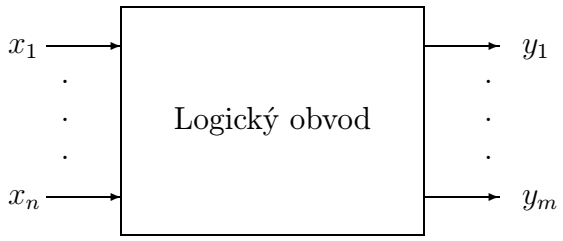
8.10. Cvičení. Nalezněte úplné disjunktní normální tvary následujících Booleových funkcí:

1) $F(x,y,z) = \overline{[x \vee (\bar{y} \wedge z)]} \wedge (\bar{z} \vee y)$,
2) $F(x,y,z) = \overline{x \vee (y \wedge (x \vee z))}$,
3) $F(x,y,z) = \overline{(x \vee y)} \vee (x \wedge z)$,
4) $F(x,y,z) = [(x \wedge y) \vee ((\bar{y} \vee z) \wedge \bar{x})] \wedge (y \vee \bar{z})$,
5) $F(x,y,z,u) = (\bar{y} \wedge \bar{u}) \vee [(x \vee z) \wedge ((\bar{y} \wedge \bar{z}) \vee (x \wedge u))]$.

9. Aplikace Booleových svazů

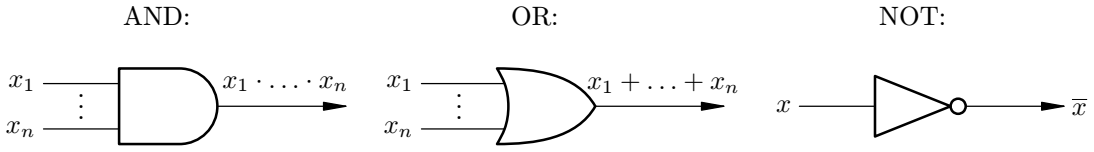
a) Logické obvody

Významného využití nacházejí Booleovy svazy při navrhováni tzv. *logických obvodů*, které jsou součástí každého počítače. Logický obvod je zařízení, které přetváří vstupní (elektrické) signály na signály výstupní. Všechny signály nabývají pouze dvou hodnot, které budeme označovat symboly 0 a 1. Symbolem **2** pak označíme Booleův svaz $(\{0,1\},\leq)$ (takže **2** obsahuje pouze nejmenší prvek 0 a největší prvek 1). Logický obvod mající n vstupů a m výstupů pak představuje zobrazení $L : \mathbf{2}^n \to \mathbf{2}^m$.



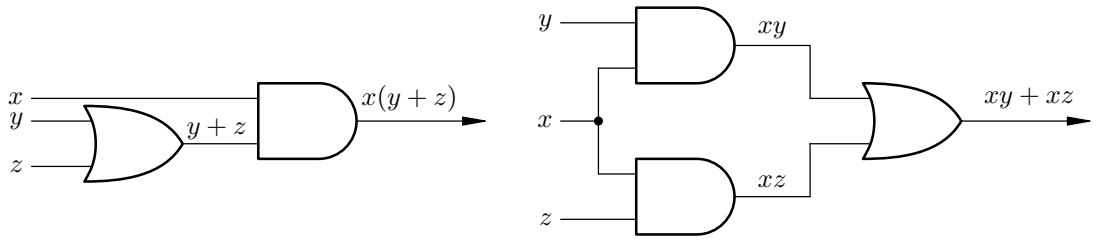
Dále budeme uvažovat logické obvody mající pouze jediný výstup. Každý takový obvod představuje Booleovu funkci n proměnných $F : \mathbf{2}^n \to \mathbf{2}$ - tato funkce je tedy daným obvodem realizována. Ve svazu **2** budeme psát $\cdot$ místo $\wedge$ a $+$ místo $\vee$ a při závorkování budeme využívat vyšší prioritu operace $\cdot$ (podobně jako to děláme při násobení a sčítání reálných čísel). Jak je zvykem, budeme při zápisu symbol $\cdot$ vynechávat. Platí tedy např. $xx = x$, $x + x = x$, $0x = 0$, $1x = x$, $0 + x = x$ a $1 + x = 1$. Operace $\cdot$ a $+$ se nazývají *logický součin* a *logický součet*.

Abychom mohli sestavit logický obvod, musíme mít prvky, které realizují základní Booleovy operace $\cdot$, $+$ a $^-$. Tyto prvky se nazývají AND, OR a NOT, souhrnně jim říkáme *logické členy* (nebo *hradla*). Logické členy se označují následovně:



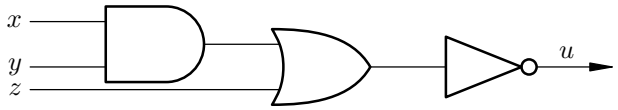
Schémata, která znázorňují logické obvody, se skládají z hran, které představují elektrické vodiče, a z uzlů tvořených logickými členy. Dva logické obvody se nazývají *ekvivalentní*, jestliže realizují tutéž Booleovu funkci.

9.1. Příklad. Následující dva logické obvody jsou ekvivalentní (v důsledku distributivního zákona):



Při analýze logických obvodů řešíme úkol nalezení Booleovy funkce, která je realizována daným logickým obvodem. Jeden z možných způsobů řešení tohoto problému je určení hledané Booleovy funkce pomocí tabulky, tj. zjištění výstupních hodnot pro všechny možné kombinace vstupních signálů. Mnohem efektivnější však je určení explicitního vyjádření hledané Booleovy funkce pomocí Booleových operací.

9.2. Příklad. Nalezněte Booleovu funkci, která je realizována logickým obvodem



Hledaná funkce je dána následující tabulkou:

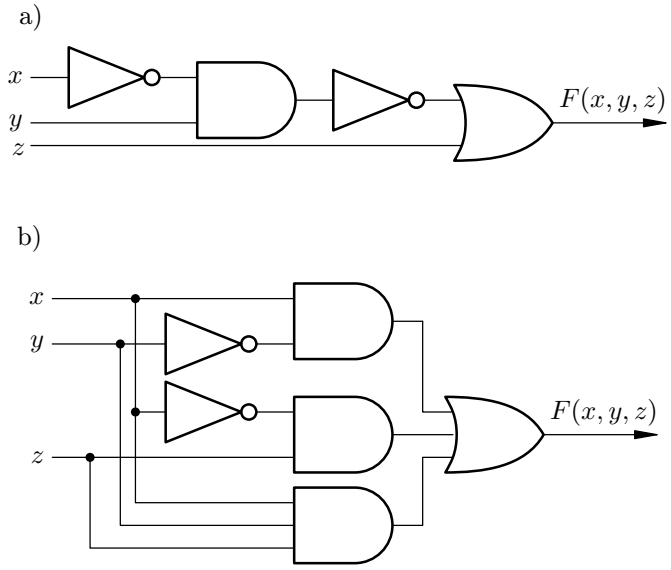
x	y	z	u
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Exlicitní tvar hledané Booleovy funkce je zřejmě

$$t = \overline{xy + z}.$$

Důležitějším problémem než analýza logických obvodů je ovšem úloha opačná, tj. jejich syntéza. Jde tedy o to, sestavit logický obvod, který realizuje danou Booleovu funkci.

9.3. Příklad. Sestavte logický obvod, který realizuje Booleovu funkci
a) $F(x, y, z) = \overline{\bar{x}y} + z$,
b) $F(x, y, z) = x\bar{y} + \bar{x}z + xyz$.
Řešením jsou zřejmě následující logické obvody:



Při syntéze logických obvodů je samozřejmě žádoucí nalézt mezi všemi logickými obvody realizujícími danou Booleovu funkci F takový, který je nejjednodušší, tj., který obsahuje nejmenší počet logických členů. K dosažení tohoto cíle se obvykle postupuje tak, že se daná Booleova funkce převede na úplný disjunktní normální tvar, který se pak minimalizuje a teprve potom realizuje logickým obvodem, který je pak minimální co do počtu logických členů. Pro minimalizaci Booleových funkcí se používá několik metod, z nichž nejjednodušší je tzv. *přímá metoda*, která je založena na vlastnostech Booleových operací. Ukažme si příklad:

9.4. Příklad. Minimalizujme následující Booleovu funkci v úplném disjunktním normálním tvaru:

$$F(x, y, z) = xyz + xy\bar{z} + x\bar{y}z + \bar{x}yz.$$

Z idempotence a distributivity operací $\cdot$ a $+$ a z vlastností největšího prvku a komplementu dostáváme:

$F(x, y, z) = (xyz + xy\bar{z}) + (x\bar{y}z + xyz) + (\bar{x}yz + xyz) = xy(z + \bar{z}) + xz(y + \bar{y}) + yz(x + \bar{x}) = xy \cdot 1 + xz \cdot 1 + yz \cdot 1 = xy + xz + yz$.

Přímá metoda je vhodná jen pro minimalizaci velmi jednoduchých Booleových funkcí a pro více než tři proměnné je prakticky nepoužitelná. Proto byly vyvinuty

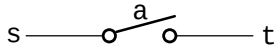
další metody, z nichž zde jmenujme alespoň metodu *Quineovu-McCluskeyovu*, neboť z ní vychází řada dalších metod, které s výhodou vyžívají počítače.

9.5. Cvičení. Minimalizujte přímou metodou Booleovu funkci $F(x, y, z) = \bar{x}\bar{y}z + xy\bar{z} + xyz$ [výsledek: $F(x, y, z) = xy + \bar{x}\bar{y}z$].

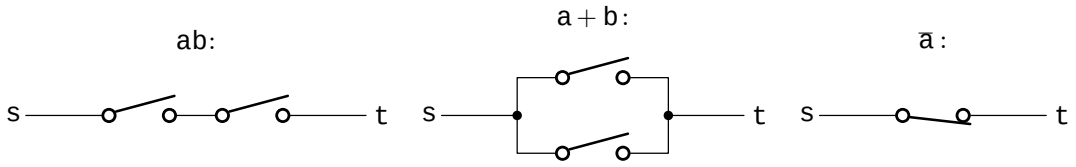
b) Spínačové obvody

Booleovy svazy se používají také při navrhování (elektrických) spínačových obvodů. Spínač je zařízení, které může mít jen stav zapnuto nebo vypnuto.

Spínač a:

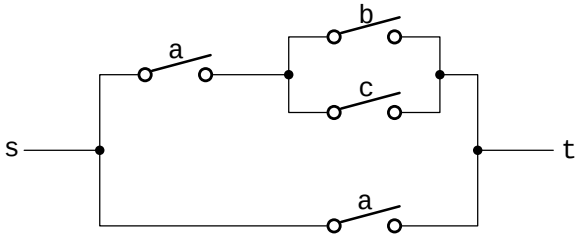


Spínačový obvod pak obsahuje pouze spínače, jeden vstup s a jeden výstup t. Obvod je otevřen, proudí-li signál (elektrický proud) od vstupu k výstupu, což značíme symbolem 1. V opačném případě je obvod uzavřen, což značíme symbolem 0. Také samotný spínač je obvodem, tedy symbol 1 značí jeho zapnutý stav a symbol 0 vypnutý stav. Sériové zapojení spínačů a a b značíme ab a jejich paralelní zapojení značíme $a + b$. Je-li a spínač, pak $\bar{a}$ značí spínač, který je zapnut, právě když a je vypnut, a naopak.



Každý spínačový obvod představuje Booleovu funkci $F : \mathbf{2}^n \to \mathbf{2}$, kde n je počet spínačů tohoto obvodu (neboť proměnné funkce F jsou právě spínače $a_1, ..., a_n$ daného spínačového obvodu). Naopak, ke každé Booleově funkci $F(a_1, ..., a_n)$ n proměnných v $\mathbf{2}$ existuje spínačový obvod, který ji realizuje. Protože libovolný spínač se může v obvodu vyskytovat vícekrát, jsou uvažované spínače chápány jako vícecestné.

9.6. Příklad. Určete Booleovu funkci, která je realizována následujícím spínačovým obvodem:

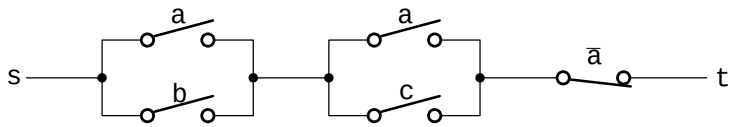


Zřejmě se jedná o funkci $F(a,b,c) = a(b+c) + a$.

9.7. Příklad. Sestavte spínačový obvod, který realizuje Booleovu funkci

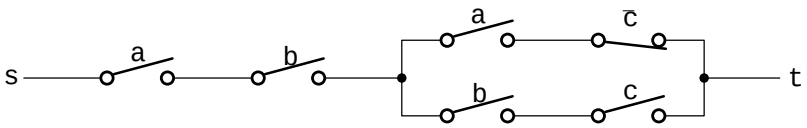
$$F(a,b,c) = (a+b)(a+c)\bar{a}.$$

Řešením je zřejmě následující spínačový obvod:



Dva spínačové obvody se nazývají *ekvivalentní*, jestliže realizují tutéž Booleovu funkci (tj., jestliže oba obvody jsou otevřené, resp. uzavřené při stejné pozici spínačů - nehledě ovšem na spínače, které namají na otevřenost či uzavřenost obvodu vliv).

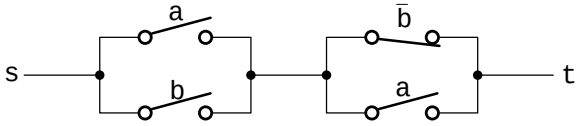
9.8. Příklad. Zjednodušte spínačový obvod



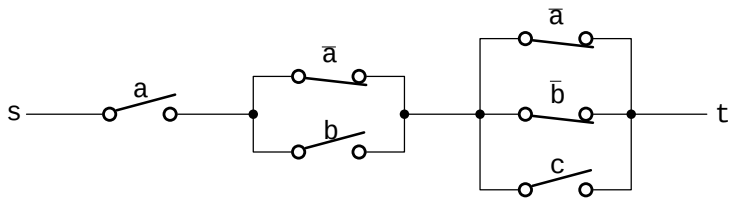
Tento obvod zřejmě realizuje Booleovu funkci $F(a,b,c) = ab(a\bar{c}+bc)$, jejíž úplný disjunktní normální tvar je $F(a,b,c) = ab\bar{c}+abc$. Minimalizací přímou metodou pak snadno dostáváme $F(a,b,c) = ab(\bar{c}+c) = ab$. Příslušným zjednodušením je tedy spínačový obvod

9.9. Cvičení. Zjednodušte následující spínačové obvody:

a)



b)



10. Formální jazyky

Formální jazyky, kterými se budeme v této kapitole zabývat, jsou společným zobecněním jak přirozených tak programovacích jazyků. Zaměříme se na studium takových vlastností formálních jazyků, které jsou významné pro programovací jazyky. Studiem formálních jazyků z hlediska jazyků přirozených se zabývá *matematická lingvistika*.

Buď Σ libovolná konečná množina, tzv. *abeceda*. Symbolem Σ^+ označíme množinu všech konečných neprázdných posloupností utvořených z prvků množiny Σ. Dále klademe $\Sigma^* = \Sigma^+ \cup \{\lambda\}$, kde λ značí prázdnou posloupnost. Posloupnosti $u = (a_1, a_2, ..., a_n) \in \Sigma^+$ budeme stručně zapisovat jako $u = a_1a_2...a_n$ a nazývat *řetězy* nebo *slovy* v abecedě Σ. Číslo n pak nazýváme délkou slova u a značíme $|u|$. Slova délky 1 v abecedě Σ ztotožňujeme s prvky množiny Σ. Také λ je slovo, tzv. *prázdné slovo*, pro něž klademe $|\lambda| = 0$. O množině Σ^* mluvíme jako o množině všech řetězů nad abecedou Σ a o množině Σ^+ jako o množině všech neprázdných řetězů nad Σ.

Na množině Σ^* definujeme binární operaci $\cdot$ následovně: Jestliže $u, v \in \Sigma^*$, $u = a_1...a_n$, $v = b_1...b_m$, pak $u \cdot v = a_1...a_nb_1...b_m$. Jak je obvyklé, symbol $\cdot$ budeme vynechávat, tj. budeme psát uv místo $u \cdot v$. Řetěz uv se nazývá *zřetězením* (nebo *konkatenací*) řetězů u a v. Protože $u\lambda = \lambda u = u$ pro libovolné $u \in \Sigma^*$ a protože operace zřetězení je zřejmě asociativní, je trojice $(\Sigma^*, \cdot, \lambda)$ monoid (tj. $(\Sigma^*, \cdot)$ je pologrupa s jednotkovým prvkem λ). Jak je běžné u pologrup, pro libovolné $u \in \Sigma^*$ a libovolné $k \in \mathbb{Z}^+$ definujeme řetěz u^k vztahem $u^0 = \lambda$ a $u^{i+1} = u^iu$ pro všechna $i = 0, ..., k-1$. Řetěz u^k se nazývá *k-násobné zřetězení* řetězu u. Jsou-li $x, y \in \Sigma^*$ řetězy, pak x se nazývá podřetězem řetězu y, jestliže existují takové řetězy $u, v \in \Sigma^*$, že platí $y = uxv$.

10.1. Definice. Libovolnou podmnožinu $L \subseteq \Sigma^*$ nazýváme *formálním jazykem* (nebo stručněji *jazykem*) nad abecedou Σ.

10.2. Příklady. (1) Každý přirozený jazyk, např. čeština, je jazykem nad abecedou, do níž patří kromě všech písmen užívané abecedy ještě mezera a interpunkční znaménka. Řetězy (slova) v této abecedě jsou všechny gramaticky správně utvořené věty.

(2) Přirozený jazyk lze chápat také jako jazyk nad abecedou, která je tvořena všemi slovními tvary tohoto jazyka, mezerou a interpunkčními znaménky. Řetězy budou opět všechny gramaticky správně utvořené věty.

(3) Každý programovací jazyk je formálním jazykem. Je tvořen právě takovými řetězy, které jsou syntakticky správnými programy v tomto jazyce.

(4) Buď $\Sigma = \{a, b\}$ a nechť $L = \{a^mb^n;\ m \geq 0,\ n \geq 0\}$. Pak L je jazyk nad Σ. Platí např. $\lambda \in L$, $a \in L$, $b \in L$, $a^2b^3 \in L$, $aba \notin L$.

10.3. Definice. *Kontextem* nad abecedou Σ rozumíme libovolnoý prvek $(u, v) \in \Sigma^* \times \Sigma^*$. Je-li L jazyk nad Σ a $x \in \Sigma^*$ řetěz, pak říkáme, že kontext (u, v) *přijímá* x v jazyce L, jestliže $uxv \in L$.

10.4. Příklady. (1) Buď Σ množina, jejímiž prvky jsou všechny slovní tvary českého jazyka, interpunkční znaménka a mezera, kterou označíme symbolem $\smile$. Nechť L je množina všech správně utvořených českých vět. Pak kontext (MÁM$\smile$RÁD$\smile$,.) přijímá v L všechna podstatná jména ve 4.pádě (která mohou být rozvinuta např. přídavnými jmény). Tento kontext tedy např. přijímá řetěz délky 3 DOBRÉ$\smile$PIVO.

(2) Buď $\Sigma = \{a, b\}$ a $L = \{a^m b^n;\ m \geq 0, n \geq 0\}$ a $(u, v) = (a, b)$. Pak zřejmě kontext (u, v) přijímá v L řetěz $x \in \Sigma^*$, právě když $x \in L$.

10.5. Definice. Buď L jazyk nad abecedou Σ a položme $\mathbf{n}(L) = \{x \in \Sigma^*;$ existuje kontext nad Σ, který přijímá x v $L\}$. Množinu $\mathbf{n}(L)$ nazýváme *relací nutnosti* a její prvky *nutnými řetězy* v jazyce L. Řetězy, které nejsou nutné (tj., které jsou prvky množiny $\Sigma^* - \mathbf{n}(L)$), se nazývají *parazitní* v jazyce L.

10.6. Poznámka. Zřejmě tedy $\mathbf{n}(L) = \{x \in \Sigma^*;$ existuje $(u, v) \in \Sigma^* \times \Sigma^*$ tak, že $uxv \in L\}$. Takže pro libovolné slovo $x \in \Sigma^*$ podmínka $x \in \mathbf{n}(L)$ znamená, že x se objevuje v alespoň jednom slově jazyka L. Protože $\mathbf{n}(L) \subseteq \Sigma^*$, je $\mathbf{n}(L)$ unární relací na Σ^*. Vždy platí $L \subseteq \mathbf{n}(L)$, neboť pro každé $x \in L$ máme $x = \lambda x \lambda \in \mathbf{n}(L)$. Je-li $L = \emptyset$, pak také $\mathbf{n}(L) = \emptyset$.

10.7. Příklady. (1) Buď Σ množina, jejímiž prvky jsou všechny české slovní tvary, interpunkční znaménka a mezera $\smile$. Nechť L je množina všech jednoduchých českých oznamovacích vět. Pak $\smile$JEDE$\smile$ZÍTRA$\smile$ je nutným řetězem (délky 5) v L, neboť existuje kontext, např. (FRANTA,DOMŮ.), který jej v L přijímá. Naproti tomu např. řetěz JEDE$\smile$NESPĚCHAJÍ je parazitní v L.

(2) Buď $\Sigma = \{a, b\}$ a $L = \{a^n b a^n;\ n \geq 0\}$. Pak $\mathbf{n}(L) = \{a^m;\ m \geq 0\} \cup \{a^p b a^q;\ p \geq 0, q \geq 0\}$.

10.8. Definice. Buď L jazyk nad abecedou Σ a položme $\mathbf{d}(L) = \{(x, y) \in \Sigma^* \times \Sigma^*;$ pokud nějaký kontext nad Σ přijímá x v L, pak tento kontext přijímá také y v $L\}$. Pak $\mathbf{d}(L)$ nazýváme *relací dominace* jazyka L.

10.9. Poznámka. $\mathbf{d}(L)$ je samozřejmě (binární) relace na Σ^*, pro niž platí $\mathbf{d}(L) = \{(x, y) \in \Sigma^* \times \Sigma^*;$ pro libovolný kontext $(u, v) \in \Sigma^* \times \Sigma^*$ platí implikace $uxv \in L \Rightarrow uyv \in L\}$. Podmínka $(x, y) \in \mathbf{d}(L)$ znamená, že kdykoliv je x podřetězem nějakého řetězu jazyka L, můžeme v tomto řetězu x nahradit řetězem y a vzniklý řetěz bude opět prvkem jazyka L. Jestliže $x \notin \mathbf{n}(L)$, pak pro libovolný řetěz $y \in \Sigma^*$ platí $(x, y) \in \mathbf{d}(L)$, neboť x není podřetězem žádného řetězu jazyka L.

10.10. Věta. Buď L jazyk nad abecedou Σ. Pak relace $\mathbf{d}(L)$ je reflexivní a tranzitivní. Jestliže $(x, y) \in \mathbf{d}(L)$, pak pro libovolné řetězce $u, v \in \Sigma^*$ platí $(uxv, uyv) \in \mathbf{d}(L)$.

Důkaz. Reflexivita relace $\mathbf{d}(L)$ je zřejmá. Je-li $(x,y) \in \mathbf{d}(L)$, $(y,z) \in \mathbf{d}(L)$, pak pro každé $(u,v) \in \Sigma^* \times \Sigma^*$ z podmínky $uxv \in L$ plyne $uyv \in L$ a odtud dále dostáváme $uzv \in L$. Tedy $(x,z) \in \mathbf{d}(L)$, a proto je $\mathbf{d}(L)$ tranzitivní. Je-li $(x,y) \in \mathbf{d}(L)$, $u,v \in \Sigma^*$ a $(w,z) \in \Sigma^* \times \Sigma^*$, pak z podmínky $wuxvz \in L$ plyne $wuyvz \in L$, takže $(uxv, uyv) \in \mathbf{d}(L)$.

10.11. Příklady. (1) Buď Σ množina, jejímiž prvky jsou všechny české slovní tvary, interpunkční znaménka a mezera $\smile$. Nechť L je množina všech správně utvořených českých vět. Buď x řetěz VÍNO (délky 1) a y řetěz DOBRÉ$\smile$PIVO (délky 3). Pak zřejmě $(x,y) \in \mathbf{d}(L)$, neboť v každé české větě lze zřejmě řetěz VÍNO nahradit řetězem DOBRÉ$\smile$PIVO. Ovšem $(y,x) \notin \mathbf{d}(L)$, neboť

VELMI$\smile$DOBRÉ$\smile$PIVO$\smile$SE$\smile$VŠECHNO$\smile$VYPILO.$\in L$,

zatímco VELMI$\smile$VÍNO$\smile$SE$\smile$VŠECHNO$\smile$VYPILO.$\notin L$.

(2) Nechť $\Sigma = \{a,b\}$ a $L = \{a^m b^n;\ m \geq 0, n \geq 0\}$. Pak například $(ab,a) \in \mathbf{d}(L)$ i $(ab,b) \in \mathbf{d}(L)$.

10.12. Definice. Buď L jazyk nad abecedou Σ. Pak klademe $\mathbf{e}(L) = \mathbf{d}(L) \cap \mathbf{d}(L)^{-1}$ a množinu $\mathbf{e}(L)$ nazýváme *relací dvojí dominace* jazyka L.

10.13. Poznámka. Zřejmě je $\mathbf{e}(L)$ ekvivalence na Σ^*. Podmínka $(x,y) \in \mathbf{e}(L)$ znamená, že kdykoliv je jeden z řetězů x, y podřetězem nějakého řetězu jazyka L, lze jej nahradit druhým a vzniklý řetěz bude také patřit do L.

Protože $(\Sigma^* - \mathbf{n}(L)) \times (\Sigma^* - \mathbf{n}(L)) \subseteq \mathbf{e}(L)$, platí následující věta:

10.14. Věta. Množina $\Sigma^* - \mathbf{n}(L)$ je třída ekvivalence $\mathbf{e}(L)$ pro každý jazyk L nad Σ.

10.15. Příklady. (1) V jazyce z příkladu 10.11(1) zřejmě platí

$$(\text{DOBRÉ,VELMI} \smile \text{DOBRÉ}) \in \mathbf{e}(L).$$

(2) Buď $\Sigma = \{a,b\}$ a $L = \{a^n b a^n;\ n \geq 0\}$. Pak $(a^p b a^q, a^{p-r} b a^{q-r}) \in \mathbf{e}(L)$ pro libovolná čísla $p,q,r \in \mathbb{N} \cup \{0\}$ s vlastností $r \leq \min(p,q)$.

10.16. Definice. Buď $(X, \cdot, e)$ libovolný monoid a r relace ekvivalence na X. Pak r se nazývá *kongruence* monoidu $(X, \cdot, e)$, jestliže z podmínek $(x,x') \in r$ a $(y,y') \in r$ plyne $(xy, x'y') \in r$.

10.17. Věta. Buď L jazyk nad abecedou Σ. Pak $\mathbf{e}(L)$ je kongruence na monoidu $(\Sigma^*, \cdot, \lambda)$.

Důkaz. Buďte $(x,x') \in \mathbf{e}(L)$ a $(y,y') \in \mathbf{e}(L)$ libovolné prvky. Nechť $u,v \in \Sigma^*$ a $uxyv \in L$. Pak kontext (u, yv) přijímá x v L, takže přijímá i x' v L (neboť

$(x, x') \in \mathbf{e}(L)$), tj. $ux'yv \in L$. To ovšem znamená, že kontext (ux', v) přijímá y v L, tedy přijímá i y' v L (neboť $(y, y') \in \mathbf{e}(L)$), tj. $ux'y'v \in L$. Tedy $(xy, x'y') \in \mathbf{d}(L)$. Podobně se dokáže, že $(x'y', xy) \in \mathbf{d}(L)$, proto $(xy, x'y') \in \mathbf{e}(L)$.

10.18. Věta. Buď L jazyk nad abecedou Σ. Pak L je sjednocení množiny všech tříd ekvivalence $\mathbf{e}(L)$, které mají s L neprázdný průnik.

Důkaz. Buď C třída ekvivalence $\mathbf{e}(L)$ taková, že $C \cap L \neq \emptyset$. Pak existuje prvek $x \in C \cap L$. Pro každé $y \in C$ ze vztahů $\lambda x \lambda = x \in L$ a $(x, y) \in \mathbf{e}(L)$ plyne $y = \lambda y \lambda \in L$, takže $C \subseteq L$. Tedy sjednocení všech takovýchto tříd C je podmnožinou jazyka L. Protože opačná inkluze je zřejmá, platí tvrzení.

10.19. Věta. Buď L jazyk nad abecedou Σ. Je-li r taková kongruence na monoidu $(\Sigma^*, \cdot, \lambda)$, že L je sjednocením nějaké množiny jejích tříd, pak $r \subseteq \mathbf{e}(L)$.

Důkaz. Nechť $(x, y) \in r$, $(u, v) \in \Sigma^* \times \Sigma^*$ a $uxv \in L$. Pak $(ux, uy) \in r$ a odtud dostáváme $(uxv, uyv) \in r$. Tedy uxv a uyv jsou prvky téže třídy kongruence r. Jelikož platí $uxv \in L$, je tato třída podmnožinou jazyka L, takže $uyv \in L$. Proto $(x, y) \in \mathbf{d}(L)$. Podobně se ukáže, že platí také $(y, x) \in \mathbf{d}(L)$, takže $(x, y) \in \mathbf{e}(L)$. Tím je inkluze $r \subseteq \mathbf{e}(L)$ dokázána.

10.20. Definice. Jazyk L se nazývá *regulární*, jestliže ekvivalence $\mathbf{e}(L)$ má konečný počet tříd.

10.21. Příklady. (1) Buď $L = \{a^m b a^m;\ m \geq 0\}$ jazyk nad abecedou $\Sigma = \{a, b\}$. Pak je zřejmě jednoprvková množina $\{a^m\}$ třída ekvivalence $\mathbf{e}(L)$ pro každé $m \geq 0$, takže L není regulární jazyk.

(2) Buď $L = \{a^m b^n;\ m \geq 0, n \geq 0\}$ jazyk nad abecedou $\Sigma = \{a, b\}$. Pak L je regulární, neboť ekvivalence $\mathbf{e}(L)$ má právě pět tříd: množiny $\Sigma^* - \mathbf{n}(L)$, $\{\lambda\}$, $\{a^m;\ m \geq 1\}$, $\{b^n;\ n \geq 1\}$, $\{a^m b^n;\ m \geq 1, n \geq 1\}$.

10.22. Věta. Každý konečný jazyk je regulární.

Důkaz. Je-li L konečný jazyk nad abecedou Σ, pak i $\mathbf{n}(L)$ je konečná množina. Tedy $\mathbf{e}(L)$ má konečný počet tříd: třídy, které jsou podmnožinami množiny $\mathbf{n}(L)$, a třídu $\Sigma^* - \mathbf{n}(L)$ parazitních řetězů.

10.23. Věta. Buď L jazyk nad abecedou Σ a $a \notin \Sigma$ libovolný prvek. Pak jazyk L je regulární, právě když L je regulární jako jazyk nad abecedou $\Sigma \cup \{a\}$.

Důkaz. Zřejmě řetězy parazitní v jazyce L nad $\Sigma \cup \{a\}$ jsou právě řetězy parazitní v L nad Σ a řetězy obsahující prvek $\{a\}$. Tedy množiny $\mathbf{n}(L)$ jsou v obou případech stejné. Proto i třídy ekvivalence $\mathbf{e}(L)$ tvořené nutnými řetězy jsou stejné. Pokud tedy $\mathbf{n}(L) \neq \Sigma^*$, pak v obou případech existuje jediná další třída ekvivalence $\mathbf{e}(L)$, která je tvořena parazitními řetězy. Takže ekvivalence $\mathbf{e}(L)$ má v obou případech

stejný počet tříd. Pokud ovšem $\mathbf{n}(L) = \Sigma^*$, pak v jazyce L nad Σ jsou všechny třídy ekvivalence $\mathbf{e}(L)$ tvořeny nutnými řetězy, zatímco v jazyce L nad $\Sigma \cup \{a\}$ existuje (jediná) další třída, která je tvořena parazitními řetězy. Takže ekvivalence $\mathbf{e}(L)$ má v případě jazyka L nad $\Sigma \cup \{a\}$ o jednu třídu více než v případě jazyka L nad Σ. Protože ekvivalence $\mathbf{e}(L)$ pro jazyk L nad Σ má konečný počet tříd, máme dokázáno, že i ekvivalence $\mathbf{e}(L)$ pro jazyk L nad $\Sigma \cup \{a\}$ má konečný počet tříd.

10.24. Cvičení. 1. Ukažte, že všechny regulární jazyky nad danou abecedou Σ tvoří vzhledem k uspořádání danému množinovou inkluzí Booleův svaz, tj., že pro libovolné regulární jazyky L_1, L_2 nad Σ jsou také $L_1 \cup L_2$, $L_1 \cap L_2$ a $\Sigma^* - L_1$ regulární jazyky nad Σ ($\emptyset$ a Σ^* jsou samozřejmě regulární jazyky nad Σ).

2. Ukažte, že pro libovolné regulární jazyky L_1, L_2 nad Σ je také $L_1 L_2 = \{xy;\ x \in L_1, y \in L_2\}$ regulární jazyk nad Σ.

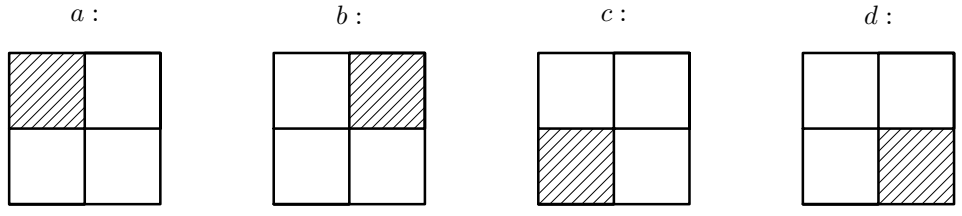
10.25. Poznámka. Regulární jazyky jsou důležité z hlediska informatiky, neboť, jak uvidíme v následující kapitole, to jsou právě ty jazyky, pro něž existují jistá zařízení, která je rozpoznávají. Mezi regulární jazyky patří zejména jednoduché jazyky programovací (na úrovni jazyků symbolických adres). Přirozené jazyky zpravidla nejsou regulární.

11. Konečné automaty

Předmětem zájmu teorie konečných automatů je každý systém, u kterého můžeme vymezit konečně mnoho stavů, do nichž se může tento systém dostat, a konečně mnoho druhů vnějších podnětů působících změny těchto stavů. Přitom je podstatné, aby stav systému a vnější podnět vždy společně jednoznačně určovaly následující stav. V běžném životě se denně setkáváme s různými automaty (např. telefonní automat) i se složitějšími zařízeními, která automaty obsahují (např. počítače). Svého významu dosáhla teorie automatů zejména s rozvojem počítačů, takže dnes tvoří významné odvětví informatiky.

11.1. Definice. *Konečným automatem* (nebo *akceptorem*) rozumíme pětici $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$, kde
S je konečná neprázdná množina, tzv. množina *stavů*,
Σ je konečná neprázdná množina, tzv. *vstupní abeceda*, jejíž prvky nazýváme *vstupní symboly*,
$\delta : S \times \Sigma \to S$ je zobrazení, které se nazývá *přechodová funkce* - tato funkce je podstatou konečného automatu, neboť určuje, do jakého stavu přejde automat ze stavu s, přivedeme-li na vstup symbol a,
$s_0 \in S$ je tzv. *počáteční* stav,
$F \subseteq S$ je množina tzv. *koncových* stavů.

11.2. Příklad. Mějme krabičku s čtvercovým dnem o hraně délky d a v ní tři stejně velké kostky o hranách délky $\frac{d}{2}$. Omezme se na situace, kdy kostky jsou umístěny v rozích krabičky. Přecházet z jednoho stavu do druhého je možné pouze posunem některé kostky na volný čtverec. Vždy lze posunout právě jednu kostku v horizontálním nebo vertikálním směru. Označme horizontální posun písmenem h a vertikální posun písmenem v. Tím dostáváme konečný automat $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$, kde S je množina všech možných umístění kostek v krabici, tedy čtyřprvková množina (kostky od sebe navzájem nerozlišujeme) $S = \{a, b, c, d\}$, jejímiž prvky jsou následující stavy (volné políčko je vyšrafováno):



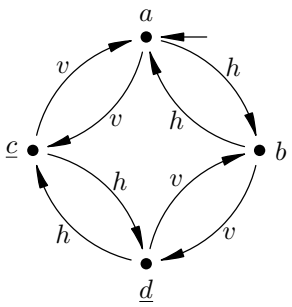
Zřejmě platí $\Sigma = \{h, v\}$, $\delta(a, h) = b$, $\delta(a, v) = c$, $\delta(b, h) = a$, $\delta(b, v) = d$, $\delta(c, h) = d$, $\delta(c, v) = a$, $\delta(d, h) = c$, $\delta(d, v) = b$. Počáteční stav a koncové stavy lze volit libovolně, např. $s_0 = a$ a $F = \{c, d\}$.

Protože všechny uvažované automaty budou konečné, budeme místo konečný automat říkat stručněji automat. Automaty se obvykle zadávají tabulkou nebo stavovým diagramem. *Tabulka* daného automatu je vlastně tabulka jeho přechodové funkce δ. Její řádky jsou nadepsány stavy a sloupce jsou nadepsány vstupními symboly. V řádku odpovídajícím stavu s a sloupci odpovídajícím vstupnímu symbolu a je umístěn stav $\delta(s, a)$. *Stavový diagram* automatu získáme tak, že jeho stavy znázorníme jako body v rovině a ze stavu s vedeme orientovanou hranu (šipku) do stavu t označenou vstupním symbolem a, právě když platí $\delta(s, a) = t$. Abychom vyznačili počáteční stav, vedeme do něj krátkou neohodnocenou šipku, koncové stavy vyznačujeme podtržením.

11.3. Příklady. (1) Uvažujme automat z příkladu 11.2. Pak jeho tabulka má tvar:

δ	h	v
$\rightarrow a$	b	c
b	a	d
$\underline{c}$	d	a
$\underline{d}$	c	b

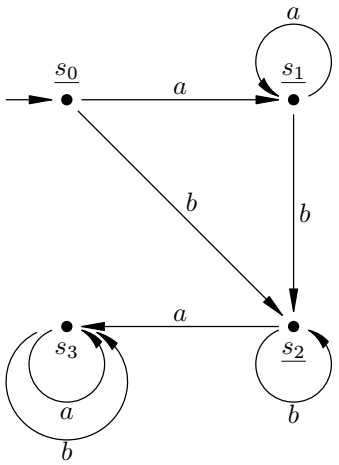
Příslušný stavový diagram je následující:



(2) Uvažujme automat $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$, kde $S = \{s_0, s_1, s_2, s_3\}$ a $\Sigma = \{a, b\}$, který je dán následující tabulkou:

δ	a	b
$\rightarrow \underline{s_0}$	s_1	s_2
$\underline{s_1}$	s_1	s_2
$\underline{s_2}$	s_3	s_2
s_3	s_3	s_3

Stavový diagram tohoto automatu má následující tvar:



11.4. Definice. Buď $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$ automat. Pak definujeme zobrazení $\delta^* : S \times \Sigma^* \to S$ takto:

1. $\delta^*(s, \lambda) = s$ pro každé $s \in S$,
2. $\delta^*(s, wa) = \delta(\delta^*(s, w), a)$ pro každé $s \in S$, $w \in \Sigma^*$ a $a \in \Sigma$.

Zobrazení δ^* se nazývá *zobecněná přechodová funkce* automatu $\mathcal{A}$. Zřejmě pro každý řetězec $w \in \Sigma^*$ délky 1 platí $\delta^*(s, w) = \delta(s, w)$. Takže δ^* je rozšířením δ. Funkce δ^* určuje, do jakého stavu postupně přejde automat ze stavu s, přivedeme-li na jeho vstup slovo w (tj. posloupnost symbolů tvořící toto slovo).

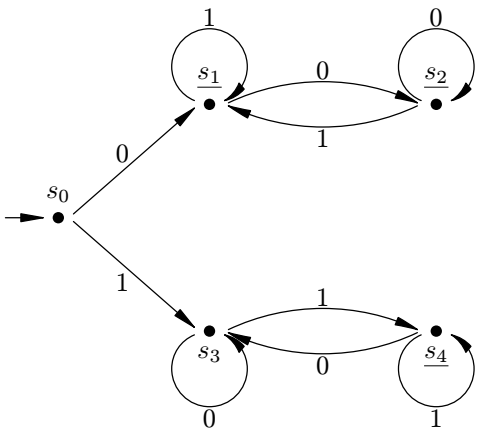
11.5. Definice. Buď $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$ automat. Potom jazyk $L(\mathcal{A}) = \{w \in \Sigma^*;\ \delta^*(s_0, w) \in F\}$ nad abecedou Σ se nazývá jazykem *rozpoznatelným* automatem $\mathcal{A}$.

Následující tzv. *Kleenova věta* udává vztah mezi regulárními jazyky a automaty. Její důkaz svojí náročností přesahuje rámec tohoto učebního textu, proto jej neuvádíme.

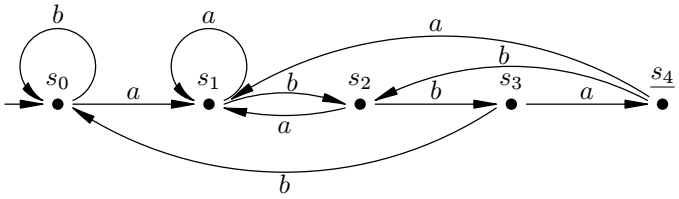
11.6. Věta. Buď L jazyk nad abecedou Σ . Pak L je regulární, právě když je rozpoznatelný nějakým automatem (se vstupní abecedou Σ).

11.7. Příklady. (1) Jazyk $L = \{a^m b^n;\ m \geq 0, n \geq 0\}$ nad abecedou $\Sigma = \{a, b\}$ je regulární, neboť je rozpoznatelný automatem z příkladu 11.3(2).

(2) Buď $\Sigma = \{0, 1\}$ a $L = \{w \in \Sigma^*;\ w$ začíná i končí stejným symbolem a $|w| \geq 2\}$ jazyk nad Σ. Pak L je regulární, neboť je rozpoznatelný automatem s následujícím stavovým diagramem:

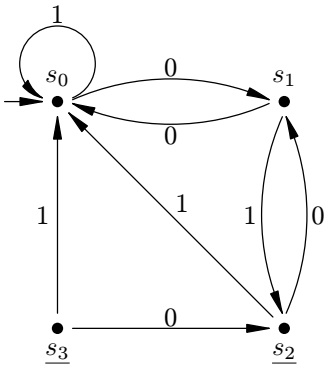


(3) Buď $\Sigma = \{a, b\}$ a $L = \{w \in \Sigma^*;\ w$ končí řetězem $abba\}$ jazyk nad Σ. Pak L je regulární, neboť je rozpoznatelný automatem s tímto stavovým diagramem:



11.8. Definice. Stav $s \in S$ automatu $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$ se nazývá *dosažitelný*, jestliže existuje slovo $w \in \Sigma^*$ takové, že $\delta^*(s_0, w) = s$. V opačném případě se stav s nazývá *nedosažitelný*.

11.9. Příklad. V automatu s následujícím stavovým diagramem jsou stavy s_0, s_1 a s_2 dosažitelné, zatímco stav s_3 je nedosažitelný:



11.10. Poznámka. Daný regulární jazyk je samozřejmě rozpoznatelný nekonečně mnoha automaty. Je-li totiž rozpoznatelný nějakým automatem $\mathcal{A}$, je rozpoznatelný i automatem, který z $\mathcal{A}$ vznikne přidáním libovolného počtu nedosažitelných stavů. Automaty, které rozpoznávají tentýž (regulární) jazyk, se nazývají

ekvivalentní. Problém najít k danému jazyku automat, který by jej rozpoznával, tedy buď nemá řešení (tj. tento jazyk není regulární), nebo má nekonečně mnoho řešení. Při výběru jednoho z těchto řešení se obvykle řídíme požadavkem, aby navržený automat měl co nejmenší počet stavů. Zejména tedy eliminujeme všechny nedosažitelné stavy.

V následující definici zeslabíme pojem kongruence monoidu zavedený v předchozí kapitole.

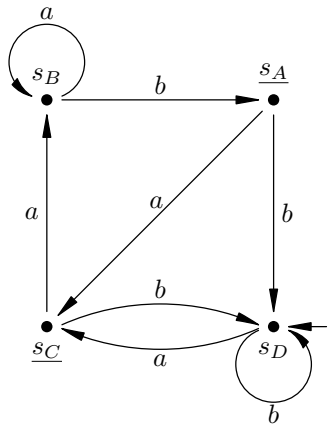
11.11. Definice. Buď $(X, \cdot, e)$ libovolný monoid a r relace ekvivalence na X. Pak r nazýváme *pravou kongruencí* monoidu $(X, \cdot, e)$, jestliže pro libovolné prvky $x, y, z \in X$ z podmínky $(x, y) \in r$ plyne $(xz, yz) \in r$.

Tedy každá kongruence monoidu (viz 10.16) je jeho pravou kongruencí.

Následující věta, která je známa jako *věta Nerodova*, poskytuje užitečné kritérium toho, zda je daný jazyk nad konečnou abecedou rozpoznatelný nějakým automatem (a tedy regulární). Její důkaz z důvodu náročnosti opět vynecháváme.

11.12. Věta. Buď L jazyk nad konečnou abecedou. Pak L je rozpoznatelný nějakým automatem, právě když existuje pravá kongruence monoidu $(\Sigma^*, \cdot, \lambda)$ s konečným počtem tříd taková, že L je sjednocením některých těchto tříd.

11.13. Příklady. (1) Buď $\Sigma = \{a, b\}$ a proveďme rozklad množiny Σ^* na následující množiny:
$A = \{x \in \Sigma^*;\ x \text{ končí řetězem } aab\}$,
$B = \{x \in \Sigma^*;\ x \text{ končí řetězem } aa\}$,
$C = \{x \in \Sigma^*;\ x \text{ obsahuje všechny řetězy končící symbolem } a, \text{ které nepatří do } B\}$,
$D = \Sigma^* - (A \cup B \cup C)$.
Snadno se nahlédne, že ekvivalence odpovídající uvedenému rozkladu je pravá kongruence na monoidu $(\Sigma^*, \cdot, \lambda)$. Položme $L = A \cup C$. Pak L je podle 11.12 rozpoznatelný nějakým automatem. Jedná se např. o automat s následujícím stavovým diagramem:



(2) Buď $L = \{a^{n^2};\ n \geq 0\}$ jazyk nad abecedou $\Sigma = \{a\}$. Ukážeme (sporem), že L není rozpoznatelný žádným automatem.

Předpokládejme, že L je rozpoznatelný nějakým automatem. Pak podle 11.12 existuje pravá kongruence r monoidu $(\Sigma^*, \cdot, \lambda)$ mající konečný počet k tříd taková, že L je sjednocením některých těchto tříd. Proto alespoň dvě z následujících $k+1$ slov leží ve stejné třídě:

$$a^{k^2},\ a^{k^2+1},\ ..., a^{k^2+k}.$$

Jinými slovy,

$$(a^{k^2+i}, a^{k^2+j}) \in r \text{ pro nějaká } i, j,\ 0 \leq i < j \leq k.$$

Protože je r pravá kongruence, dostáváme

$$(a^{k^2+i}a^{2k-j+1}, a^{k^2+j}a^{2k-j+1}) \in r.$$

Přitom ale

$$a^{k^2+j}a^{2k-j+1} = a^{k^2+2k+1} = a^{(k+1)^2} \in L,$$

zatímco

$$k^2 < k^2 + i + 2k - j + 1 < (k+1)^2,$$

takže

$$a^{k^2+i+2k-j+1} \notin L.$$

Označíme-li tedy symbolem A třídu ekvivalence r obsahující řetěz $a^{(k+1)^2}$, máme $A \cap L \neq \emptyset$, avšak nikoliv $A \subseteq L$. To je spor s předpokladem, že L je sjednocením některých tříd ekvivalence r. Proto L není rozpoznatelný žádným automatem.

11.14. Cvičení. Podobně jako v příkladu 11.13(2) ukažte, že jazyk $L = \{a^n b^n;\ n \geq 1\}$ nad abecedou $\Sigma = \{a, b\}$ není rozpoznatelný žádným automatem.

12. Gramatiky

Z 10. kapitoly víme, že jazyk nad danou abecedou je jistá podmnožina množiny všech řetězů nad touto abecedou, tedy je určen právě těmi řetězy, které jej vytvářejí. Některé jazyky lze však *reprezentovat*, tj. charakterizovat i jiným způsobem. V 11. kapitole jsme ukázali, že regulární jazyky je možno reprezentovat pomocí konečných automatů. Nyní ukážeme další způsob reprezentace jazyků, totiž reprezentaci pomocí gramatik. Tato reprezentace je ještě efektivnější než reprezentace pomocí konečných automatů, neboť jazyky, pro které je použitelná, zahrnují i jazyky regulární.

Podle příkladu 10.2 lze chápat přirozený jazyk jako množinu všech gramaticky správně utvořených vět, tedy vět, které jsou utvořeny podle gramatických pravidel tohoto jazyka. Stejný přístup můžeme užít i pro formální jazyky: formální jazyk lze specifikovat jako množinu těch řetězů, které jsou vytvořeny podle jisté gramatiky, tj. soustavy pravidel. Tato gramatika pak příslušný jazyk reprezentuje. Vytváření řetězů podle dané gramatiky se děje postupným přepisováním řetězů v souhlase s jejími pravidly.

12.1. Definice. Buď Σ konečná abeceda. *Přepisovací pravidlo* (stručněji *pravidlo*) je libovolný prvek $(x, y) \in \Sigma^* \times \Sigma^*$. Přepisovací pravidlo (x, y) obvykle zapisujeme ve tvaru $x \to y$. Je-li P konečná množina přepisovacích pravidel, pak dvojici (Σ, P) nazýváme *přepisovací systém*.

Přepisovací pravidla jsou tedy vlastně kontexty - viz 10.3. Avšak přepisovací pravidla budeme používat v jiných souvislostech než kontexty.

12.2. Definice. Buď (Σ, P) přepisovací systém a $u, z \in \Sigma^*$.

a) Je-li $x \to y \in P$, pak řekneme, že u se *přímo přepíše* na z nebo že z se *přímo odvodí* z u (pomocí pravidla $x \to y$), a píšeme $u \Rightarrow z$, jestliže existují řetězy $v, w, x, y \in \Sigma^*$ tak, že $u = vxw$ a $z = vyw$.

b) Řekneme, že u se *přepíše* na z nebo že z se *odvodí* z u, a píšeme $u \Rightarrow^* z$, jestliže existuje posloupnost řetězů $u_0, u_1, ..., u_n \in \Sigma^*$ ($n \in \mathbb{Z}^+$) tak, že platí $u = u_0 \Rightarrow u_1 \Rightarrow ... \Rightarrow u_n = z$. Posloupnost $u_0, u_1, ..., u_n$ se pak nazývá *odvozením* (nebo *derivací*) z z u a n se nazývá *délkou* tohoto odvození.

Všimněme si, že $\Rightarrow$ i $\Rightarrow^*$ jsou relace na Σ^* a že $\Rightarrow^*$ je reflexivní a tranzitivní obal relace $\Rightarrow$. Relace $\Rightarrow^*$ je reflexivní z toho důvodu, že každý řetěz u lze přepsat na sebe sama, jelikož jej můžeme považovat za odvození u z u délky 0.

12.3. Příklady. (1) Buď $\Sigma = \{a, b\}$, $P = \{a \to b^2, ab \to b^3, b \to \lambda\}$. Pak $b^2 \Rightarrow b$, neboť položíme-li $v = x = b$ a $w = y = \lambda$, pak $b^2 = vxw$, $b = vyw$ a $x \to y$. Platí také např. $ba \Rightarrow b^3$, $ba \Rightarrow a$, $aba \Rightarrow b^3a$, $aba \Rightarrow ab^3$, atd. Dále máme např. $a^2b \Rightarrow^* b^5$

(což plyne z odvození a^2b, ab^3, b^5) nebo $a^2b \Rightarrow^* \lambda$ (což plyne z odvození a^2b, ab^3, b^5, b^4, b^3, b^2, b, λ).

(2) Buď $\Sigma = \{a, b\}$, $P = \{a \to ab, ba \to b^3a\}$. Pak $aba \Rightarrow^* ab^5a$, přičemž existují různá odvození řetězu ab^5a z řetězu aba, např. posloupnosti aba, ab^2a, ab^3a, ab^4a, ab^5a a aba, ab^3a, ab^5a.

12.4. Definice. *Gramatikou* rozumíme čtveřici $\mathcal{G} = (\Sigma, T, s_0, P)$, kde

Σ je konečná množina (abeceda),

$T \subseteq \Sigma$ je podmnožina, jejíž symboly se nazývají *terminály*; symboly množiny $\Sigma - T$ se nazývají *neterminály*,

$s_0 \in \Sigma - T$ je tzv. *počáteční symbol*

a (Σ, P) je přepisovací systém takový, že pro každé pravidlo $x \to y$ z P řetěz x obsahuje alespoň jeden neterminál.

12.5. Definice. Buď $\mathcal{G} = (\Sigma, T, s_0, P)$ gramatika. Pak jazyk

$$L(\mathcal{G}) = \{w \in T^*;\ s_0 \Rightarrow^* w\}$$

se nazývá *jazyk generovaný gramatikou* $\mathcal{G}$.

Jazyk generovaný danou gramatikou je tedy tvořen všemi řetězy skládajícími se jen z terminálů, které lze odvodit z počátečního symbolu.

12.6. Příklad. Buď $\mathcal{G} = (\Sigma, T, s_0, P)$ gramatika, kde $\Sigma = \{a, b, s, t\}$, $T = \{a, b\}$, $s_0 = s$ a $P = \{s \to ata,\ t \to ata,\ t \to b\}$. Ukážeme, že platí $L(\mathcal{G}) = \{a^nba^n\ \ n \geq 1\}$.

Buď $n \geq 1$ a položme $u_0 = s$, $u_i = a^ita^i$ pro každé i s vlastností $1 \leq i \leq n$, $u_{n+1} = a^nba^n$. Zřejmě je $u_0, u_1, ..., u_{n+1}$ odvození řetězu a^nba^n z řetězu s, takže $a^nba^n \in L(\mathcal{G})$. Platí tedy $\{a^nba^n\ \ n \geq 1\} \subseteq L(\mathcal{G})$.

Nechť $w \in L(\mathcal{G})$. Pak $w \in T^*$ a $s \Rightarrow^* w$, tj. existuje odvození $u_0, u_1, ..., u_n$ řetězu w z s. Protože $s \to ata$ je jediné pravidlo z P, které má na levé straně s, je $u_1 = ata$. Protože u_1 obsahuje neterminál t, je $n > 1$. Protože $t \to b$ je jediné pravidlo z P, které nemá na pravé straně ani jeden neterminál, musí se u_n přímo odvodit z u_{n-1} pomocí tohoto pravidla. Proto každý z řetězů $u_1, ..., u_{n-1}$ obsahuje alespoň jeden neterminál. Pokud by se totiž některý z nich skládal ze samých terminálů, nedal by se z něho přímo odvodit žádný řetěz, neboť všechna pravidla z P mají na levých stranách jen neterminály. To by byl spor, neboť z každého z řetězů $u_1, ..., u_{n-1}$ lze přímo odvodit řetěz následující. Matematickou indukcí nyní ukážeme, že pro každé $i \in \{1, ..., n-1\}$ platí rovnost $u_i = a^ita^i$. Víme již, že pro $i = 1$ tato rovnost platí. Budeme předpokládat, že platí pro nějaké $i \in \{1, ..., n-2\}$. Protože $i + 1 \leq n - 1$, obsahuje u_{i+1} alespoň jeden neterminál. Jelikož t je jediný neterminál obsažený v u_i, u_{i+1} se přímo odvodí z u_i pomocí pravidla $t \to ata$ nebo pomocí pravidla $t \to b$. V případě pravidla $t \to b$ bychom měli $u_{i+1} = a^iba^i$, takže řetěz u_{i+1} by neobsahoval

žádný neterminál. Tedy by se z něj nedal přímo odvodit žádný řetěz, což je spor (neboť z u_{i+1} se přímo odvodí u_{i+2}). Proto se u_{i+1} přímo odvodí z u_i pomocí pravidla $t \to ata$, takže $u_{i+1} = a^{i+1}ta^{i+1}$. Ukázali jsme, že pro každé $i \in \{1, ..., n-1\}$ platí $u_i = a^i t a^i$. Zejména tedy máme $u_{n-1} = a^{n-1}ta^{n-1}$. Protože $w = u_n$ se odvodí z u_{n-1} pomocí pravidla $t \to b$, dostáváme $w = a^{n-1}ba^{n-1}$. Takže jsme dospěli k závěru, že pro každé $w \in L(\mathcal{G})$ existuje $n \geq 2$ tak, že $w = a^{n-1}ba^{n-1}$. Proto $L(\mathcal{G}) \subseteq \{a^n b a^n;\ n \geq 1\}$. Celkem tedy máme $L(\mathcal{G}) = \{a^n b a^n;\ n \geq 1\}$.

Různé speciální gramatiky lze definovat předepsáním vlastností jejich přepisovacích pravidel. V následující definici uvedeme základní typy gramatik, které budou seřazeny podle tzv. *Chomského hierarchie*.

12.7. Definice. (i) Gramatiku G v obecném tvaru, jak byla definována v 12.4, nazýváme *gramatikou typu 0*.

(ii) Gramatika $\mathcal{G} = (\Sigma, T, s_0, P)$ se nazývá *gramatika typu 1*, jestliže každé přepisovací pravidlo z P je tvaru $uav \to uwv$, kde $u, v \in \Sigma^*$, $a \in \Sigma - T$ a $w \in \Sigma^+$. Jedinou výjimkou může být pravidlo $s_0 \to \lambda$, při jehož výskytu se pak s_0 nesmí objevit na pravé straně žádného přepisovacího pravidla.

(iii) Jestliže gramatika $\mathcal{G} = (\Sigma, T, s_0, P)$ má tu vlastnost, že P obsahuje jen pravidla tvaru $a \to w$, kde $a \in \Sigma - T$ a $w \in \Sigma^*$, pak se nazývá *gramatikou typu 2*.

(iv) Gramatika $\mathcal{G} = (\Sigma, T, s_0, P)$ se nazývá *gramatika typu 3*, jestliže každé pravidlo z P má tvar $a \to wb$ nebo $a \to w$, kde $a, b \in \Sigma - T$ a $w \in T^*$.

12.8. Poznámka. Použití tzv. *kontextového* pravidla $uav \to uwv$ v definici gramatik typu 1 i tzv. *bezkontextového* pravidla $a \to w$ v definici gramatik typu 2 má stejný efekt, totiž neterminál a se přepíše na w a ostatní části řetězu zůstanou nezměněny. V prvním případě však k tomu dojde jen tehdy, jestliže existuje kontext (u, v), v němž se a vyskytuje. Ve druhém případě přepsání a na w na žádném kontextu nezávisí. Proto se také gramatiky typu 1 i jimi generované jazyky nazývají *kontextové*, zatímco gramatiky typu 2 i jimi generované jazyky se nazývají *bezkontextové*.

Chomského hierarchie se přenáší také na jazyky:

12.9. Definice. Buď $i \in \{0, 1, 2, 3\}$. Jazyk generovaný gramatikou typu i se nazývá *jazyk typu i*.

12.10. Věta. Nechť $i \in \{1, 2, 3\}$. Pak každý jazyk typu i je také jazykem typu $i - 1$.

Důkaz. Jestliže $i = 1$ nebo $i = 3$, pak tvrzení plyne přímo z definice 12.7, neboť každá gramatika typu 1 je také typu 0 a každá gramatika typu 3 je také typu 2. Ovšem gramatika typu 2 nemusí být typu 1, protože může obsahovat větší počet pravidel typu $a \to \lambda$, zatímco gramatika typu 1 může obsahovat pouze jediné pravidlo s prázdným řetězem na pravé straně, a to $s_0 \to \lambda$, přičemž s_0 se pak

nesmí vyskytovat na pravé straně žádného pravidla. Dá se však dokázat, že ke každé gramatice typu 2 existuje gramatika typu 1, která generuje tentýž jazyk. Tento důkaz je ale poněkud náročnější, a proto jej neuvádíme.

Pro jazyky typu 3 platí následující důležité tvrzení:

12.11. Věta. Jazyk je typu 3, právě když je regulární.

Důkaz. Nechť L je regulární jazyk. Pak podle věty 11.6 existuje konečný automat $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$, který L rozpoznává. Definujme gramatiku $\mathcal{G} = (S \cup \Sigma, \Sigma, s_0, P)$ takto:

$P = \{s \to as';\ \delta(s,a) = s',\ s, s' \in S,\ a \in \Sigma\} \cup \{s \to \lambda;\ s \in F\}$.

$\mathcal{G}$ je zřejmě gramatika typu 3 taková, že pro libovolný řetěz $a_1...a_n \in \Sigma^+$ a libovolný stav $s \in S$ platí $s_0 \Rightarrow^* a_1...a_n s$, právě když automat $\mathcal{A}$ přejde ze stavu s_0 po přivedení řetězu $a_1, ..., a_n$ na vstup do stavu s. Takže máme (viz 11.5)

$a_1...a_n \in L(\mathcal{A}) \Leftrightarrow$ existuje $s \in F$ s vlastností $s_0 \Rightarrow^* a_1...a_n s \Leftrightarrow s_0 \Rightarrow^* a_1...a_n \Leftrightarrow a_1...a_n \in L(\mathcal{G})$.

Proto pro každý neprázdný řetěz w v Σ platí $w \in L(\mathcal{A}) \Leftrightarrow w \in L(\mathcal{G})$.

Pro prázdný řetěz dostáváme

$\lambda \in L(\mathcal{A}) \Leftrightarrow s_o \in F \Leftrightarrow s_o \to \lambda \in P \Leftrightarrow \lambda \in L(\mathcal{G})$.

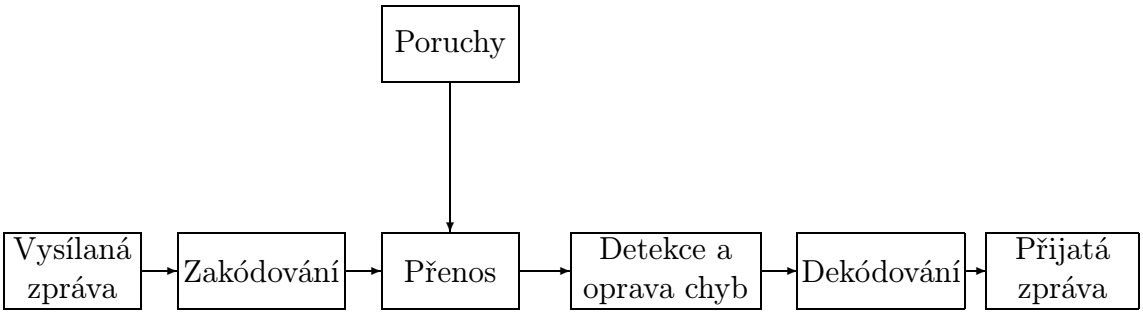
Úhrnem tedy máme $L(\mathcal{G}) = L(\mathcal{A})$.

Zbývá dokázat, že k libovolnému jazyku L typu 3 existuje konečný automat, který L rozpoznává (L je pak regulární podle 11.6). Tato část důkazu je však poněkud komplikovanější, proto ji vynecháváme.

12.12. Cvičení. Buď $\mathcal{G} = (\Sigma, T, s_0, P)$ gramatika, kde $\Sigma = \{a, b, s, t, r\}$, $T = \{a, b\}$, $s_0 = s$ a $P = \{s \to at,\ t \to ar,\ r \to bt,\ r \to b\}$. Ověřte, že $\mathcal{G}$ je typu 3 a že generuje regulární jazyk $\{a(ab)^m;\ m \geq 1\}$ (připomeňme, že $(ab)^m$ značí m-násobné zřetězení řetězu ab, tedy řetěz $ababab....ab$, v němž se a i b vyskytuje právě m-krát). Nakreslete stavový diagram konečného automatu, který tento jazyk rozpoznává.

13. Samoopravné kódy

Při předávání zpráv z jednoho místa na druhé je často z důvodu snadnějšího přenosu vhodné zprávy zakódovat. Jakmile vyslaná zakódovaná zpráva dorazí na místo určení, je nutno ji dekódovat do původní podoby. Studiem vhodných metod kódování se zabývá matematická disciplína, která se nazývá *teorie kódování*. Při hledání optimálních kódů jsou základními kritérii jednoduchost, možnost jendoznačného dekódování a odolnost proti poruchám, které mohou vzniknout při přenosu. V této kapitole se zaměříme především na kódy odolné proti přenosovým poruchám. Všimneme si podrobněji takových kódů, které umožňují přenosové chyby nejen zjišťovat, ale také opravovat. Takovéto kódy se nazývají *samoopravné* a předávání zpráv při jejich užití probíhá dle následujícího schématu:



Je-li např. telexová zpráva přenášena na velkou vzdálenost, může dojít k různým interferencím (způsobeným letadly, bouřkami, apod.), které způsobí chyby, takže přijatá zpráva není shodná se zprávou vyslanou. Je tedy třeba vzniklé chyby najít a opravit. Přirozený jazyk poskytuje prostředky pro takovouto opravu, neboť některá slova (posloupnosti písmen) nedávají smysl. Je-li přijatá zpráva MÁM RÁF IVU, zřejmě došlo k chybě a my víme, jak tuto chybu opravit: na tvar MÁM RÁD IVU. To však nelze provést vždy. Je-li totiž přijatá zpráva MÁM RÁD KVU, pak jsme sice přesvědčeni, že došlo k chybě, ale nevíme, jak ji opravit na správný tvar (neboť vyslaná zpráva mohla být jak MÁM RÁD IVU tak MÁM RÁD EVU).

Systematický přístup k tomuto problému poskytuje užití samoopravných kódů, které reprezentují zprávy vybranými *binárními slovy*, tj. slovy abecedy $\{0, 1\}$. Pak je totiž každá chyba jen vzájemnou záměnou 0 a 1. Vybraná binární slava odpovídají skutečným zprávám podle dohodnutých pravidel, která jsou známa jak odesílateli tak příjemci zprávy. Odesílatel zakóduje zprávu tak, že jí přiřadí příslušné binární slovo. Zakódovaná verze je pak přenášena kanálem, přičemž na ni působí interference, které ji poruší. Na příjmu jsou chyby v porušené (zakódované) zprávě nalezeny a opraveny. Pak je zpráva dekódována užitím dohodnutých pravidel, čímž obdržíme původní vyslanou zprávu. Zdůrazněme, že utajení obsahu zprávy není cílem našeho kódovacího procesu (teorií kódování zpráv za účelem jejich utajení při přenosu se

zabývá matematická disciplína nazývaná *kryptografie*).

Při kódování zprávy je vhodné se omezit na slova stejné délky n. Množinu všech binárních slov délky n označíme symbolem $V(n)$. Tato množina má zřejmě 2^n prvků, např.

$$V(3) = \{000, 100, 010, 001, 110, 101, 011, 111\}.$$

Každý symbol v binárním slovu nazýváme *bit* (což je zkratka anglického výrazu *binary digit*).

13.1. Definice. *Binárním kódem* rozumíme libovolnou neprázdnou podmnožinu $C \subseteq V(n)$. Číslo n se nazývá *délka* binárního kódu C a prvky množiny C se nazývají *kódová slova* binárního kódu C.

Zřejmě tedy je možno každý binární kód chápat jako jazyk nad abecedou $\{0, 1\}$ (nebo jako n-ární relaci na množině $\{0, 1\}$, kde n je délka tohoto kódu - viz 3.17).

13.2. Příklad. Předpokládejme, že chceme poslat zprávy *nahoru, dolů, doleva, doprava*. Můžeme užít následující tři kódy:

kód	délka	*nahoru*	*dolů*	*doleva*	*doprava*
C_1	2	00	10	01	11
C_2	3	000	110	011	101
C_3	6	000000	111000	001110	110011

Příjemce zpráv samozřejmě zná kód, který byl užit, i způsob zakódování (tj. přiřazení mezi zprávami a kódovými slovy).

Kód C_1 je velice ekonomický, ale neposkytuje prostředek pro detekci chyb. Je-li totiž odesláno kódové slovo 00 a dojde k přenosové chybě v prvním bitu, pak přijaté slovo je 10. Ale toto slovo je také kódové, takže příjemce není schopen zjistit chybu.

Kód C_2 je lepší, neboť umožnuje nalezení každé chyby, pokud k ní došlo pouze v jediném bitu vyslaného kódového slova. Přijaté slovo pak totiž není kódové, takže příjemce ví, že došlo k chybě. Tuto chybu ale není možno vhodným způsobem opravit. Je-li například odesláno kódové slovo 000 a dojde k chybě v prvním bitu, přijaté slovo 100 může být také kódové slovo 110 s chybou v druhém bitu nebo kódové slovo 101 s chybou v třetím bitu. Tento kód tedy umožnuje chybu najít, ale neumožnuje ji opravit. V praxi je proto použitelný jen v případě, kdy je možno požádat o opakované zaslání zprávy, v níž byla nalezena chyba.

Kód C_3 je samozřejmě komplikovanější než předchozí dva, ale umožňuje chyby nejen nalézt, nýbrž je i opravit. Předpokládáme-li totiž, že při přenosu došlo pouze k jediné chybě, pak umíme určit, které kódové slovo bylo odesláno. Je-li např. přijato slovo 110000, pak jediné kódové slovo, které můžeme obdržet změnou jednoho bitu, je 111000. Samozřejmě, k chybě může dojít ve dvou nebo více bitech vyslaného

kódového slova. To je však v praxi mnohem méně pravděpodobné než výskyt právě jedné chyby.

Formalizujme nyní úvahy z předchozího příkladu. Binární slova budeme označovat tučnými písmeny **a**, **b**, **c**, atd. a zavedeme pro ně následující pojem:

13.3. Definice. Jsou-li **a** a **b** binární slova téže délky, pak jejich *vzdálenost* $d(\mathbf{a}, \mathbf{b})$ definujeme jako počet bitů, v nichž se **a** liší od **b**.

13.4. Příklady. Zřejmě platí
(1) $d(1101, 1000) = 2$,
(2) $d(1010101, 1100100) = 3$.

13.5. Poznámka. Snadno se ověří, že vzdálenost binárních slov je *metrika*, tj., že jsou splněny následující tři axiomy:
(i) $d(\mathbf{a}, \mathbf{b}) = 0 \Leftrightarrow \mathbf{a} = \mathbf{b}$,
(ii) $d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a})$,
(iii) $d(\mathbf{a}, \mathbf{b}) \leq d(\mathbf{a}, \mathbf{c}) + d(\mathbf{c}, \mathbf{b})$.

Buď C binární kód. Symbolem δ označíme tzv. *minimální vzdálenost kódu* C, tj. minimální vzdálenost mezi všemi různými dvojicemi kódových slov kódu C:
$\delta = \min\{d(\mathbf{a}, \mathbf{b}); \mathbf{a}, \mathbf{b} \in C,\ \mathbf{a} \neq \mathbf{b}\}$.

13.6. Příklad. Pro kódy C_1, C_2 a C_3 z příkladu 13.2 má δ zřejmě hodnotu 1, 2 a 3.

Vzdálenost mezi dvěma slovy daného binárního kódu vyjadřuje počet chyb, ke kterému musí dojít, aby se jedno z těchto slov transformovalo v druhé. Je-li tedy minimální vzdálenost binárního kódu δ a při přenosu jistého jeho slova došlo k chybám, kterých není více než $\delta - 1$, pak příjemce ví, že přijaté slovo není kódové. Došlo-li k právě δ chybám, vyslané slovo může být přijato jako jiné kódové slovo. Lze tedy konstatovat, že binární kód s minimální vzdáleností δ umí zjistit chyby, pokud jich není více než $\delta - 1$.

Samozřejmě, nestačí chyby jen zjistit, ale je třeba je též opravit. Jestliže binární kód C umožňuje určit vyslané slovo ze slova přijatého a ze znalosti počtu e chyb, ke kterým došlo (tj. počtu e bitů vyslaného slova, ve kterých došlo k chybám) při přenosu, pak říkáme, že C *opravuje* e *chyb*. Nejčastěji se opravy provádí způsobem, který byl diskutován v příkladu 13.2 pro kód C_3: je-li přijato slovo s chybami, předpokládáme, že bylo vysláno kódové slovo, které je mu nejblíže (ve smyslu vzdálenosti d). Přitom je samozřejmě nutné, aby takové slovo bylo jediné. Hovoříme pak o tzv. kódovacím principu *nejbližšího souseda*. Následující věta udává vztah mezi minimální vzdáleností δ a počtem chyb, které mohou být opraveny při užití tohoto principu.

13.7. Věta. Binární kód C opravuje e chyb principem nejbližšího souseda, jestliže jeho minimální vzdálenost δ splňuje podmínku

$$\delta \geq 2e + 1.$$

Důkaz. Buď C binární kód, pro jehož minimální vzdálenost δ platí $\delta \geq 2e + 1$. Předpokládejme, že bylo odesláno kódové slovo $\mathbf{c}$ a během přenosu v něm došlo k e chybám, takže přijato bylo slovo $\mathbf{z}$. Tedy máme $d(\mathbf{c}, \mathbf{z}) = e$. Buď $\mathbf{x}$ jakékoliv jiné kódové slovo kódu C. Pak platí

$$d(\mathbf{c}, \mathbf{z}) + d(\mathbf{z}, \mathbf{x}) \geq d(\mathbf{c}, \mathbf{x}) \geq \delta \geq 2e + 1.$$

Odtud vyplývá, že $e + d(\mathbf{z}, \mathbf{x}) \geq 2e + 1$, takže

$$d(\mathbf{z}, \mathbf{x}) \geq e + 1.$$

Ukázali jsme, že $\mathbf{c}$ je jediné slovo kódu C, jehož vzdálenost od $\mathbf{z}$ je e. Proto princip nejbližšího souseda opravuje všech e chyb ve slově $\mathbf{z}$ a dává správné slovo $\mathbf{c}$.

13.8. Cvičení. (1) Dokažte (tzv. trojúhelníkovou) nerovnost (iii) z poznámky 13.5.

(2) Určete minimální vzdálenost δ pro každý z následujících tří binárních kódů:
a) $\{0000, 1100, 1010, 1001, 0110, 0101, 0011, 1111\} \subseteq V(4)$,
b) $\{10000, 01010, 00001\} \subseteq V(5)$,
c) $\{000000, 101010, 010101\} \subseteq V(6)$.
Pro každý z těchto kódů určete počet chyb, které mohou být nalezeny a opraveny.

(3) Vytvořte binární kód pro pět zpráv, který opravuje jednu chybu (principem nejbližšího souseda).

Množinu $V(n)$ všech binárních slov délky n můžeme algebraicky strukturovat několika způsoby. Nejjednodušším způsobem je vytvořit z $V(n)$ komutativní grupu tak, že pro libovolná dvě slova $\mathbf{x}$, $\mathbf{y} \in V(n)$ definujeme jejich součet $\mathbf{x}+\mathbf{y}$ tak, že sečteme odpovídající si bity modulo 2 (viz 4.12), což znamená, že sčítáme následovně:

$$0 + 0 = 0,\ 0 + 1 = 1 + 0 = 1,\ 1 + 1 = 0.$$

Máme tedy například

$$1011001 + 1000111 = 0011110.$$

Je snadné ověřit, že množina $V(n)$ s takto definovanou operací sčítání je komutativní grupa (jejímž nulovým prvkem je slovo $\mathbf{0}$, jehož všechny bity mají hodnotu 0).

13.9. Definice. Binární kód $C \subseteq V(n)$ se nazývá *lineární*, jestliže z podmínky $\mathbf{x},\mathbf{y} \in C$ plyne $\mathbf{x}+\mathbf{y} \in C$.

13.10. Příklad. Kódy C_1 a C_2 z příkladu 13.2 jsou lineární, zatímco kód C_3 lineární není, neboť 111000 a 110011 patří do C_3, zatímco 111000+110011=001011 nepatří do C_3.

Jelikož je grupa $V(n)$ konečná, z definice 13.9 ihned vyplývá, že binární kód $C \subseteq V(n)$ je lineární, právě když C je podgrupa grupy $V(n)$. Tedy podle známé Lagrangeovy věty mohutnost card C libovolného lineárního kódu C délky n dělí mohutnost card $V(n) = 2^n$. Proto platí card $C = 2^k$, kde k je celé číslo s vlastností $0 \leq k \leq n$. Číslo k se nazývá *dimenze* lineárního kódu C.

13.11. Poznámka. Samozřejmě, grupa $V(n)$ je lineární prostor nad $V = V(1) = \{0, 1\}$, přičemž na V je kromě součtu modulo 2 definován také součin modulo 2, který je totožný s logickým součinem (viz kapitolu 9), tj.

$$0 \cdot 0 = 0,\ 0 \cdot 1 = 1 \cdot 0 = 0,\ 1 \cdot 1 = 1$$

(zřejmě 0 je nulovým a 1 jednotkévým prvkem ve V). Tedy binární kód délky n je lineární, právě když C je lineárním podprostorem lineárního prostoru $V(n)$. Lze snadno ukázat, že dimenze lineárního kódu C délky n je totožná s dimenzí tohoto kódu chápaného jako lineární podprostor prostoru $V(n)$.

Nyní máme definovány tři parametry, které určují praktickou použitelnost zvoleného lineárního kódu: jeho délku n, dimenzi k a minimální vzdálenost δ. Přejeme si, aby k bylo dost velké, což umožní poslat značné množství zpráv. Je-li ale k příliš velké, tj. obsahuje-li kód C velmi mnoho slov, pak jeho minimální vzdálenost bude malá, což má za následek, že může být opraveno jen málo chyb. Je proto třeba volit dimenzi navrhovaného kódu v závislosti na charakteru vysílaných zpráv.

13.12. Příklad. Buď C lineární kód délky n a dimenze k a nechť e je maximální počet chyb, které C opravuje. Buď $\mathbf{c} \in C$ libovolné kódové slovo délky n. Pak počet slov, která můžeme obdržet z $\mathbf{c}$ změnou právě r bitů ($r \leq n$) je $\begin{pmatrix} n \\ r \end{pmatrix}$, protože můžeme vybrat libovolnou r-tici. Nechť $S_e(\mathbf{c})$ označuje množinu slov, která můžeme obdržet změnou nejvýše e bitů v $\mathbf{c}$. Pak

$$\text{card } S_e(\mathbf{c}) = 1 + \begin{pmatrix} n \\ 1 \end{pmatrix} + \begin{pmatrix} n \\ 2 \end{pmatrix} + ... + \begin{pmatrix} n \\ e \end{pmatrix}.$$

Jestliže C opravuje nejvýše e chyb, pak množiny $S_e(\mathbf{c})$ a $S_e(\mathbf{d})$ musí být disjunktní, jakmile $\mathbf{c}$ a $\mathbf{d}$ jsou různá kódová slova kódu C. Takže $V(n)$ obsahuje card $C = 2^k$ po dvou disjunktních podmnožin mohutnosti card $S_e(\mathbf{c})$. Proto

$$2^n \geq 2^k \text{card } S_e(\mathbf{c})$$

a odtud dostáváme

$$2^{n-k} \geq 1 + \left(\begin{array}{c} n \\ 1 \end{array} \right) + \left(\begin{array}{c} n \\ 2 \end{array} \right) + ... + \left(\begin{array}{c} n \\ e \end{array} \right).$$

Nechť například $n = 17$, $k = 10$ a $e = 2$. Pak máme

$$1 + \left(\begin{array}{c} n \\ 1 \end{array} \right) + \left(\begin{array}{c} n \\ 2 \end{array} \right) = 1 + 17 + 136 = 154.$$

Protože $154 > 2^{17-10} = 128$, žádný lineární kód délky $k = 17$ a dimenze $n = 10$ neopraví více než jednu chybu.

Velkou předností lineárních kódů je poměrně snadný způsob určení jejich minimální vzdálenosti. Je to důsledkem skutečnosti, že vzdálenost dvou kódových slov libovolného lineárního kódu se nezmění, pokud k oběma z nich přičteme totéž kódové slovo, tj.

$$d(\mathbf{x} + \mathbf{a}, \mathbf{y} + \mathbf{a}) = d(\mathbf{x}, \mathbf{y}) \quad (\mathbf{x}, \mathbf{y}, \mathbf{a} \in V(n)).$$

13.13. Definice. *Váhou* $w(\mathbf{a})$ binárního slova $\mathbf{a}$ rozumíme počet jedniček v $\mathbf{a}$.

Pro libovolné binární slovo $\mathbf{a}$ zřejmě platí $w(\mathbf{a}) = d(\mathbf{a}, \mathbf{0})$ a pro libovolná binární slova $\mathbf{x}$, $\mathbf{y}$ téže délky máme

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x} - \mathbf{y}, \mathbf{y} - \mathbf{y}) = d(\mathbf{x} - \mathbf{y}, \mathbf{0}) = w(\mathbf{x} - \mathbf{y}).$$

Poznamenejme také, že pro libovolná binární slova $\mathbf{x}$, $\mathbf{y}$ téže délky zřejmě platí $\mathbf{x}$-$\mathbf{y}$=$\mathbf{x}$+$\mathbf{y}$.

Pro daný lineární kód C označíme symbolem w_{min} minimální délku jeho nenulových kódových slov, tj.

$$w_{mim} = min\{w(\mathbf{a});\ \mathbf{a} \in C - \{\mathbf{0}\}\}.$$

13.14. Věta. Pro minimální vzdálenost δ libovolného lineárního kódu platí

$$\delta = w_{min}.$$

Důkaz. Buď C libovolný lineární kód a nechť $\mathbf{c}$ je jeho kódové slovo takové, že platí $w(\mathbf{c}) = w_{min}$. Pak máme

$$\delta \leq d(\mathbf{c}, \mathbf{0}) = w(\mathbf{c}) = w_{min}.$$

Buďte $\mathbf{a}$ a $\mathbf{b}$ kódová slova kódu C mající minimální vzdálenost. Pak

$$\delta = d(\mathbf{a}, \mathbf{b}) = w(\mathbf{a} - \mathbf{b}) \geq w_{min}.$$

Proto $\delta = w_{min}$.

13.15. Cvičení. (1) Určete dimenzi k a minimální vzdálenost δ lineárního kódu (délky n) $C = \{00...0, 11...1\}$.

(2) Určete maximální dimenzi lineárního kódu délky 8, který opravuje dvě chyby. Sestrojte takový kód.

(3) Dokažte, že váha všech kódových slov lineárního kódu nebo právě poloviny z nich je sudé číslo.

Pro libovolné binární slovo $\mathbf{a} \in V(n)$ označíme symbolem $\mathbf{a}'$ toto slovo chápané jako sloupcový vektor.

13.16. Věta. Buď H libovolná matice tvořená nulami a jedničkami a mající n sloupců. Pak množina $C_H = \{\mathbf{a} \in V(n); H\mathbf{a}' = \mathbf{0}'\}$ je lineární kód délky n.

Důkaz. C je zřejmě binární kód délky n. Jestliže $\mathbf{a}, \mathbf{b} \in C$, pak $H(\mathbf{a} + \mathbf{b})' = H\mathbf{a}' + H\mathbf{b}' = \mathbf{0}'$. Tedy $\mathbf{a} + \mathbf{b} \in C$, takže C je lineární.

Matice H se nazývá *kontrolní matice* lineárního kódu C_H.

13.17. Příklad. Určeme všechna kódová slova kódu C_H, který je dán kontrolní maticí

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Kód C_H je tvořen všemi binárními slovy tvaru $\mathbf{a} = x_1x_2x_3x_4$, pro která platí

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

tj.

$x_1 + x_3 = 0$

$x_2 + x_3 + x_4 = 0.$

Tento systém dvou lineárních rovnic lze přepsat do následujícího tvaru (uvědomíme-li si, že sčítáme modulo 2):

$x_1 = x_3$

$x_2 = x_3 + x_4.$

Tedy hodnoty bitů x_3 a x_4 můžeme volit libovolně a hodnoty bitů x_1 a x_2 jsou pak jednoznačně určeny. Takže máme

$$C_H = \{0000, 0101, 1011, 1110\}.$$

Metoda popsaná v příkladu 13.17 se užívá ke konstrukci lineárního kódu s předepsanou dimenzí k a délkou n. Při této konstrukci uvažujeme následující horní trojúhelníkovou kontrolní matici H s $r = n - k$ řádky (a n sloupci):

$$\begin{pmatrix} 1 & 0 & 0 & ... & 0 & 0 & b_{11} & b_{12} & ... & b_{1\ n-r} \\ 0 & 1 & 0 & ... & 0 & 0 & b_{21} & b_{22} & ... & b_{2\ n-r} \\ . & & & & & & & & & \\ . & & & & & & & & & \\ . & & & & & & & & & \\ 0 & 0 & 0 & ... & 0 & 1 & b_{r1} & b_{r2} & ... & b_{r\ n-r} \end{pmatrix}.$$

Lineární kód C_H je tedy tvořen binárními slovy $\mathbf{a} = x_1x_2...x_n$, která splňují systém lineárních rovnic

$x_1 = b_{11}x_{r+1} + b_{12}x_{r+2} + ... + b_{1\ n-r}x_n$
$x_2 = b_{21}x_{r+1} + b_{22}x_{r+2} + ... + b_{2\ n-r}x_n$
.
.
.
$x_r = b_{r1}x_{r+1} + b_{r2}x_{r+2} + ... + b_{r\ n-r}x_n.$

Při hledání řešení tohoto systému volíme hodnoty bitů $x_{r+1}, x_{r+2}, ..., x_n$, které pak určují hodnoty bitů $x_1, x_2, ..., x_r$. Protože existuje 2^{n-r} možností volby bitů x_{r+1}, $x_{r+2}, ..., x_n$, je dimenze lineárního kódu C_H rovna číslu $n-r$, tedy je skutečně rovna předepsané hodnotě k, neboť $n - r = n - (n - k) = k$.

13.18. Cvičení. (1) Napište všechna kódová slova lineárního kódu C_H daného maticí

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Určete délku, dimenzi a minimální vzdálenost tohoto kódu.

(2) Zkonstruujte lineární kód pro zaslání 128 různých zpráv, má-li být každá zpráva reprezentována binárním slovem délky 11.

Následující věta udává jednoduchou dostatečnou podmínku k tomu, aby lineární kód daný kontrolní maticí opravoval alespoň jednu chybu.

13.19. Věta. Buď H kontrolní matice a nechť žádný sloupec matice H neobsahuje samé nuly a žádné dva sloupce nejsou stejné. Pak kód C_H opravuje jednu chybu.

Důkaz. V důsledku věty 13.7 stačí dokázat nerovnost $\delta \geq 3$, která je podle věty 13.14 ekvivalentní nerovnosti $w_{min} \geq 3$. Předpokládejme, že C_H obsahuje kódové slovo $\mathbf{a}$ takové, že $w(\mathbf{a}) = 1$. Pak v $\mathbf{a}$ je právě jeden bit roven 1. Předpokládejme, že se jedná o i-tý bit. Protože všechny ostatní bity se rovnají 0, $H\mathbf{a}'$ se rovná i-tému sloupci $\mathbf{h}^{(i)}$ matice H. Tedy podmínka $H\mathbf{a}' = \mathbf{0}'$ znamená, že sloupec $\mathbf{h}^{(i)}$ je tvořen samými nulami, což je spor s předpokladem věty. Takže C_H neobsahuje žádné slovo váhy 1.
Předpokládejme, že C_H obsahuje kódové slovo $\mathbf{b}$ takové, že $w(\mathbf{b}) = 2$. Pak v $\mathbf{b}$ jsou právě dva bity rovny 1. Předpokládejme, že se jedná o i-tý a j-tý bit. Pak

$$H\mathbf{b}' = \mathbf{h}^{(i)} + \mathbf{h}^{(j)},$$

takže podmínka $H\mathbf{b}' = \mathbf{0}'$ implikuje $\mathbf{h}^{(i)} = \mathbf{h}^{(j)}$. To je ovšem také spor s předpokladem věty. Proto C_H neobsahuje žádné slovo délky 2. Dostáváme tedy $w_{min} \geq 3$, čímž je věta dokázána.

13.20. Definice. Buď H kontrolní matice splňující podmínky věty 13.19 a mající maximální možný počet sloupců. Pak lineární kód C_H se nazývá *Hammingův*.

13.21. Věta. Buď C_H Hammingův kód, kde kontrolní matice H má r řádků. Pak pro jeho délku n, dimenzi k a minimální vzdálenost δ platí $n = 2^r - 1$, $k = 2^r - 1 - r$ a pokud $r > 2$, pak $\delta = 3$.

Důkaz. Kontrolní matice H s r řádky může mít nejvýše 2^r různých sloupců a vyloučíme-li sloupec tvořený samými nulami, pak těchto sloupců je nejvýše $2^r - 1$. Tedy maximální možný počet sloupců v kontrolní matici H s r řádky, která splňuje podmínky věty 13.19, je $2^r - 1$. Hammingův kód C_H má tedy délku $n = 2^r - 1$.
Protože matice H obsahuje všech r sloupců obsahujících právě jednu jedničku, je její hodnost rovna r. Kódová slova $\mathbf{a} = x_1x_2...x_{2^r-1}$ kódu C_H tedy dostaneme tak, že volíme $2^r - 1 - r$ bitů $x_{r+1},\ x_{r+2}, ..., x_{2^r-1}$ a zbývajících r bitů $x_1,\ x_2, ..., x_r$ dopočítáváme. Takže C_H má celkem 2^{2^r-1-r} kódových slov, tj. má dimenzi $2^r - 1 - r$. Podle věty 13.14 platí $\delta = w_{min}$ a z důkazu věty 13.19 víme, že $w_{mim} \geq 3$. Volíme-li v kódovém slovu $\mathbf{a} = x_1x_2...x_{2^r-1}$ všechny bity $x_{r+1},\ x_{r+2}, ..., x_{2^r-1}$ nulové, pak i bity $x_1,\ x_2, ..., x_r$ jsou nulové. Zvolme bity $x_{r+1},\ x_{r+2}, ..., x_{2^r-1}$ tak, že jsou všechny nulové s výjimkou jednoho z nich, řekněmě x_l, který má hodnotu 1. Z podmínky $H\mathbf{x}' = \mathbf{0}'$ zřejmě vyplývá, že dopočítané bity $x_1,\ x_2, ..., x_r$ musí být všechny nulové s výjimkou dvou z nich, které musí mít hodnotu jedna. Těmito dvěma bity jsou x_i a x_j, kde i a j jsou řádky, v nichž má matice v l-tém sloupci jedničky. Tedy výsledné slovo má váhu 3. Proto $w_{min} = 3$.

Hammingovy kódy jsou nejefektivnějšími lineárními kódy opravujícími jednu chybu, neboť mají největší možný počet kódových slov.

13.22. Příklad. Buď H kontrolní matice se třemi řádky taková, že C_H je Hammingův kód. Pak zřejmě platí

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Určeme všechna kódová slova Hammingova kódu C_H. Systém lineárních rovnic $H\mathbf{x}' = \mathbf{0}'$ má tvar

$x_1 = x_5 + x_6 + x_7$
$x_2 = x_4 + x_6 + x_7$
$x_3 = x_4 + x_5 + x_7$.

Tedy volbou $x_4,\ x_5,\ x_6,\ x_7$ určujeme $x_1,\ x_2,\ x_3$. Hammingův kód C_H má proto právě následujících $2^4 = 16$ kódových slov $x_1x_2x_3x_4x_5x_6x_7$:

0000000
1110001
1100010
0010011
1010100
0100101
0110110
1000111
0111000
1001001
1011010
0101011
1101100
0011101
0001110
1111111.

Věta 13.19 udává dostatečnou podmínku k tomu, aby lineární kód C_H daný kontrolní maticí H opravoval jednu chybu podle principu nejbližšího souseda. V praxi ovšem potřebujeme tuto chybu opravit, aniž bychom museli srovnávat došlé slovo se všemi kódovými slovy. To lze provést snadno následujícím způsobem.

Předpokládejme, že bylo vysláno kódové slovo $\mathbf{c}$ lineárního C_H daného maticí H a že došlo k chybě v jeho i-tém bitu. Pro přijaté slovo $\mathbf{z}$ platí

$$\mathbf{z} = \mathbf{c} + \mathbf{e},$$

kde $\mathbf{e}$ je slovo, jehož všechny bity mají hodnotu 0 s výjimkou i-tého bitu, který má hodnotu 1. Tedy máme

$$H\mathbf{z}' = H(\mathbf{c}+\mathbf{e})' = H\mathbf{c}' + H\mathbf{e}'.$$

Ovšem $H\mathbf{c}' = \mathbf{0}'$ a $H\mathbf{e}' = \mathbf{h}^{(i)}$, kde $\mathbf{h}^{(i)}$ je i-tý sloupec matice H. Tedy následujícím postupem nalezneme a opravíme jednu chybu kódu C_H:

1. Vypočteme $H\mathbf{z}'$, kde $\mathbf{z}$ je přijaté slovo.
2. Je-li $H\mathbf{z}' = \mathbf{0}'$, je $\mathbf{z}$ kódové slovo.
3. Je-li $H\mathbf{z}' \neq \mathbf{0}'$, nalezname sloupec $\mathbf{h}^{(i)}$ matice H, pro nějž platí $H\mathbf{z}' = \mathbf{h}^{(i)}$, a změníme i-tý bit slova $\mathbf{z}$.

13.23. Příklad. Uvažujme kód ze cvičení 13.18(1) a předpokládejme, že bylo přijato slovo $\mathbf{z} = 1111101$. Snadno vypočteme, že

$$H\mathbf{z}' = [1010]'.$$

Protože $[1010]'$ je pátý sloupec matice H, došlo k chybě v pátém bitu. Opravené kódové slovo tedy je 1111001.

13.24. Cvičení. (1) Uvažujme lineární kód C_H, kde

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Jestliže bylo přijato kódové slovo 010111 a víme, že nedošlo k více než jedné chybě, jaké slovo bylo vysláno?

(2) Napište kontrolní matici Hammingova kódu délky 15. Kolik má tento kód kódových slov? Určete, která z následujících binárních slov jsou jeho kódovými slovy:

011010110111000

100000000000011

110110110111111.

Ta slova, která nejsou kódová, opravte za předpoklau, že došlo k právě jedné chybě.

(3) Předpokládejme, že chceme poslat 256 zpráv lineárním kódem, který opravuje jednu chybu.

a) Jaká je nejmenší možná délka takového kódu?

b) Napište vhodnou kontrolní matici.

c) Určete nejmenší možnou délku příslušného lineárního kódu, jestliže má opravovat dvě chyby.

14. Literatura.

[1] N.L. Biggs, *Discrete Mathematics*, Oxford University Press, Oxford, 1999.

[2] M. Demlová a V. Koubek, *Algebraická teorie automatů*, SNTL, Praha, 1990.

[3] R. Faure a E. Heurgonová, Uspořádání a Booleovy algebry, Academia, Praha, 1984.

[4] D.R. Hankerson, D.G. Hoffman, D.A. Leonard, C.C. Linder, K.T. Phelps, C.A. Rodger and J.R. Wall, *Coding Theory and Cryptography*, Marcel Dekker, Inc., New York - Basel, 2000.

[5] J. Holenda a Z. Ryjáček, Lineární algebra II - Úvod do diskrétní matematiky, Západočeská Univerzita, fakulta aplikovaných věd, 2000.

[6] M. Chytil, *Automaty a gramatiky*, SNTL, Praha, 1984.

[7] S.V. Jablonskij, *Úvod do diskrétnej matematiky*, Alfa, Bratislava, 1984.

[8] J. Karásek a L. Skula, *Algebra a geometrie*, skripta FSI VUT v Brně, 2001.

[9] J. Kopka, *Svazy a Booleovy algebry*, Univerzita J.E. Purkyně v Ústí nad Labem, 1991.

[10] M. Novotný, *S algebrou od jazyka ke gramatice a zpět*, Academia, Praha, 1988.

[11] M. Piff, *Discrete Mathematics*, Cambridge University Press, Cambridge, 1991.

[12] A.D. Polimeni and H.J. Straight, *Foundations of Discrete Mathematics*, Brooks/Cole Publ. Comp., Pacific Grove, California, 1990.

[13] F.P. Preparata a R.T. Yeh, *Úvod do teórie diskrétnych matematických štruktúr*, Alfa, Bratislava, 1982

[14] L. Procházka a kol., *Algebra*, Academia, Praha, 1990.

[15] J.A. Šrejder, *Binární relace*, SNTL, Praha, 1978.