

Státnice – Obecná Informatika

(Snad budoucí Bc.) Karel Velička

25. září 2025

Tento dokument obsahuje teoretickou přípravu na státnice z Obecné informatiky na MFF UK pro rok 2024/25. Snažil jsem se, jak jen to šlo, dodržet strukturu zveřejněných požadavků. Valná většina uvedeného textu je z mých poznámek, které jsem si v průběhu studia udělal. Výjimkou jsou Logika, ADS a TeMno, kde jsem čerpal přímo z materiálů vyučujících (resp. spolužáka).

Další zdroje a veškeré, mnou sepsané poznámky za 3 roky Bc. studia, jsou k dispozici zde.

P.S. Za nic neručím. Určitě se tu nějaké chyby najdou.

Obsah

1 Společná matematika	4
1.1 Základy diferenciálního a integrálního počtu	4
1.1.1 Posloupnosti reálných čísel a jejich limity	4
1.1.2 Řady	5
1.1.3 Reálné funkce jedné reálné proměnné	5
1.1.4 Derivace a její aplikace	6
1.1.5 Integrály a jejich aplikace	8
1.2 Algebra a Lineární algebra	10
1.2.1 Algebraické struktury	10
1.2.2 Soustavy lineárních rovnic	10
1.2.3 Matice	11
1.2.4 Vektorové prostory	12
1.2.5 Lineární zobrazení	13
1.2.6 Skalární součin	14
1.2.7 Determinanty	15
1.2.8 Vlastní čísla a vlastní vektory	16
1.2.9 Positivně semidefinitní a pozitivně definitní matice	17
1.3 Diskrétní matematika	18
1.3.1 Relace	18
1.3.2 Ekvivalence a rozkladové třídy	18
1.3.3 Částečná uspořádání	18
1.3.4 Funkce	19
1.3.5 Permutace	19
1.3.6 Kombinační čísla	19
1.3.7 Princip inkluze a exkluze	20
1.3.8 Hallova věta	21
1.4 Teorie grafů	22
1.4.1 Základní pojmy teorie grafů	22
1.4.2 Základní příklady grafů	22
1.4.3 Souvislost grafů	23
1.4.4 Stromy	23
1.4.5 Rovinné grafy	24
1.4.6 Barevnost grafů	24
1.4.7 Hranová a vrcholová souvislost grafů	25
1.4.8 Orientované grafy	25
1.4.9 Toky v sítích	26
1.5 Pravděpodobnost a statistika	27
1.5.1 Pravděpodobnost, jevy, prostor	27
1.5.2 Náhodné veličiny a jejich rozdělení	27
1.5.3 Limitní věty	30
1.5.4 Bodové odhady	30
1.5.5 Intervalové odhady	31

1.5.6	Testování hypotéz	31
1.6	Logika	33
1.6.1	Syntaxe	33
1.6.2	Sémantika	35
1.6.3	Extenze teorií	36
1.6.4	Dokazatelnost	36
1.6.5	Věty o kompaktnosti a úplnosti výrokové a predikátové logiky	38
1.6.6	Rozhodnutelnost	38
2	Společná informatika	39
2.1	Automaty a jazyky	39
2.1.1	Regulární Jazyky	39
2.1.2	Bezkontextové jazyky a Zásobníkový automat	40
2.1.3	Rekurzivně spočetné jazyky	41
2.1.4	Chomského Hierarchie	42
2.2	Algoritmy a datové struktury	44
2.2.1	Časová složitost algoritmu.	44
2.2.2	Třídy složitosti	44
2.2.3	Metoda rozděl a panuj	45
2.2.4	Binární vyhledávací stromy	46
2.2.5	Třídění	46
2.2.6	Grafové algoritmy	47
2.3	Programovací jazyky (Java)	52
2.3.1	Základy	52
2.3.2	Vlákna	53
2.3.3	Lambda funkce	54
2.3.4	Race condition a Deadlock	54
2.4	Architektura počítačů a operačních systémů	55
3	Obecná informatika	57
3.1	Kombinatorika	57
3.1.1	Vytvořující funkce	57
3.1.2	Odhady faktoriálu a kombinačních čísel	58
3.1.3	Ramseyovy věty	60
3.1.4	Extremální kombinatorika	61
3.1.5	Samoopravné kódy	62
3.2	Diferenciální a integrální počet ve více rozměrech	63
3.2.1	Riemannův integrál	63
3.2.2	Funkce více proměnných	64
3.2.3	Metrický prostor	67
3.3	Pokročilé algoritmy a datové struktury	69
3.3.1	Dynamické programování	69
3.3.2	Grafové algoritmy	70
3.3.3	Algoritmy vyhledávání v textu	73
3.3.4	Algebraické algoritmy	75
3.3.5	RSA	77
3.3.6	Aproximační algoritmy	77
3.3.7	Paralelní třídění komparátorových sítí	79
3.4	Pokročilá diskretní matematika	80
3.4.1	Barvení grafů	80
3.4.2	Párování v grafech	81
3.4.3	Kreslení grafů na plochách.	82

3.4.4	Grafové minory	83
3.4.5	Množiny a zobrazení	83
3.4.6	Subvalence a ekvivalence množin	85
3.4.7	Dobré uspořádání	86

Společná matematika

Z následujících 6 témat budou vybrána 2.

1.1 Základy diferenciálního a integrálního počtu

1.1.1 Posloupnosti reálných čísel a jejich limity

Definice (Reálná posloupnost). Reálná posloupnost $(a_n)_{n=1}^{\infty} = (a_1, a_2, \dots) \in \mathbb{R}$ je funkce $a : \mathbb{N} \rightarrow \mathbb{R}$.

Definice (Limita posloupnosti). Nechť (a_n) je reálná posloupnost a $L \in \mathbb{R}^*$, kde $\mathbb{R}^*$ je $\mathbb{R}$ spolu s $\pm\infty$. Potom L je limita posloupnosti (a_n) , pokud:

$$\forall \varepsilon, \exists n_0 : n \geq n_0 \implies a_n \in U(L, \varepsilon), \quad U(b, \varepsilon) = (b - \varepsilon, b + \varepsilon).$$

Píšeme $\lim_{n \rightarrow \infty} a_n = L$.

Definice (Nevlastní limita posloupnosti). Pokud $L \in \mathbb{R}$, pak konverguje a mluvíme o limitě vlastní, pokud $L = \pm\infty$, pak diverguje a mluvíme o limitě nevlastní.

Definice (Podposloupnost). (b_n) je podposloupností posloupnosti (a_n) , pokud existuje taková posloupnost

$$\forall m \in \mathbb{N} : m_1 < m_2 < \dots \in \mathbb{N},$$

kde $\forall n : b_n = a_{m_n}$. Značíme jako $(b_n) \prec (a_n)$.

Definice (Hromadný bod). Hromadný bod A posloupnosti (a_n) , pokud je limitou nějaké podposloupnosti posloupnosti (a_n) .

Věta (Aritmetika limit). Nechť $(a_n), (b_n)$ jsou posloupnosti reálných čísel s $\lim_{n \rightarrow \infty} (a_n) = K \in \mathbb{R}^*$, $\lim_{n \rightarrow \infty} (b_n) = L \in \mathbb{R}^*$. Potom, pokud jsou výrazy na pravých stranách definovány, platí

$$(i) \quad \lim_{n \rightarrow \infty} (a_n + b_n) = K + L,$$

$$(ii) \quad \lim_{n \rightarrow \infty} (a_n \cdot b_n) = K \cdot L,$$

$$(iii) \quad \lim_{n \rightarrow \infty} \left(\frac{a_n}{b_n} \right) = \frac{K}{L}, \text{ pokud } b_n \neq 0 \text{ pro každé } n > n_0.$$

Věta (O dvou policajtech). Nechť posloupnosti $(a_n), (b_n), (c_n) \in \mathbb{R}$ splňují, že $\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = a \in \mathbb{R}$ a $\forall n > n_0 : a_n < c_n < b_n$. Pak (c_n) konverguje a $\lim_{n \rightarrow \infty} c_n = a$.

Věta (O limitě a uspořádání). Nechť posloupnosti $(a_n), (b_n) \in \mathbb{R}$ mají limity $\lim a_n = A \in \mathbb{R}^*$ a $\lim b_n = B \in \mathbb{R}^*$. Potom:

$$(i) \quad A < B \implies \exists n_0 : \forall n \geq n_0 \text{ platí } a_n < b_n,$$

$$(ii) \quad \forall n : a_n \leq b_n \implies A \leq B.$$

1.1.2 Řady

Definice (Řada). Řada je posloupnost $(a_n) \subseteq \mathbb{R}$.

Definice (Částečný součet řady). Částečný součet řady (a_n) je $(s_n) := (a_1 + a_2 + \dots + a_n)$

Definice (Součet řady). Součet řady je limita $\sum a_n = \sum_{n=1}^{\infty} a_n = a_1 + a_2 + \dots := \lim_{n \rightarrow \infty} (a_1 + a_2 + \dots + a_n) = \lim s_n \in \mathbb{R}^*$.

Definice (Geometrická řada). Geometrická řada je řada $\sum_{n=0}^{\infty} q^n = 1 + q + q^2 + \dots + q^n + \dots$, kde $q \in \mathbb{R}$ je kvocient.

Definice (Součet geometrické řady). $\sum_{n=0}^{\infty} q^n = \begin{cases} \frac{1}{1-q} & \text{pro } |q| < 1 \\ +\infty & \text{pro } q \geq 1 \\ \text{neexistuje} & \text{pro } q \leq -1 \end{cases}$

Definice (Absolutně konvergentní řada). Řada $\sum a_n$ je absolutně konvergentní, pokud konverguje řada $\sum |a_n|$.

Definice (Harmonická řada). Harmonickou řadu definujeme jako $h_n = \sum_{n=1}^{\infty} \frac{1}{n^s}$, kde $s \in \mathbb{R}$. Platí, že h_n konverguje, pokud $s > 1$, jinak diverguje.

1.1.3 Reálné funkce jedné reálné proměnné

Definice (Okolí bodu). ε -okolí bodu $b \in \mathbb{R}$, kde $\varepsilon \in \mathbb{R}^+$ je interval

$$U(b, \varepsilon) := (b - \varepsilon, b + \varepsilon) = \{x \in \mathbb{R} : d(x, b) < \varepsilon\}, \quad \text{pro metriku } d.$$

Definice (Prstencové okolí bodu). Prstencové okolí bodu $b \equiv P(b, \varepsilon) := (b - \varepsilon, b) \cup (b, b + \varepsilon)$.

Definice (Limita funkce v bodě). Funkce f má v bodě $a \in \mathbb{R}^*$ limitu $A \in \mathbb{R}^*$, když

$$\forall \varepsilon > 0, \exists \delta > 0, \forall x \in P(a, \delta) \implies f(x) \in U(A, \varepsilon), \quad \text{tedy} \quad \lim_{x \rightarrow a} f(x) = A.$$

Definice (Jednostranná limita funkce). Podobně, jen $\forall x \in P^{\pm}(a, \delta) \dots$

Věta (Vztah limit funkce s uspořádáním). Nechť $c \in \mathbb{R}^*$ a funkce f, g, h jsou definované na nějakém prstencovém okolí bodu c .

- (1) Mají-li funkce f, g v bodě c limitu a $\lim_{x \rightarrow c} f(x) > \lim_{x \rightarrow c} g(x)$, pak $\exists \delta > 0$ takové, že $f(x) > g(x)$ pro každé $x \in P(c, \delta)$. (Limita zachovává ostré nerovnosti.)
- (2) Existuje-li $\delta > 0 : f(x) \geq g(x)$ pro každé $x \in P(c, \delta)$, a mají-li funkce f, g limitu v bodě c , potom $\lim_{x \rightarrow c} f(x) \geq \lim_{x \rightarrow c} g(x)$. (Limita zachovává neostré nerovnosti.)
- (3) Existuje-li $\delta > 0 : f(x) \leq h(x) \leq g(x)$ pro každé $x \in P(c, \delta)$ a $\lim_{x \rightarrow c} f(x) = \lim_{x \rightarrow c} g(x) = A \in \mathbb{R}^*$, potom i $\lim_{x \rightarrow c} h(x) = A$. (O dvou policajtech.)

Definice (Spojitost funkce v bodě). Funkce f je spojitá v bodě $a \in \mathbb{R}$, pokud

$$\forall \varepsilon > 0, \exists \delta > 0, \forall x \in P(a, \delta) \implies f(x) \in U(f(a), \varepsilon).$$

Neboli funkce f je v bodě a spojitá, pokud $\lim_{x \rightarrow a} f(x) = f(a)$.

Definice (Limita složené funkce). Necht' $a, b, c \in \mathbb{R}^*$ a necht' funkce mají limity $\lim_{x \rightarrow a} g(x) = b$ a $\lim_{x \rightarrow b} f(x) = c$. Složená funkce má potom limitu $\lim_{x \rightarrow a} f(g(x)) = c \iff$ platí jedna z podmínek:

- (i) $f(x)$ je spojitá v c ,
- (ii) $\exists \delta, \forall x \in P(a, \delta) : b \notin g(x)$, tedy na nějakém prstencovém okolí funkce nenabývá hodnotu b .

Definice (Spojitost na intervalu). Funkce je spojitá na intervalu, je-li spojitá v každém vnitřním bodu a jednostraně spojitá v mezích.

Věta (Nabývání mezihodnot). Funkce spojitá na intervalu nabývá všech hodnot mezi mezemi intervalu. Tedy pro $a, b, c \in \mathbb{R}; a < b; f : [a, b] \rightarrow \mathbb{R}$ je spojitá a $f(a) \geq c \geq f(b)$, pak $\exists d \in (a, b) : f(d) = c$.

Definice (Maximum). Necht' $M \subseteq \mathbb{R}$ a $f : M \rightarrow \mathbb{R}$. Řekneme, že funkce f v bodě $a \in M$ nabývá svého maxima, když $\forall x \in M : f(x) \leq f(a)$. (minimum analogicky, jen opačná nerovnost).

Věta (Princip minima a maxima). Necht' $a, b \in \mathbb{R}, a \leq b$ a $f : [a, b] \rightarrow \mathbb{R}$ je spojitá funkce. Potom f nabývá na intervalu $[a, b]$ svého maxima i minima.

1.1.4 Derivace a její aplikace

Definice (Derivace funkce). Necht' bod $a \in M$ je limitní bod množiny $M \subseteq \mathbb{R}$ a $f = f(x) : M \rightarrow \mathbb{R}$ je funkce. Potom derivace f v bodě a je limita

$$f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} = \lim_{h \rightarrow 0} \frac{f(a + h) - f(a)}{h}.$$

Definice (Diferencovatelnost). Jestliže má f v bodě b vlastní derivaci, říkáme, že f je v b diferencovatelná. (Diferencovatelnost $\implies$ spojitost).

Věta (Derivace složené funkce (řetízkové pravidlo)). Necht' f, g funkce a g má spojitou derivaci v x :

$$(f(g(x)))' = f'(g(x)) \cdot g'(x).$$

Pravidlo	Vzorec
Derivace součtu	$(f + g)' = f' + g'$
Derivace násobku	$(\alpha f)' = \alpha f' \quad \forall \alpha \in \mathbb{R}$
Leibnizovo pravidlo (součin)	$(f \cdot g)' = f' \cdot g + f \cdot g'$
Derivace podílu	$\left(\frac{f}{g}\right)' = \frac{f' \cdot g - f \cdot g'}{g^2}, \quad g \neq 0$
Elementární funkce	Derivace
Mocninná funkce	$(x^n)' = nx^{n-1}, \quad n \in \mathbb{R}$
Exponenciální funkce	$(e^x)' = e^x, \quad (a^x)' = a^x \ln a$
Logaritmická funkce	$(\ln x)' = \frac{1}{x}, \quad (\log_a x)' = \frac{1}{x \ln a}$
Goniometrické funkce	$(\sin x)' = \cos x$ $(\cos x)' = -\sin x$ $(\tan x)' = \frac{1}{\cos^2 x} = 1 + \tan^2 x$ $(\cot x)' = -\frac{1}{\sin^2 x} = -(1 + \cot^2 x)$

Věta (l'Hospitalovo pravidlo). *Nechť $a \in \mathbb{R}$; $f, g : P(a, \delta) \rightarrow \mathbb{R}$ mají vlastní derivace, $g' \neq 0$ a $\lim_{x \rightarrow a} f(x) = \lim_{x \rightarrow a} g(x) = 0$ nebo $\lim_{x \rightarrow a} g(x) = \pm\infty$, potom:*

$$\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \lim_{x \rightarrow a} \frac{f'(x)}{g'(x)}, \text{ pokud poslední limita existuje.} \quad \boxed{\frac{0}{0}, \frac{\pm\infty}{\pm\infty}}$$

Věta platí i pro $P^\pm(a, \delta)$ a pro $a = \pm\infty$.

Definice (Taylorův polynom funkce). *Nechť $\forall n \in \mathbb{N} : f, f', \dots, f^{(n-1)} : U(b, \delta) \rightarrow \mathbb{R}$ a $\exists f^{(n)}(b) \in \mathbb{R}$. Potom Taylorův polynom funkce f řádu n se středem v bodě b je*

$$T_n^{f,b}(x) := \sum_{k=0}^n \frac{f^{(k)}(b)}{k!} (x-b)^k.$$

Věta (Taylorova řada funkce). *Má-li funkce f v bodě $a \in \mathbb{R}$ derivace všech řádů, rozumíme pro $x \in \mathbb{R}$ její Taylorovou řadou se středem v b řadu*

$$T^{f,b}(x) := \sum_{n=0}^{\infty} \frac{f^{(n)}(b)}{n!} (x-b)^n.$$

Funkce	Taylorův polynom	Konvergence
Exponenciální	$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$	$\forall x \in \mathbb{R}$
Sinus	$\sin x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$	$\forall x \in \mathbb{R}$
Kosinus	$\cos x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!}$	$\forall x \in \mathbb{R}$
Přirozený logaritmus	$\ln(1+x) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^k}{k}$	$x \in (-1, 1]$

Vyšetření průběhu funkce

Určíme definiční obor, speciální tvar (sudost/lichost/periodicita), limity krajních bodů, 1. derivace, extrémy a monotonie, 2. derivace, inflexní body a konvexita/konkavita, asymptoty, náčrt.

Věta (Lagrangeova). *Pokud f je spojitá funkce, pak $\exists c \in (a, b) : f'(c) = \frac{f(b) - f(a)}{b - a} =: z$.*

Definice (Extrémy a monotonie). *Nechť f má první derivaci, potom pokud*

- $f'(x) = 0$, pak má extrém,*
- $f'(x) > 0$, pak roste (resp. $f'(x) < 0$ klesá).*

Definice (Konvexita/ Konkavita). *Nechť f má druhou derivaci, potom pokud $f''(x) > 0$, pak je konvexní a pokud $f''(x) < 0$, pak je konkávní.*

Definice (Inflexní bod). *Inflexní bod je bod, ve kterém $f'' = 0$ a $f' = 0$ nebo f' neexistuje. (Dochází ke změně směru funkce).*

1.1.5 Integrály a jejich aplikace

Definice (Primitivní funkce). *Nechť $I \subseteq \mathbb{R}$ je netriviální interval a $F, f : I \rightarrow \mathbb{R}$. Potom F je primitivní funkce k f , neboli $F = \int f$, pokud $F' = f$ na celém I .*

Definice (Metody výpočtu: Substitute).

$$\int f(g(x)) g'(x) dx = \left| \begin{array}{l} y = g(x) \\ dy = g'(x) dx \end{array} \right| = \int f(y) dy = F(y) + C \stackrel{y=g(x)}{=} F(g(x)) + C$$

$$\int_a^b f(g(x)) g'(x) dx = \left| \begin{array}{l} y = g(x) \\ dy = g'(x) dx \\ x = a \rightsquigarrow y = g(a) \\ x = b \rightsquigarrow y = g(b) \end{array} \right| = \int_{g(a)}^{g(b)} f(y) dy$$

Definice (Metody výpočtu: Per partes). $\int f'g = fg - \int fg'$

Elementární funkce	Neurčitý integrál $\int f(x) dx$ (bez konst. C)
Mocninná funkce	$\int x^n = \frac{x^{n+1}}{n+1} \quad (n \neq -1)$
Exponenciální funkce	$\int e^x = e^x$, $\int a^x = \frac{a^x}{\ln a} \quad (a > 0, a \neq 1)$
Logaritmická funkce	$\int \frac{1}{x} = \ln x $
Goniometrické funkce	$\int \sin x = -\cos x$ $\int \cos x = \sin x$ $\int \tan x = -\ln \cos x $ $\int \cot x = \ln \sin x $
Goniometrické funkce složitější	$\int \frac{1}{\cos^2 x} = \tan x$ $\int \frac{1}{\sin^2 x} = -\cot x$ $\int \frac{1}{\sqrt{1-x^2}} = \arcsin x$ $\int \frac{1}{1+x^2} = \arctan x$

Newtonův a Riemannův integrál: definice a souvislost.

Definice (Newtonův integrál funkce). Newtonův integrál funkce f na intervalu (a, b) , $a < b$:

$$(N) \int_a^b f(x) dx := [F]_a^b = F(b^-) - F(a^+) = \lim_{x \rightarrow b^-} F(x) - \lim_{x \rightarrow a^+} F(x)$$

Definice (Dělení). Rozdělení intervalu $[a, b]$ je posloupnost $P = (t_0, \dots, t_n)$, kde:

$$a = t_0 < t_1 < \dots < t_{n-1} < t_n = b.$$

Definice (Horní/dolní Riemannův součet). Pro omezenou funkci $f : J = [a, b] \rightarrow \mathbb{R}$ a rozklad P definujeme dolní a horní součty:

$$s(f, P) = \sum_{j=1}^n m_j(t_j - t_{j-1}), \quad \text{resp.} \quad S(f, P) = \sum_{j=1}^n M_j(t_j - t_{j-1}),$$

kde

$$m_j = \inf\{f(x) : t_{j-1} \leq x \leq t_j\}, \quad a \quad M_j = \sup\{f(x) : t_{j-1} \leq x \leq t_j\}.$$

Definice (Horní/dolní Riemannův integrál). Horní a dolní Riemannův integrál f přes $[a, b]$ je:

$$\int_a^b f(x) dx = \sup\{s(f, P) : P \text{ dělení}\} \quad a \quad \overline{\int}_a^b f(x) dx = \inf\{S(f, P) : P \text{ dělení}\}$$

Definice (Riemannův integrál funkce). Riemannův integrál funkce f přes $[a, b]$ je

$$(R) \int_a^b f(x) dx, \text{ pokud } \int_a^b f(x) dx = \overline{\int}_a^b f(x) dx.$$

Věta (Základní věta analýzy 1).¹ Pro Riemannovsky integrovatelnou f a primitivní F je

$$F(x) = \int_a^x f(t) dt$$

a platí, že

(1) F je spojitá na $[a, b]$ a

(2) v každém bodě spojitosti $x \in [a, b]$ funkce f existuje $F'(x) = f(x)$.

Věta (Základní věta analýzy 2). Pokud f je Newtonovsky i Riemannovsky integrovatelná, pak

$$(R) \int_a^b f = (N) \int_a^b f$$

Aplikace integrálů.

Věta (Odhady součtu řad). Pro neklesající (resp. nerostoucí) f na intervalu $[1, n]$ platí

$$\sum_{k=1}^{n-1} f(k) \leq \int_1^n f \leq \sum_{k=2}^n f(k)$$

Pro neklesající (resp. nerostoucí) funkci f na intervalu $[1, \infty)$ platí

$$\sum_{k=1}^{\infty} f(k) \text{ konverguje} \iff \int_1^{\infty} f(x) dx \text{ konverguje}.$$

Věta (Obsahy rovinných útvarů). Plocha mezi dvěma funkcemi $f(x)$ a $g(x)$:

$$S = \int_a^b |f(x) - g(x)| dx$$

Věta (Délka křivky). Pro graf funkce $y = f(x)$: $L = \int_a^b \sqrt{1 + [f'(x)]^2} dx$

Věta (Objem rotačního tělesa). Zvlášť pro objem a povrch rotačního tělesa:

• Objem vzniklý rotací kolem osy x : $V = \pi \int_a^b [f(x)]^2 dx$.

• Povrch rotační plochy: $S = 2\pi \int_a^b f(x) \sqrt{1 + [f'(x)]^2} dx$.

¹Primitivní funkce lze spočítat Riemannovským integrálem.

1.2 Algebra a Lineární algebra

1.2.1 Algebraické struktury

Definice (Grupa). Grupa je dvojice $\mathcal{G} = (G, \circ)$, kde G je nosná množina, na které je definována binární operace $\circ : G \times G \rightarrow G$, splňující:

- (i) $\forall a, b, c \in G : a \circ (b \circ c) = (a \circ b) \circ c$ (asociativita),
- (ii) $\exists e \in G \forall a \in G : a \circ e = e \circ a = a$ (existence neutrálního prvku),
- (iii) $\forall a \in G \exists b \in G : a \circ b = b \circ a = e$ (existence inverzního prvku).

Pokud platí navíc ještě komutativita, jedná se o Abelovskou grupu.

Definice (Podgrupa). Nechť $\mathcal{G} = (G, \circ)$ a $\mathcal{H} = (H, \tilde{\circ})$ jsou grupy, potom $\mathcal{H}$ je podgrupa grupy $\mathcal{G}$, značeno $\mathcal{H} \subseteq \mathcal{G}$, pokud: $\forall a, b \in H : a \tilde{\circ} b = a \circ b$ a pokud jsou zachovány inverzní a neutrální prvky.

Definice (Permutace). Permutace na množině $[n]$ je bijekce $p : [n] \rightarrow [n]$.

Definice (Inverze v permutaci). Pro inverze v permutaci platí $p(i) = j \iff p^{-1}(j) = i$.

Definice (Znaménko permutace). Znaménko permutace p je číslo $\text{sgn}(p) = (-1)^{\# \text{inverzí v } p}$.

Definice (Těleso). Nechť K je množina a $(\oplus, *)$ jsou binární operace na K . Trojici $\mathbb{K} = (K, \oplus, *)$ potom nazýváme tělesem, splňuje-li:

- (i) $(K, \oplus)$ tvoří Abelovskou grupu s neutrálním prvkem 0,
- (ii) $(K \setminus \{0\}, *)$ tvoří Abelovskou grupu s neutrálním prvkem 1,
- (iii) platí distributivita, tedy $(\forall a, b, c \in K) : a * (b \oplus c) = a * b \oplus a * c$.

Definice (Charakteristika tělesa). Pokud $\exists n \in \mathbb{N}$ t.ž. v tělese $\mathbb{K}$ platí $\underbrace{1 + 1 + \dots + 1}_{n\text{-krát}} = 0$, potom nejmenší takové n je $\text{char}(\mathbb{K})$ tělesa $\mathbb{K}$. Jinak má těleso charakteristiku 0.

Věta (Konečná tělesa). Existují konečná tělesa právě o velikostech p^n , kde p je prvočíslo a $n \geq 1$.

1.2.2 Soustavy lineárních rovnic

Definice (Maticový zápis). Pro soustavu $Ax = b$, kde $A \in \mathbb{R}^{m \times n}$ je matice soustavy, $x = (x_1, \dots, x_n)^T$ je vektor neznámých a b je vektor pravých stran, je rozšířená matice soustavy:

$$A^{m \times n} = \left(\begin{array}{ccc|c} a_{1,1} & \cdots & a_{1,n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & \cdots & a_{m,n} & b_m \end{array} \right)$$

Definice (Elementární řádkové úpravy). Elementární řádkovou úpravou vznikne z matice A matice A' ($A \sim A'$):

- (i) vynásobením i -tého řádku $t \in \mathbb{R} \setminus \{0\}$,
- (ii) přičtením j -tého řádku k i -tému, když $i \neq j$,
- (iii) přičtení t -násobku j -tého řádku k i -tému, když $j \neq i$,
- (iv) prohození dvou řádků.

Definice (Odstupňovaný tvar matice (REF)). Matice A je v REF, pokud **(a)** nenulové řádky jsou seřazeny podle počátečních nul a **(b)** nulové řádky jsou pod nenulovými.

Matice je v RREF, pokud je v REF, pivoty jsou jednotkové a nad i pod pivoty jsou pouze nuly.

Definice (Pivot). Pivot je první nenulový prvek $a_{i,j(i)}$ na i -tém řádku.

Algorithm 1: Gaussova eliminace – převod elementárními úpravami na REF

- 1 Seřaď řádky podle počtu počátečních nul.
 - 2 Pokud mají dva nenulové řádky stejný počet počátečních nul (i -tý a $i + 1$ -ní), tak od $i + 1$ -ního odečteme $\frac{a_{i+1,j(i)}}{a_{i,j(i)}}$ -násobek i -tého.
 - 3 Opakuj, dokud nemají každé dva nenulové řádky různé počty počátečních nul.
-

Algoritmus je konečný, protože po kroku 2. vždy vzroste celkový počet počátečních 0 alespoň o 1.

Algorithm 2: Gauss-Jordanova eliminace – převod na RREF

- 1 Seřaď řádky podle počtu počátečních nul.
 - 2 **forall** pivotní prvek $a_{i,j(i)}$ (první nenulový prvek v řádku i) **do**
 - 3 Normalizuj řádek i : vyděl řádek i hodnotou $a_{i,j(i)}$
 - 4 **forall** ostatní řádky $k \neq i$ **do**
 - 5 Odečti $a_{k,j(i)}$ -násobek řádku i od řádku k
 - 6 **end**
 - 7 **end**
 - 8 Opakuj, dokud není matice REF.
-

Algoritmus je konečný, protože v každé iteraci se buď zvýší počet počátečních nul, nebo se pivotní prvek přesune blíže k diagonále.

Věta (Frobeniova). Soustava $Ax = b$ má řešení $\iff \text{rank}(A)$ se rovná hodnotě rozšířené matice.

Počet řešení	Podmínka
Žádné	Existuje řádek $(0 \ \cdots \ 0 \mid b)$, $b \neq 0$
Právě jedno	$\# \text{ pivotů} = \# \text{ neznámých}$, žádné konflikty
Nekonečně mnoho	$\# \text{ pivotů} < \# \text{ neznámých}$, žádné konflikty

1.2.3 Matice

Definice (Jednotková matice). Pro $(\forall n \in \mathbb{N})$ je jednotková matice $I_n \in \mathbb{R}^{n \times n}$ definována vztahy:

$$(I_n)_{i,j} = \begin{cases} 1 & \text{pokud } i = j \\ 0 & \text{pokud } i \neq j \end{cases} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Definice (Transponovaná matice). Transponovaná matice k matici $A \in \mathbb{R}^{m \times n}$ je taková matice $A^T \in \mathbb{R}^{n \times m}$, pro kterou platí:

$$A_{i,j}^T = A_{j,i} \quad \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \rightsquigarrow \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

Definice (Symetrická matice). Symetrická matice je taková čtvercová matice $A \in \mathbb{R}^{n \times n}$, že:

$$A_{j,i} = A_{i,j}, \text{ neboli } A = A^T$$

Definice (Inverzní matice). Inverzní matice k čtvercové matici $A \in \mathbb{R}^{n \times n}$ je taková matice $A^{-1} \in \mathbb{R}^{n \times n}$, pro kterou platí:

$$A \cdot A^{-1} = I_n$$

Definice (Regulární matice). Regulární matice je čtvercová matice, ke které existuje inverzní matice. Ekvivalentně, pokud je $\text{rank}(A) = n$. Ekvivalentně, pokud $Ax = 0$ má pouze triviální řešení $x = 0$. Ekvivalentně, pokud $\det(A) \neq 0$. Ekvivalentně, pokud její sloupce nebo řádky jsou lineárně nezávislé. Ekvivalentně, pokud žádné vlastní číslo není nulové.

Definice (Singularní matice). Singularní matice je taková matice, která není regulární.

Definice (Rovnost matic). Dvě matice se rovnají, $A = B$, pokud mají stejné rozměry $m \times n$ a

$$A_{ij} = B_{ij} \quad , \text{ pro } i = 1, \dots, m, j = 1, \dots, n.$$

Definice (Součet Matic). Bud' $A, B \in \mathbb{R}^{m \times n}$. Pak $A + B$ je matice typu $m \times n$ s prvky

$$(A + B)_{ij} = A_{ij} + B_{ij} \quad , \text{ pro } i = 1, \dots, m, j = 1, \dots, n.$$

Definice (Maticový součin). Pro součin dvou matic $A \in \mathbb{R}^{m \times n}$ a $B \in \mathbb{R}^{n \times p}$ platí $(AB) \in \mathbb{R}^{m \times p}$:

$$(AB)_{i,j} = \sum_{k=1}^n a_{i,k} \cdot b_{k,j}$$

Definice (Hodnost matice). Hodnost matice A , značená jako $\text{rank}(A)$, je počet pivotů v libovolné matici A' v REF takové, že $A \sim A'$.

1.2.4 Vektorové prostory

Definice (Vektorový prostor). Vektorový prostor $(V, \oplus, *)$ nad tělesem $(\mathbb{K}, \oplus, *)$ je množina V spolu s binárními operacemi $\oplus$ na V a binární operací skalárního násobku $*$: $\mathbb{K} \times V \rightarrow V$, kde:

- (i) $(V, \oplus)$ tvoří Abelovskou grupu
- (ii) $(\forall v \in V) : 1 * v = v$, (kde 1 je neutrální prvek pro násobení v $\mathbb{K}$)
- (iii) $(\forall a, b \in \mathbb{K})(\forall v \in V) : (a * b) * v = a * (b * v)$ – asociativita
- (iv) $(\forall a, b \in \mathbb{K})(\forall v \in V) : (a \oplus b) * v = (a * v) \oplus (b * v)$ – distributivita
- (v) $(\forall a \in \mathbb{K})(\forall u, v \in V) : a * (u \oplus v) = (a * u) \oplus (a * v)$ – distributivita

Prvky $\mathbb{K}$ se nazývají skaláry a prvky V vektory.

Definice (Podprostor vektorového prostoru). Podprostor U vektorového prostoru V je neprázdna podmnožina $U \subseteq V$, splňující uzavřenost na operaci $\oplus, *$ a obsahující nulový vektor.

Definice (Lineární kombinace). Lineární kombinace vektorů $v_1, \dots, v_n \in V$ nad $\mathbb{K}$ je libovolný vektor $u = a_1 \cdot v_1 + \dots + a_n \cdot v_n = \sum_{i=1}^n a_i v_i$, kde $a_1, \dots, a_n \in \mathbb{K}$.

Definice (Lineární obal). Lineární obal $\text{span}(X)$ množiny $X \subseteq V$, kde V je vektorový prostor nad $\mathbb{K}$, je průnik všech podprostorů U z V obsahující X . Tedy množina všech jejích lineárních kombinací.

$$\text{span}(X) = \bigcap \{U : X \subseteq U, U \text{ je podprostor } V\} = \left\{ \sum_{i=1}^n a_i v_i \mid a_i \in \mathbb{K}, v_i \in X \right\}.$$

Definice (Lineární nezávislost). Množina vektorů X ve vektorovém prostoru V je lineárně nezávislá, pokud nelze nulový vektor získat netriviální lineární kombinací vektorů z X .

Tedy, vektory $v_1, \dots, v_n$ jsou LN $\iff \sum_{i=1}^n a_i v_i = 0$ má pouze triviální řešení $a_1 = \dots = a_n = 0$.

Definice (Báze). Báze vektorového prostoru V je lineárně nezávislá množina X , která generuje² V .

- (1) $\text{span}(X) = V$, každý vektor V je lineární kombinací vektorů báze X
- (2) X je lineárně nezávislá, proto je lin. kombinace unikátní pro každý vektor V .

Definice (Dimenze). Dimenze vektorového prostoru V , $\dim(V)$, je mohutnost jeho (konečné) báze.

Definice (Jádro). Jádro matice $A \in \mathbb{K}^{m \times n}$ je podprostor $\mathbb{K}^n$ tvořen řešeními soustavy $Ax = 0$.

$$\ker(A) = \{(x \in \mathbb{K}^n) : Ax = 0\}$$

Platí vztah: $\dim(\ker(A)) + \text{rank}(A) = n$.

Definice (Řádkový prostor). Řádkový prostor matice je prostor generovaný jejími řádky. Pro matici $A \in \mathbb{K}^{m \times n}$:

$$\mathcal{R}(A) = \mathcal{S}(A^T) = \sum_{j=1}^m x_j A_{j,*}$$

$$\mathcal{R}(A) = \{(v \in \mathbb{K}^n) : v = A^T y, y \in \mathbb{K}^m\}, \text{ všechny lineární kombinace řádků}$$

Řádkový prostor je kolmý na jádro, tedy $\mathcal{R}(A)^\perp = \ker(A)$.

Definice (Sloupcový prostor). Sloupcový prostor matice je prostor generovaný jejími sloupci. Pro matici $A \in \mathbb{K}^{m \times n}$:

$$\mathcal{S}(A) = \mathfrak{L}\{A_{*,1}, \dots, A_{*,n}\} = \sum_{j=1}^n x_j A_{*,j}$$

$$\mathcal{S}(A) = \{(u \in \mathbb{K}^m) : u = Ax, x \in \mathbb{K}^n\}, \text{ všechny lineární kombinace sloupců}$$

Platí vztah: $\dim(\mathcal{R}(A)) = \dim(\mathcal{S}(A)) = \text{rank}(A)$.

Definice (Vektor souřadnic). Nechť $B = (v_1, \dots, v_n)$ je konečná uspořádaná báze vektorového prostoru V nad tělesem $\mathbb{K}$. Vektor souřadnic $u \in V$ vzhledem k bázi B je $[u]_B = (a_1, \dots, a_n)^T \in \mathbb{K}^n$, kde $u = \sum_{i=1}^n a_i v_i$. Neboli, obsahuje v B koeficienty LK bázeckých vektorů B , která tvoří u .

Věta (Steinitzova o výměně). Nechť X je konečná LN množina vektorového prostoru V nad $\mathbb{K}$ a Y je systém generátorů V . Potom platí $|X| \leq |Y|$ a existuje Z , taková že:

$$\text{span}(Z) = V, \quad X \subseteq Z, \quad |Z| = |Y|, \quad Z \setminus X \subseteq Y.$$

1.2.5 Lineární zobrazení

Definice (Lineární zobrazení). Nechť U a V jsou vektorové prostory nad stejným tělesem $\mathbb{K}$. Potom zobrazení $f : U \rightarrow V$ se nazývá lineární zobrazení, pokud splňuje:

- (1) $(\forall u, v \in U) : f(u + v) = f(u) + f(v)$
- (2) $(\forall u \in U), (\forall a \in \mathbb{K}) : f(a \cdot u) = a \cdot f(u)$

Maticová reprezentace LZ pro matici A a vektor u je $f(u) = Au$.

²Pro “ X generuje V ” píšeme $V = \text{span}(X)$. Prvky X jsou generátory V .

Definice (Matice lineárního zobrazení). *Nechť V a W jsou vektorové prostory nad stejným tělesem $\mathbb{K}$ s bázemi $X = (v_1, \dots, v_n)$, $Y = (w_1, \dots, w_m)$. Matice lineárního zobrazení $f : V \rightarrow W$ vzhledem k bázím X a Y je $[f]_{X,Y} \in \mathbb{K}^{m \times n}$, jejíž sloupce jsou vektory souřadnic obrazů vektorů báze X vzhledem k bázi Y . Neboli,*

$$[f]_{X,Y} = \left(\begin{array}{c|ccc|c} & & & & \\ \hline & [f(v_1)]_Y & \dots & [f(v_n)]_Y & \\ \hline \end{array} \right) \cdot \rightsquigarrow [f(u)]_Y = [f]_{X,Y} [u]_X$$

Definice (Jádro lineárního zobrazení). Jádro lineárního zobrazení $f : U \rightarrow V$ je

$$\ker(f) = \{w \in U : f(w) = 0\}.$$

Definice (Izomorfismus prostorů). *Bijektivní lineární zobrazení $f : V \rightarrow W$, nazýváme izomorfismem prostorů V a W . Ekvivalentně, pokud je $[f]_{X,Y}$ regulární, kde X, Y jsou báze V, W .*

1.2.6 Skalární součin

Definice (Skalární součin). Skalární součin na vektorovém prostoru V nad $\mathbb{C}$ je zobrazení, které přiřadí každé dvojici vektorů $u, v \in V$ skalár $\langle u | v \rangle \in \mathbb{C}$ tak, že jsou splněny následující axiomy:

- $\forall u \in V : \langle u | u \rangle \geq 0$, rovnost pokud $u = 0$,
- $\forall u, v \in V : \langle v | u \rangle = \overline{\langle u | v \rangle}$, (komplexně sdružené)
- $\forall u, v, w \in V : \langle u + v | w \rangle = \langle u | w \rangle + \langle v | w \rangle$,
- $\forall u, v \in V, \forall \alpha \in \mathbb{C} : \langle \alpha u | v \rangle = \alpha \langle u | v \rangle$.

Definice (Norma indukovaná skalárním součinem). *Nechť V je prostor se skalárním součinem nad $\mathbb{C}$ nebo $\mathbb{R}$, pak norma odvozená ze skalárního součinu je zobrazení $V \rightarrow \mathbb{R}_0^+$ přiřazující vektoru u jeho normu $\|u\| = \sqrt{\langle u | u \rangle}$.*

Věta (Pythagorova). *Pokud V je VP nad $\mathbb{C}$ a $x, y \in V$ jsou kolmé, tak $\|x + y\|^2 = \|x\|^2 + \|y\|^2$.*

Věta (Cauchy-Schwartzova nerovnost). *Pro skalární součin libovolných dvou vektorů u a v ve vektorovém prostoru nad $\mathbb{C}$ platí:*

$$|\langle u | v \rangle| \leq \sqrt{\langle u | u \rangle \cdot \langle v | v \rangle} = \|u\| \cdot \|v\|$$

Věta (Trojúhelníková nerovnost). *Každá norma odvozená ze skalárního součinu splňuje:*

$$\|u + v\| \leq \|u\| + \|v\|.$$

Definice (Kolmé (ortogonální) vektory). *Vektory u, v z prostoru se skalárním součinem jsou kolmé, $u \perp v$, pokud $\langle u | v \rangle = 0$.*

Definice (Ortonormální báze). *Báze $Z = \{v_1, \dots, v_n\}$ prostoru V se skalárním součinem je ortonormální, pokud $v_i \perp v_j$ pro každé $i \neq j$ a $\|v_i\| = 1$ pro každý vektor $v_i \in Z$.*

Definice (Fourierovy koeficienty). *Nechť $Z = \{v_1, \dots, v_n\}$ je ortonormální báze prostoru V . Pro každé $u \in V$ platí: $u = \langle u | v_1 \rangle v_1 + \dots + \langle u | v_n \rangle v_n = \sum_{i=1}^n \langle u | v_i \rangle v_i$. Koeficienty $\langle u | v_i \rangle$ se nazývají Fourierovy.*

Definice (Ortogonalní doplněk). *Ortogonalní doplněk podmnožiny V prostoru se skalárním součinem W je $V^\perp = \{u \in W : (\forall v \in V)(u \perp v)\}$.*

Algorithm 3: Gram-Schmidtova ortogonalizace

Data: lineárně nezávislé vektory $x_1, \dots, x_n \in V$.

Result: ortonormální báze $z_1, \dots, z_n$ prostoru $\text{span}\{x_1, \dots, x_n\}$.

1 **for** $k := 1, \dots, n$ **do**

2 $y_k = x_k - \sum_{j=1}^{k-1} \langle x_k | z_j \rangle z_j$

▷ vypočítáme kolmici

3 $z_k = \frac{1}{\|y_k\|} y_k$

▷ normalizujeme délku na 1

4 **end**

Věta. Pro konečně generovaný prostor W se skalárním součinem a podprostor V platí:

$$(V^\perp)^\perp = V \quad \text{a} \quad \dim V + \dim V^\perp = \dim W.$$

Definice (Ortogonalní projekce). Nechť W je prostor se skalárním součinem a V je jeho podprostor s ortonormální bází $Z = (v_1, \dots, v_n)$. Potom ortogonalní projekce W na V je zobrazení $p_Z : W \rightarrow V$ definované jako $p_Z(u) = \sum_{i=1}^n \langle u | v_i \rangle v_i$.

Poznámka. Ortogonalní projekce je lineární zobrazení.

Definice (Ortogonalní matice). Matice $Q \in \mathbb{R}^{n \times n}$ je ortogonalní, pokud $Q^T Q = I_n$. Pro ortogonalní matici Q tedy platí $Q^T = Q^{-1}$.

Věta (Vlastnosti ortogonalních matic). Matice Q je ortogonalní, právě když sloupce tvoří ortonormální bázi $\mathbb{R}^n$. Součin ortogonalních matic je ortogonalní matice. Dále platí vztahy:

$$\langle Qx | Qy \rangle = \langle x | y \rangle, \quad \|Qx\| = \|x\|.$$

1.2.7 Determinanty

Definice (Determinant). Determinant matice $A \in \mathbb{K}$ je dán výrazem:

$$\det(A) = \sum_{p \in S_n} \text{sgn}(p) \prod_{i=1}^n a_{i,p(i)}, \quad (S_n \text{ je grupa permutací na množině } \{1, \dots, n\}).$$

Pro horní trojúhelníkové matice je determinant roven součtu na diagonále.

Věta (Linearita determinantu). Determinant matice je lineárně závislý na každém jejím řádku i sloupci. Tedy vzhledem ke sčítání řádků a násobení řádku skalárem. Také platí $\det(A) = \det(A^T)$.

Věta (Multiplikativnost determinantu). Pro libovolné $A, B \in \mathbb{K} : \det(AB) = \det(A) \cdot \det(B)$.

Definice (Adjungovaná matice). Pro matici $A \in \mathbb{K}$ je adjungovaná matice definována vztahem

$$\text{adj}(A)_{j,i} = (-1)^{i+j} \det(A^{i,j}).$$

Dále pro regulární matici $A \in \mathbb{K}$ platí vztah $A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$.

Věta (Laplaceův rozvoj). Nechť $A^{i,j}$ je podmatice získaná z A odstaněním i -tého řádku a j -tého sloupce, potom pro libovolné $A \in \mathbb{K}$ a jakékoli $i \in \{1, \dots, n\}$ platí, že:

$$\det(A) = \sum_{j=1}^n a_{i,j} (-1)^{i+j} \det(A^{i,j}) = \sum_{j=1}^n a_{i,j} \text{adj}(A)_{j,i}.$$

$$\begin{vmatrix} 1 & 2 & 5 \\ 2 & 3 & 0 \\ 3 & 5 & 3 \end{vmatrix} = 1 \cdot (-1)^{2+1} \begin{vmatrix} 2 & 5 \\ 3 & 3 \end{vmatrix} + 2 \cdot (-1)^{2+2} \begin{vmatrix} 1 & 5 \\ 3 & 3 \end{vmatrix} + 0 \cdot (-1)^{2+3} \begin{vmatrix} 1 & 2 \\ 3 & 5 \end{vmatrix} = -2 \cdot -19 + 3 \cdot -12 = 2.$$

Poznámka (Geometrická reprezentace determinantu). Absolutní hodnota determinantu rozhoduje, zda se daná plocha geometrického útvaru proměňuje.

- $|\det| > 1 \implies$ zvětšuje se,
- $|\det| \in (0, 1) \implies$ zmenšuje se,
- $|\det| = 1 \implies$ zachová se,
- $|\det| = 0 \implies$ zkolabuje na menší dimenzi.

1.2.8 Vlastní čísla a vlastní vektory

Definice (Vlastní číslo). Nechť V je vektorový prostor nad $\mathbb{K}$ a f je lineární zobrazení $f : V \rightarrow V$, potom vlastní číslo zobrazení f je jakékoli $\lambda \in \mathbb{K}$, pro které existuje vektor $x \in V \setminus \{0\}$, t.ž.: $f(x) = \lambda x$.

Definice (Vlastní vektor). Nechť λ je vlastní číslo, potom jemu odpovídající vlastní vektor je libovolný vektor $x \in V$, t.ž.: $f(x) = \lambda x$. Také, pokud $0 \neq x \in \ker(A - \lambda I_n)$.

Definice (Vlastní číslo a Vlastní vektor matice). Jestliže V má konečnou dimenzi n , pak f může být reprezentováno maticí $A = [f]_{XX} \in \mathbb{K}$ vzhledem k nějaké bázi X prostoru V . Vlastní číslo matice je potom $\lambda \in \mathbb{K}$ a vlastní vektor matice $x \in \mathbb{K}^n$, oba splňující $Ax = \lambda x$.

Věta. Vlastní vektory odpovídající stejnému vlastnímu číslu tvoří podprostor.

Věta. Nechť $f : V \rightarrow V$ je lineární zobrazení a $\lambda_1, \dots, \lambda_k$ jsou různá vlastní čísla f a $u_1, \dots, u_k$ odpovídající netriviální vlastní vektory. Potom $u_1, \dots, u_k$ jsou lineárně nezávislé.

Poznámka. Platí, že rank matice je $\leq$ než počet nenulových vlastních čísel.

Poznámka. V každé matici řádu n je nejvýše n vlastních čísel.

Věta (Vlastnosti vlastních čísel). Nechť $A \in \mathbb{C}^{n \times n}$ má vlastní čísla $\lambda_1, \dots, \lambda_n$ a jim odpovídající vlastní vektory $x_1, \dots, x_n$. Pak:

- (1) A je regulární $\iff 0$ není její vlastní číslo,
- (2) je-li A regulární, pak A^{-1} má vlastní čísla $\lambda_1^{-1}, \dots, \lambda_n^{-1}$ a vlastní vektory $x_1, \dots, x_n$,
- (3) A^2 má vlastní čísla $\lambda_1^2, \dots, \lambda_n^2$ a vlastní vektory $x_1, \dots, x_n$,
- (4) αA má vlastní čísla $\alpha \lambda_1, \dots, \alpha \lambda_n$ a vlastní vektory $x_1, \dots, x_n$,
- (5) $A + \alpha I_n$ má vlastní čísla $\lambda_1 + \alpha, \dots, \lambda_n + \alpha$ a vlastní vektory $x_1, \dots, x_n$,
- (6) A^T má vlastní čísla $\lambda_1, \dots, \lambda_n$, ale vlastní vektory obecně jiné.

Definice (Charakteristický polynom). Charakteristický polynom $A \in \mathbb{K}$ je $p_A(t) = \det(A - tI_n)$.

Věta. Číslo $\lambda \in \mathbb{K}$ je vlastním číslem matice $A \in \mathbb{K} \iff \lambda$ je kořenem charakteristického polynomu $p_A(\lambda)$. Tedy když $\det(A - \lambda I_n) = 0$.

Definice (Algebraická násobnost). Algebraická násobnost vlastního čísla λ je násobnost λ^* jako kořene charakteristického polynomu $p_A(\lambda)$.

Definice (Geometrická násobnost). Geometrická násobnost vlastního čísla λ je dimenze (pod)prostoru jeho vlastních vektorů. Tedy je rovna $n - \text{rank}(A - \lambda^* I_n)$.

Definice (Podobné matice). Matice $A, B \in \mathbb{K}$ jsou si podobné, pokud existuje regulární matice R taková, že $A = RBR^{-1} \rightsquigarrow AR = RB$.

Definice (Diagonalizovatelná matice). *Matice podobná diagonální matici je diagonalizovatelná.*

Věta. *Matice A je podobná diagonální $\iff$ prostor $\mathbb{K}^n$ má bázi z vlastních vektorů A .*

Věta. *Pro jakoukoli symetrickou matici $A \in \mathbb{K}$ s $\text{char}(\mathbb{K}) \neq 2$ existuje regulární matice R taková, že $R^T A R$ je diagonální.*

Definice (Spektrální rozklad). *Diagonalizovatelná matice A lze vyjádřit ve tvaru $A = S \Lambda S^{-1}$, kde S je regulární a Λ je diagonální.*

Věta (Spektrální rozklad symetrických matic). *Pro každou symetrickou matici $A \in \mathbb{R}^{n \times n}$ existuje ortogonální $Q \in \mathbb{R}^{n \times n}$ a diagonální $\Lambda \in \mathbb{R}^{n \times n}$ takové, že $A = Q \Lambda Q^T$.*

1.2.9 Positivně semidefinitní a pozitivně definitní matice

Definice (Positivně definitní matice). *Pokud symetrická $A \in \mathbb{R}^{n \times n}$ splňuje*

$$\forall x \in \mathbb{R}^n \setminus \{0\} : x^T A x > 0,$$

pak je matice pozitivně definitní. Je semidefinitní, pokud platí $\geq$.

Věta (Vlastnosti pozitivně definitních matic). *Nechť $A, B \in \mathbb{R}^{n \times n}$ jsou pozitivně definitní matice. Pak platí:*

1. *$A + B$ je pozitivně definitní matice,*
2. *Pro libovolné $\alpha > 0$ je αA pozitivně definitní matice,*
3. *Je-li A pozitivně definitní, pak je regulární a A^{-1} je též pozitivně definitní.*

Věta (Charakterizace PD). *Pro symetrickou matici A jsou následující podmínky ekvivalentní:*

1. *A je pozitivně definitní*
2. *A má všechna vlastní čísla kladná (resp. nezáporná pro semi)*
3. *Existuje regulární matice U , t.ž.: $A = U^T U$.*

Věta (Choleskeho rozklad³). *Pro každou pozitivně definitní matici A existuje unikátní horní trojúhelníková matice U s kladnou diagonálou, t.ž.: $A = U^T U$. Matice U se nazývá Choleského rozklad.*

Věta (Sylvestrovo kritérium). *Symetrická $A \in \mathbb{R}^{n \times n}$ je pozitivně definitní $\iff$ determinanty hlavních vedoucích podmatic $A_1, \dots, A_n$ jsou kladné.*

³ Věta je existenčního charakteru. Prakticky se jedná se o metodu na testování PD matic, slouží k tomu algoritmus.

1.3 Diskrétní matematika

1.3.1 Relace

Definice (Relace mezi množinami). Relace $R \subseteq X \times Y$ je podmnožina kartézského součinu dvou množin X a Y .

Definice (Relace na množině). Relace R na X je podmnožina kartézského součinu dvou identických množin, tj. $X = Y \implies R \subseteq X \times X$.

Definice (Vlastnosti binárních relací). Pro relaci R na X definujeme:

- **Reflexivita:** $\forall x \in X : xRx$.
- **Symetrie:** $\forall x, y \in X : xRy \iff yRx$. (Také: $R = R^{-1}$)
- **Antisymetrie:** $\forall x, y \in X : xRy \wedge yRx \implies x = y$.
- **Transitivita:** $\forall x, y, z \in X : xRy \wedge yRz \implies xRz$. (Také: $R \circ R \subseteq R$)

1.3.2 Ekvivalence a rozkladové třídy

Definice (Ekvivalence). Relace R na X je ekvivalentní $\iff$ je reflexivní, symetrická a tranzitivní.

Definice (Ekvivalenční třída). Ekvivalenční třídu definujeme jako $R[x] = \{y \in X \mid xRy\}$.

1.3.3 Částečná uspořádání

Definice (Uspořádání). Relace R na X je uspořádání $\iff$ je reflexivní, antisymetrická a tranzitivní.

Definice (Lineární uspořádání). Uspořádání je lineární $\iff \forall x, y \in X : xRy \vee yRx$. Neboli, prvky jsou porovnatelné (=trichomické).

Definice (Částečné uspořádání). Uspořádání je částečné, když není lineární.

Definice (Ostré uspořádání). Uspořádání je ostré $\iff$ je ireflexivní – žádný prvek není v relaci sám se sebou.

Definice (Minimální a maximální prvek). Nechť $(X, \preceq)$ je ČUM. Potom $a \in X$ je maximální prvek, pokud $\nexists x \in X$, pro které $x \succ a$. (resp. minimální $x \prec a$).

Definice (Nejmenší a největší prvek). Nechť $(X, \preceq)$ je ČUM. Potom $a \in X$ je největší prvek, pokud $\forall x \in X$ platí $a \succeq x$. (resp. nejmenší $a \preceq x$).

Definice (Řetězec a antiřetězec). Nechť $(X, \preceq)$ je ČUM a $A \subseteq X$, potom pro:

- Řetězec platí, že $\forall a, b \in A$ jsou porovnatelné.
- Antiřetězec platí, že $\nexists a, b \in A$, které jsou různé a porovnatelné.

Definice (Výška uspořádání). Výška uspořádání v P : $\omega(P) = \max\{P\}$. (maximum z délek řetězců)

Definice (Šířka uspořádání). Šířka uspořádání v P : $\alpha(P) = \max\{|A|; A \text{ nezávislá v } P\}$. (maximum z délek antiřetězců)

Věta (O dlouhém a širokém). Nechť $(X, \preceq)$ je konečná ČUM, potom $\alpha(X, \preceq) \cdot \omega(X, \preceq) \geq |X|$

1.3.4 Funkce

Definice (Funkce). Funkce $f : X \rightarrow Y$ je relace $f \subseteq X \times Y$ taková, že $(\forall x \in X)(\exists! y \in Y) : xfy$.

Definice (Prosté/injektivní zobrazení). Funkce $f : X \rightarrow Y$ je prostá $\iff$ pro všechna Y existuje nejvýše jedno x , $\forall x \in X$. (nebo $\forall x, y : f(x) = f(y) \implies x = y$).

Definice (Na/surjektivní zobrazení). Funkce $f : X \rightarrow Y$ je na $\iff$ pro všechna Y existuje alespoň jedno x , $\forall x \in X$. (nebo $\forall y \exists x : f(x) = y$).

Definice (Bijekce). Funkce $f : X \rightarrow Y$ je bijektivní $\iff$ je prostá i na.

Věta. Nechť A je n -prvková a B je m -prvková množina, potom počet funkcí mezi A a B je m^n .

Věta. Nechť A , $|A| = n$ a B je m -prvková množina, potom počet prostých funkcí mezi A a B je m^n .

Věta. Počet všech n -prvkových podmnožin je roven 2^n , tedy $|2^X| = 2^{|X|}$.

1.3.5 Permutace

Definice (Permutace). Permutace je bijekce $\pi : [n] \rightarrow [n]$.

Definice (Pevný bod). Pro permutaci π a prvek x je pevný bod $\pi(x) = x$. Zobrazí se sám na sebe.

Věta. Pokud A , $|A| = n$ je konečná množina, tak permutace A je bijekce z A do A , tedy $n!$.

1.3.6 Kombinační čísla

Definice (Množina všech k -prvkových podmnožin). Nechť N je množina. Potom $\binom{N}{k}$ je množina všech k -prvkových podmnožin množiny N .

$$\binom{N}{k} = \{A \subseteq N : |A| = k\}$$

Definice (Kombinační číslo). Pro čísla $n, k > 0$ platí:

$$\binom{n}{k} = \frac{n!}{k!} = \frac{n \cdot (n-1) \cdot \dots \cdot (n-k+1)}{1 \cdot 2 \cdot \dots \cdot k} = \frac{n!}{k! \cdot (n-k)!}$$

Věta. Počet uspořádaných k -tic bez opakování a k -prvkových podmnožin je roven $\binom{n}{k}$.

Věta (Vlastnosti kombinačních čísel).

$$\begin{aligned}\binom{n}{0} &= 1 = \binom{n}{n} \\ \binom{n}{1} &= n = \binom{n}{n-1} \\ \binom{n}{k} &= \binom{n}{n-k} = \binom{n-1}{k} + \binom{n-1}{k-1} \\ \sum_{k=0}^n \binom{n}{k} &= 2^n\end{aligned}$$

Věta (Binomická). $(\forall x, y \in \mathbb{R})(\forall n \in \mathbb{N}) : (x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$

1.3.7 Princip inkluze a exkluze

Princip inkluze a exkluze

Věta (Princip inkluze a exkluze #1). Pro konečné A_1 až A_n platí:

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{k=1}^n (-1)^{k+1} \sum_{I \in \binom{[n]}{k}} \left| \bigcap_{i \in I} A_i \right|$$

Důkaz. Nechť $A := \bigcup_i A_i$

Levá i pravá strana jsou součty velikostí nějakých množin, takže se můžeme ptát, kolikrát levé a pravé straně přispěje každý prvek $a \in A$. Víme, že k levé přispěje jednou, chceme dokázat, že k pravé také jednou. Zdefinujme si kolikrát se započítá # i . Nechť $t = \#i : a \in A_i : \sum_{I \in \binom{[n]}{k}} \left| \bigcap_{i \in I} A_i \cap \{a\} \right|$.

Pro $k > t$: 0-krát. Pro $k \leq t$: $(-1)^{k+1} \binom{t}{k}$ -krát. To nám dá celkem $\sum_{k=1}^t (-1)^{k+1} \binom{t}{k} = -(0 - 1) = 1$. □

Aplikace (Problém šatnářky). Do divadla přišlo n pánů s n klobouky, každý pán si odložil klobouk v šatně a po představení si jej zase vyzvedl. Šatnářka však pánům vybrala klobouky náhodně. Jaká je pravděpodobnost, že žádný pán nedostal svůj klobouk?

Pravděpodobnost, že žádný pán nedostane svůj klobouk je $\frac{\text{\#permutací bez pevného bodu}}{\text{\#permutací}}$. Víme, že # počet permutací = $n!$. Nejprve spočítáme #počet permutací s pevným bodem. Pro každý prvek $i \in [n]$ definujeme $A_i = \{\pi \mid \pi(i) = i\}$ (permutace s pevným bodem i). Pak platí, že $|A_i| = (n-1)!$, a když se podíváme na velikost průniku množin $A_1, A_2, \dots, A_k$, tak je to $(n-k)!$. Z PIE:

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{k=1}^n (-1)^{k+1} \sum_{I \in \binom{[n]}{k}} (n-k)! = \sum_{k=1}^n (-1)^{k+1} \binom{n}{k} (n-k)! = \sum_{k=1}^n (-1)^{k+1} \frac{n!}{k!}.$$

Dosadíme do počtu permutací bez pevného bodu: $|\pi_s| = n! - \sum_{k=1}^n (-1)^{k+1} \frac{n!}{k!} \approx \frac{n!}{e}$. Pravděpodobnost,

že žádný pán nedostal svůj klobouk je tedy $\frac{n! \cdot e^{-1}}{n!} = \frac{1}{e}$.

Věta (Počet surjekcí). Počet surjekcí je $\frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^n$.

Důkaz. Máme množiny A , $|A| = n$ a B , $|B| = k$. Počet surjekcí $f : A \rightarrow B$ určíme za pomoci PIE. Víme, že počet všech funkcí mezi množinami je $|B^A| = k^n$. Když z množiny B^A odebereme jeden prvek, dostaneme $(k-1)^n$, když odebereme dva, dostaneme $(k-2)^n$ atd. Obecně nám to pro i odebraných prvků dává $(k-i)^n$.

Dále chceme počítat jen čistě surjektivní zobrazení, nikoliv bijektivní, které nastane v případě $n = k$. Konkrétně: Nechť $g : A \rightarrow B$ a nechť Φ je počet tohoto zobrazení. Množinu A rozdělíme do k částí $A_1, \dots, A_k$, kde pro každou tuto část a existuje přiřazení do množiny B . Nastává tedy bijekce z A do B , kde počet bijektivních zobrazení odpovídá $k!$. Celkem bychom dostali " $\Phi = k! \times \text{počet } f : A \rightarrow B$ ", kde f je zbylá (námi hledaná) funkce na.

$$\frac{1}{k!} \cdot \left(k^n - \binom{k}{1} (k-1)^n + \binom{k}{2} (k-2)^n - \dots \pm \underbrace{\binom{k}{k} (k-k)^n}_0 \right) = \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^n$$

□

Definice (Eulerova funkce). Zobrazení $\varphi : \mathbb{N} \rightarrow \mathbb{N}$ značí pro $n \in \mathbb{N}$ počet čísel $k \in \{1, \dots, n-1\}$ nesoudělných s číslem n . Tedy jinak $\varphi(n) = |\{k \in \{1, \dots, n-1\} \mid \gcd(k, n) = 1\}|$.

Výpočet eulerovy funkce za pomoci principu inkluze a exkluze Necht' $n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$ je rozklad n na prvočinitele. Aplikujeme PIE: Počet čísel $\leq n$ nesoudělných s n získáme odečtením počtů násobků všech prvočísel p_i a přičtením zpět násobků jejich dvojic atd.:

$$\varphi(n) = n - \sum_{i=1}^m \frac{n}{p_i} + \sum_{1 \leq i < j \leq m} \frac{n}{p_i p_j} - \dots + (-1)^m \frac{n}{p_1 p_2 \dots p_m}.$$

Nakonec vytkneme n a tím získáme:

$$\varphi(n) = n \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \dots \left(1 - \frac{1}{p_m}\right).$$

1.3.8 Hallova věta

Definice (Párování). Párování v grafu $G = (V, E)$ je množina hran $M \subseteq E$, t.ž. každý vrchol patří do nejvýše jedné hrany z M .

Definice (Vrcholové prokrytí). Vrcholové prokrytí v grafu $G = (V, E)$ je množina vrcholů $C \subseteq V$, t.ž. každá hrana obsahuje alespoň jeden vrchol z C .

Definice (Systém různých reprezentantů). SRR v hypergrafu $H = (V, E)$ je funkce $r : E \rightarrow V$, t.ž.:

- (1) $\forall e \in E : r(e) \in e$, kde $r(e)$ je reprezentant hyperhrany e ,
- (2) $\forall e, f \in E : e \neq f \implies r(e) \neq r(f)$, tedy funkce r je prostá.

Věta (König-Egerváty). V každém bipartitním grafu má největší párování stejnou velikost, jako nejmenší vrcholové pokrytí.

Věta (Hallova). Necht' G je bipartitní graf s partitami A, B . Potom G má párování velikosti

$$|A| \iff \forall X \subseteq A : |N(X)| \geq |X|.$$

Důkaz. Musím dokázat obě implikace.

$\implies$ Pokud existuje párování velikosti $|A|$, tak pro každou $X \subseteq A$ existuje $|X|$ vrcholů spárovaných s X a ty patří do $N(X)$. Tedy $|N(X)| \geq |X|$.

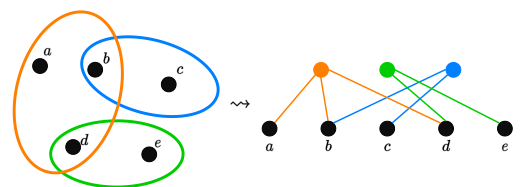
$\Leftarrow$ Pro spor. Necht' M je největší párování G , t.ž.: $|M| < |A|$. Existuje pokrytí C , kde $|C| = |M| < |A|$. Definujeme si $C_A := C \cap A$, $C_B := C \cap B$ a $X := A \setminus C_A$.

Zjistíme, že $N(X) \subseteq C_B$ a navíc, že $|X| = |A| - |C_A| > |C_B| \geq |N(X)|$, což nám dává spor. □

Věta (Hallova - hypergrafová verze). Hypergraf $H = (V, E)$ má SRR $\iff \forall F \subseteq E : \left| \bigcup_{e \in F} e \right| \geq |F|$.

Důkaz. Necht' $H = (V, E)$ je hypergraf a I_H jeho graf incidence. H má SRR $\iff I_H$ má párování velikosti $|E|$.

Hallova podmínka pro $H \iff \forall F \subseteq E : \left| \bigcup_{e \in F} e \right| \iff$ bipartitní Hallova podmínka pro I_H a partitu E . Ekvivalence platí díky bipartitní Hallově podmínce. □



Algoritmus (Nalezení SRR v polynomiálním čase). Modelujeme bipartitní graf. Levá partita jsou množiny $S_1, \dots, S_m$. Pravá partita prvky $\bigcup_i S_i$. Hrana vede z $S_i \rightarrow x$, pokud $x \in S_i$. Najdeme max párování v bipartitním grafu (třeba Ford-Fulkerson – polytime) a z Hallové věty: SRR existuje $\iff$ párování pokrývá všechny množiny S_i .

1.4 Teorie grafů

1.4.1 Základní pojmy teorie grafů

Definice (Graf). Graf G je uspořádaná dvojice (V, E) , kde V je konečná neprázdná množina vrcholů a $E \subseteq \binom{V}{2}$ konečná neprázdná množina hran, neboli všech dvouprvkových podmnožin V .

Definice (Isomorfismus grafů). Grafy G a H jsou isomorfní, pokud existuje bijekce mezi vrcholy:

$$\exists f : V_G \rightarrow V_H \quad , \quad \text{t.ž.} \quad \{u, v\} \in E_G \iff \{f(u), f(v)\} \in E_H$$

Definice (Podgraf). Graf H je podgrafem grafu G , pokud $V_H \subseteq V_G$ a $E_H \subseteq E_G \cap \binom{V_H}{2}$.

Definice (Indukovaný podgraf). Podgraf H je indukovaný, pokud $E(H) = E(G) \cap \binom{V(H)}{2}$.

Definice (Stupeň vrcholu). Stupeň vrcholu ' v ' v grafu $G = (V, E)$ je

$$\deg_G(v) := |\{u \in V : \{u, v\} \in E\}|.$$

Neboli počet hran grafu G , které obsahují hranu s v .

Definice (k -regulární graf). Graf je k -regulární, pokud pro $k \in \mathbb{N}$ platí $\forall u \in V(G) : \deg_G(u) = k$.

Definice (Skóre grafu). Skóre grafu G je posloupnost stupňů všech vrcholů (krom uspořádání).

Definice (Doplňek grafu). Doplňek grafu $G = (V, E)$ je graf $\bar{G} = (V, \binom{V}{2} \setminus E)$.

1.4.2 Základní příklady grafů

Definice (Úplný graf). Úplný graf na n vrcholech značíme K_n , kde $V = [n]$ a $E = \binom{V}{2}$.

Definice (Bipartitní graf). Graf G je bipartitní, pokud lze V rozdělit na dvě disjunktní množiny V_1 a V_2 takové, že každá hrana z E obsahuje jeden bod z V_1 a druhý z V_2 .

Tedy pokud $V = V_1 \cup V_2$, t.ž. $V_1 \cap V_2 = \emptyset$. Hrany jsou mezi V_1 a V_2 , neboli $\forall e \in E : |e \cap V_1| = 1$.

Definice (Úplný bipartitní graf). Úplný bipartitní graf na $n + m$ vrcholech značíme $K_{n,m}$, kde $V = \{u_1, \dots, u_n\} \cup \{v_1, \dots, v_m\}$ (=dvě partity) a $E = \{\{u_i, v_j\} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$.

Definice (Eulerovský graf). Graf je Eulerovský $\Leftrightarrow$ je souvislý a všechny vrcholy mají sudý stupeň.

Definice (Cesta). Cestu na n vrcholech značíme P_n , kde $V = [n]$ a $E = \{\{i-1, i\} \mid 1 \leq i \leq n\}$.

Definice (Kružnice). Kružnici na n vrcholech značíme C_n , kde $V = [n]$ a

$$E = \{\{i, i+1\} \mid 1 \leq i \leq n-1\} \cup \{\{1, n\}\}.$$

Definice (Sled). Sled z v_0 do v_n v grafu G je posloupnost $(v_0, e_1, v_1, e_2, \dots, e_n, v_n)$, pokud $\forall i$ platí $e_i = \{v_{i-1}, v_i\}$, kde v jsou vrcholy a e hrany.

(Mohou se opakovat vrcholy i hrany).

Definice (Tah). Tah z v_0 do v_n v grafu G je posloupnost $(v_0, e_1, v_1, e_2, \dots, e_n, v_n)$, pokud $\forall i$ platí $e_i = \{v_{i-1}, v_i\}$, kde v jsou vrcholy a e navzájem různé hrany.

(Mohou se opakovat pouze vrcholy, ne hrany).

1.4.3 Souvislost grafů

Definice (Souvislý graf). *Graf je souvislý, pokud $(\forall u, v \in V)$ existuje cesta z u do v .*

Definice (Relace dosažitelnosti (ekvivalence)). *Ekvivalence v grafu G je binární relace $\sim$ na $V(G)$, t.ž. $u \sim v$, pokud existuje cesta z u do v .*

Definice (Komponenty souvislosti). *Komponenty souvislosti jsou podgrafy indukované třídami ekvivalence.*

Definice (Vzdálenost v grafu). *Vzdálenost v souvislém grafu G je definována jako $d_G : V^2 \rightarrow \mathbb{R} : \forall u, v : d_G(u, v)$ je minimum z délek mezi u a v .*

Pro metriku musí platit $\forall u, v, w \in V$:

- $d_G(u, v) \geq 0$... je minimum z délek cest, cesty jsou také nezáporné
- $d_G(u, v) = 0 \iff u = v$... nikde jinde (krom $d_G(u, u)$) vzdálenost nulová není
- $d_G(u, v) \leq d_G(u, w) \leq d_G(w, v)$... vzdálenost mezi u a v je shora omezená mezi vzdáleností u, w a w, v
- $d_G(v, u) = d_G(u, v)$

1.4.4 Stromy

Definice (Strom). *Strom je souvislý graf bez kružnic. (acyklický graf)*

Definice (Les). *Les je acyklický graf. Jeho komponenty souvislosti jsou stromy.*

Definice (List). *List je vrchol stupně 1.*

Věta. *Každý strom s alespoň dvěma vrcholy má alespoň jeden list.*

Věta. *Pro graf G s listem v platí, že G je strom $\iff G - v$ je strom.*

Věta (Eulerova formule). *Pro G souvislý máme počet hran stromu $|E(G)| = |V(G)| - 1$.*

Věta (Ekvivalentní charakteristiky). *Pro graf G jsou následující tvrzení ekvivalentní:*

- G je souvislý a acyklický (=strom)
- $\forall u, v \in V(G) \exists!$ cesta v G mezi u a v (=jednoznačná souvislost)
- G je souvislý a $\forall e \in E(G) : G - e$ není souvislý (=minimální souvislost)
- G je acyklický a $\forall e \in \binom{V(G)}{2} \setminus E : G + e$ má cyklus (=maximálně acyklický)
- G je souvislý a $|E(G)| = |V(G)| - 1$ (=Eulerova formule)

Věta. *V grafu $G = (V, E)$ platí:*

$$\sum_{v \in V} \deg_G(v) = 2|E|$$

1.4.5 Rovinné grafy

Definice (Rovinný graf). *Rovinný graf je takový graf, pro nějž existuje nějaké nakreslení v rovině.*

Definice (Rovinné nakreslení). *Rovinné nakreslení G je přiřazení různých bodů v rovině různým vrcholům G , spojené s přiřazením oblouků každé hraně $z E(G)$ – aby bylo nakreslení rovinné, tak žádné dva oblouky nesmí sdílet jeden bod v rovině, jediné ten koncový.*

Definice (Stěna). *Stěna je oblast ohraničená hranami.*

Věta (Kuratowski). *Graf G je nerovinný $\iff G$ obsahuje podgraf izomorfní s dělením K_5 nebo $K_{3,3}$.*

Věta (Eulerova formule). *Nechť G je souvislý graf nakreslený do roviny, $v := |V(G)|$, $e := |E(G)|$, $f := \# \text{stěn nakreslení}$. Potom platí $v - e + f = 2$. (pro c počet komponent: $v - e + f = 1 + c$.)*

Důkaz. Zvolíme v pevně a pak indukci podle e .

(i) $e = v - 1$ (G je strom), $f = 1$:

$$v + 1 = v - 1 + 2$$

(ii) $e - 1 \rightarrow e$: mějme graf G s e hranami. Nechť λ je hrana na kružnici v G .

Potom $G' = G - \lambda$, $v' = v$, $e' = e - 1$, $f' = f - 1$. Nyní použijeme indukční předpoklad:

$$\begin{aligned} v' + f' &= e' + 2 \\ v + f - 1 &= e - 1 + 2 \\ v + f &= e + 2 \end{aligned}$$

□

Věta (Maximální počet hran rovinného grafu). *V každém rovinném grafu s alespoň 3 vrcholy platí:*

$$|E| \leq 3|V| - 6.$$

Důkaz. Doplním do G hrany, až získám maximální rovinný graf G' . Takže $v' = v$, $e' \geq e$. Každá stěna přispěje třemi hranami, každá hrana dvěma stěnám, tedy $3f = 2e \rightsquigarrow f = \frac{2}{3}e$. Nyní jen dosadíme:

$$v' - e' + \frac{2}{3}e = 2 \iff 3v - e' + 2e = 6 \implies e \leq 3v - 6$$

□

Věta. *V rovinném grafu existuje vrchol stupně nejvýše 5.*

Věta (O 4 barvách). *Každý rovinný graf lze obarvit nejvýše 4 barvami.*

1.4.6 Barevnost grafů

Definice (k -obarvení grafu). *Obarvení grafu k barvami je $c : V(G) \rightarrow [k]$ tak, že kdykoli $\{x, y\} \in E(G)$, pak $c(x) \neq c(y)$.*

Definice (Barevnost grafu). *Barevnost $\chi(G)$ je nejmenší k takové, že existuje k -obarvení G .*

Věta. $\chi(G) \leq 2 \iff G$ je bipartitní $\iff G$ nemá lichou kružnici.

Definice (Klikovost). *Klikovost $\omega(G)$ je rovna velikosti největší kliky (úplného podgrafu K_k) v G .*

Věta (Barevnost a klikovost). $\chi(G) \geq \omega(G)$.

1.4.7 Hranová a vrcholová souvislost grafů

Definice (Hranový řez). $F \subseteq E$ je hranový řez v G pokud $G \setminus F$ je nesouvislý.

Definice (Hranová k -souvislost). G je hranově k -souvislý, pokud neobsahuje žádný hranový řez velikosti menší než k .

Definice (Vrcholový řez). $U \subseteq V$ je vrcholový řez v G pokud podgraf indukovaný $V \setminus U$ je nesouvislý.

Definice (Vrcholová k -souvislost). Graf G je vrcholově k -souvislý, pokud má alespoň $k + 1$ vrcholů a neobsahuje žádný vrcholový řez velikosti $< k$.

Definice (Vrcholová souvislost). Vrcholová souvislost grafu G , značeno $K_v(G)$, je největší k , t.ž.: G je vrcholově k -souvislý.

Věta (Menger – hranová xy verze). Pro dva různé vrcholy x, y grafu G platí, že G obsahuje $k \in \mathbb{N}$ hranově disjunktích cest z x do $y \iff G$ neobsahuje hranový xy -řez velikosti menší než k .

Věta (Menger – vrcholová xy verze). Pro dva různé nesousední vrcholy x, y grafu G platí, že G obsahuje $k \in \mathbb{N}$ navzájem vrcholově disjunktích cest z x do $y \iff G$ neobsahuje vrcholový xy -řez velikosti $< k$.

Věta (Menger – hranová verze). Graf G je hranově k -souvislý $\iff$ mezi každými dvěma různými vrcholy existuje k hranově disjunktích cest.

Věta (Menger – vrcholová verze). Graf G je vrcholově k -souvislý $\iff$ mezi každými dvěma různými vrcholy existuje k navzájem vnitřně vrcholově disjunktích cest.

1.4.8 Orientované grafy

Definice (Otevřený eulerovský tah). Otevřený eulerovský tah z v_0 do v_n je takový tah, který obsahuje všechny vrcholy a hrany grafu právě jednou.

Definice (Uzavřený eulerovský tah). Uzavřený eulerovský tah je takový eulerovský tah, kde $v_0 = v_n$.

Definice (Orientovaný graf). Orientovaný graf je uspořádaná dvojice (V, E) , kde

$$E \subseteq V^2 \setminus \{(x, x) \mid x \in V\}.$$

Definice (Podkladový graf). Podkladový graf grafu $G = (V, E)$ je graf $H = (V, F)$, kde

$$F = \left\{ \{u, v\} \in \binom{V}{2} \mid (u, v) \in E \vee (v, u) \in E \right\}.$$

Neboli množina všech neuspořádaných dvojic vrcholů, kde v jednom nebo druhém pořadí je hrana.

Definice (Silná souvislost). Pokud pro $\forall u, v \in V$ existuje orientovaná cesta z u do v .

Definice (Slabá souvislost). Pokud podkladový graf je souvislý.

Věta. Pro orientovaný graf G je ekvivalentní:

- (i) je vyvážený a slabě souvislý
- (ii) má uzavřený eulerovský tah
- (iii) je vyvážený a silně souvislý

1.4.9 Toky v sítích

Definice (Toková síť). Je pětice (V, E, z, s, c) :

- $V \equiv$ množina vrcholů
- $E \equiv$ množina orientovaných hran $E \subseteq V \times V$
- $z \in V \equiv$ zdroj
- $s \in V \setminus \{z\} \equiv$ stok/spotřebič
- $c : E \rightarrow [0, +\infty) \equiv c(e)$ je kapacita hrany e

Definice (Tok). V síti (V, E, z, s, c) je funkce $f : E \rightarrow [0, +\infty)$ splňující:

- (i) $\forall e \in E : 0 \leq f(e) \leq c(e)$,
- (ii) $\forall u \in V \setminus \{z, s\} : \sum_{v:uv \in E} f(uv) = \sum_{v:vu \in E} f(vu)$, respektive $f_{In} = f_{Out}$.

Definice (Velikost toku). Velikost toku f v síti (V, E, z, s, c) je $w(f) := f[Out(z)] - f[In(z)]$.

Definice (Maximální tok). Maximální tok je takový tok, který má největší velikost.

Definice (Řez). v síti (V, E, z, s, c) je množina hran $R \subseteq E$, t.ž.: každá orientovaná cesta ze z do s má neprázdný průnik s R .

- Kapacita řezu $R \equiv c(R) = \sum_{e \in R} c(e)$
- Minimální řez je řez, který má ze všech řezů nejmenší kapacitu.

Definice (Nenasycená cesta). Nechť f je tok v síti (V, E, z, s, c) . Nenasycená cesta pro f je neorientovaná cesta $x_1 e_1 x_2 e_2 \dots x_{k-1} e_{k-1} x_k e_k x_{k+1}$, kde $\forall i = 1, \dots, k$:

- e_i je buď dopředná hrana, tedy $e_i = (x_i, x_{i+1})$, nebo
- e_i je zpětná hrana, tedy $e_i(x_{i+1}, x_i)$.

Zároveň platí $f(e_i) < c(e_i)$ pro každou dopřednou hranu a $f(e_i) > 0$ pro každou zpětnou hranu.

Definice (Zlepšující cesta). Zlepšující cesta pro f je nenasycená cesta ze z do s .

Algorithm 4: Ford-Fulkerson – Princip hledání maximálního toku v síti.

```

1  $f \leftarrow$  nulový tok
2 while existuje zlepšující cesta  $P$  ze  $z \rightarrow s$  do
3    $\varepsilon \leftarrow \min_{e \in E(P)} r(e)$ .  $\triangleright$  Rezerva  $r(uv) = c(uv) - f(uv) + f(vu)$ 
4   Zvětšíme tok  $f$  podél  $P$  o  $\varepsilon$ .  $\triangleright \forall e$  po směru zvětšíme a proti směru zmenšíme  $f(e)$ 
5 end
6 return tok  $f$ .
```

Věta. Nechť f je tok v síti (V, E, z, s, c) , potom následující tvrzení jsou ekvivalentní:

- (i) f je maximální
- (ii) f nemá zlepšující cestu
- (iii) Existuje řez R , t.ž.: $w(f) = c(R)$.

Věta (Minimax). Nechť $f_{\max}$ je maximální tok a $R_{\min}$ je minimální řez v (V, E, z, s, c) , potom $w(f_{\max}) = c(R_{\min})$.

1.5 Pravděpodobnost a statistika

1.5.1 Pravděpodobnost, jevy, prostor

Definice (Pravděpodobnost). Necht' Ω je množina elementárních jevů a $\mathcal{F} \subseteq 2^\Omega$ je podmnožina elementárních jevů. Potom pravděpodobnost P je funkce $P : \mathcal{F} \rightarrow [0, 1]$ = $\begin{cases} P(A) = 1 & \text{jev jistý} \\ P(A) = 0 & \text{jev nemožný} \end{cases}$

Definice (Diskrétní pravděpodobnostní prostor). Diskrétní pravděpodobnostní prostor je trojice $(\Omega, \mathcal{F}, P)$, kde Ω je konečná nebo spočetná, $\mathcal{F} = 2^\Omega$; $P(\Omega) = 1$; a $P(A) = \sum_{w \in A} P(\{w\})$.

Definice (Konečný pravděpodobnostní prostor). Konečný pravděpodobnostní prostor je diskrétní, kde Ω je konečná.

Definice (Klasický pravděpodobnostní prostor). Klasický pravděpodobnostní prostor je konečný, kde $P(A) = \frac{|A|}{|\Omega|}$.

Definice (Elementární jev). Všechny výsledky nějakého pravděpodobnostního experimentu. Značíme jako Ω .

Definice (Složený jev). Složený jev je takový jev, který není elementární. Složený jev nastane $\iff$ nastane některý z elementárních jevů v něm obsažený.

Definice (Pravděpodobnost jevu). Pravděpodobnost jevu udává, jakou máme šanci, že daný jev nastane.

Věta (Základní pravidla pro počítání). V pravděpodobnostním prostoru $(\Omega, \mathcal{F}, P)$ platí pro $A, B \in \mathcal{F}$:

- $A \subseteq B \implies P(A) \leq P(B)$,
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$,
- $P(A_1 \cup A_2 \cup \dots) \leq \sum_i P(A_i)$.

Definice (Nezávislost náhodných jevů). Jevy A a B jsou nezávislé $\iff P(A \cap B) = P(A) \cdot P(B)$.

Definice (Podmíněná pravděpodobnost). Podmíněná pravděpodobnost je pravděpodobnost, že nastal jev A za podmínky, že nastal jev B .

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Věta (Bayesova věta). Necht' A je jev, kde $P(A) \neq 0$ a $B_1, \dots, B_k$ je rozklad Ω na jevy t.ž. $\forall i : P(B_i) \neq 0$. Potom:

$$P(B_i|A) = \frac{P(A|B_i) \cdot P(B_i)}{\sum_j P(A|B_j) \cdot P(B_j)}$$

1.5.2 Náhodné veličiny a jejich rozdělení

Definice (Náhodná veličina). Náhodná veličina je zobrazení $X : \Omega \rightarrow \mathbb{R}$, které $\forall x \in \mathbb{R}$ splňuje:

$$\{\omega \in \Omega : X(\omega) \leq x\} \in \mathcal{F}.$$

Definice (Diskrétní náhodná veličina). Funkci $X : \Omega \rightarrow \mathbb{R}$ nazveme diskrétní náhodná veličina, pokud je obor hodnot X spočetná množina a pokud $\forall x \in \mathbb{R}$ platí: $\{\omega \in \Omega : X(\omega) = x\} \in \mathcal{F}$.

Definice (Spojitá náhodná veličina). Náhodná veličina X se nazývá spojitá, pokud existuje nezáporná reálná funkce f_X tak, že

$$F_X(x) = P(X \leq x) = \int_{-\infty}^x f_X(t) dt.$$

Definice (Indikátorová náhodná veličina). Pro libovolný jev $A \in \mathcal{F}$ definujeme indikátorovou n.v. I_A : $I_A(\omega) = 1$ pokud $\omega \in A$, $I_A(\omega) = 0$ jinak.

Definice (Bernoulliho rozdělení). Pro $X \sim I_A \sim \text{Bern}(p)$ je

- $\mathbb{E}(X) = p$,
- $\text{var}(X) = p(1 - p)$

(X může být například počet orlů při jednom hodu nespravedlivou mincí.)

Definice (Geometrické rozdělení). Pro $X \sim \text{Geom}(p)$ je

- $\mathbb{E}(X) = \frac{1}{p}$,
- $\text{var}(X) = \frac{1-p}{p^2}$

Pro dané $p \in [0, 1]$ máme $p_X(k) = (1 - p)^{k-1}p$, pro $k \in \mathbb{N}$.

(X může být například kolikátým hodem mincí padl první orel.)

Definice (Binomické rozdělení). Pro $X \sim \text{Bin}(n, p)$ je

- $\mathbb{E}(X) = np$,
- $\text{var}(X) = np(1 - p)$

Pro dané $p \in [0, 1]$ máme $p_X(k) = \binom{n}{k} p^k (1 - p)^{n-k}$, pro $k \in \{0, \dots, n\}$.

(X může být například počet orlů při n hodech nespravedlivou mincí.)

Definice (Hypergeometrické rozdělení). Pro $X \sim \text{Hyper}(N, K, n)$ je

- $\mathbb{E}(X) = n \frac{K}{N}$,
- $\text{var}(X) = n \frac{K}{N} (1 - \frac{K}{N}) \frac{N-n}{N-1}$.

Pro dané $p \in [0, 1]$, n, N, K máme $p_X(k) = P(X = k) = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}$.

(X může být například počet vytažených červených míček při n tazích, v osudí je K červených z N celkových míčeků.)

Definice (Poissonovo rozdělení). Pro $X \sim \text{Pois}(\lambda)$ je

- $\mathbb{E}(X) = \lambda$,
- $\text{var}(X) = \lambda$.

Pro dané $p \in [0, 1]$ a reálné $\lambda > 0$ máme $p_X(k) = \frac{\lambda^k}{k!} e^{-\lambda}$. Platí, že $\text{Pois}(\lambda)$ je limitou $\text{Bin}(n, \lambda/n)$.

(X může být například počet emailů, které dostaneme za jednu hodinu.)

Definice (Normální rozdělení). Pro $X \sim N(0, 1)$ je

- $\mathbb{E}(X) = 0$,

- $\text{var}(X) = 1$.

Platí, že $f_X = \varphi$, kde $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$.

Definice (Exponenciální rozdělení). Pro $X \sim \text{Exp}(\lambda)$ je

- $\mathbb{E}(X) = \frac{1}{\lambda}$,
- $\text{var}(X) = \frac{1}{\lambda^2}$.

$$F_X(x) = \begin{cases} 0 & \text{pro } x \leq 0 \\ 1 - e^{-\lambda x} & \text{pro } x \geq 0 \end{cases} \quad f_X(x) = \begin{cases} 0 & \text{pro } x \leq 0 \\ \lambda e^{-\lambda x} & \text{pro } x \geq 0 \end{cases}$$

(X může být například čas před příchodem dalšího telefonního hovoru do call-centra/ dotazu na web-server/čas do dalšího blesku v bouři apod.)

Definice (Distribuční funkce). Distribuční funkce n.v. X je funkce

$$F_X(x) := P(X \leq x) = P(\{\omega \in \Omega : X(\omega) \leq x\}).$$

(1) F_X je neklesající a zprava spojitá funkce

$$(2) \lim_{x \rightarrow -\infty} F_X(x) = 0$$

$$(3) \lim_{x \rightarrow +\infty} F_X(x) = 1$$

Střední hodnota

Definice (Střední hodnota). Střední hodnota náhodné veličiny X je $\mathbb{E}(X) = \sum_{w \in \Omega} P(\{w\}) \cdot X(w)$.

Věta (Linearita střední hodnoty). Necht' X, Y jsou nezávislé náhodné veličiny a $\alpha \in \mathbb{R}$, potom

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y] \quad a \quad \mathbb{E}[\alpha X] = \alpha \mathbb{E}[X].$$

Věta. Necht' X, Y jsou nezávislé diskrétní náhodné veličiny, potom $\mathbb{E}[X \cdot Y] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$.

Definice (Markovova nerovnost). Necht' X je nezáporná náhodná veličina a $\forall t \geq 1$, potom platí:

$$P[X \geq t \cdot \mathbb{E}(X)] \leq \frac{1}{t}.$$

Rozptyl

Definice (Rozptyl). Rozptyl n.v. X , značeno $\text{var}(X)$, nazveme číslo $\mathbb{E}((X - \mathbb{E}X)^2)$.

Věta. $\text{var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)^2$

Věta (Rozptyl součtu). Pro $X_1, \dots, X_n$ nezávislé diskrétní nebo spojitě n.v. platí

$$\text{var}(X_1 + \dots + X_n) = \text{var}(X_1) + \dots + \text{var}(X_n).$$

Definice (Spojitý rozptyl). Rozptyl spojitě n.v. X je $\text{var}(X) := \mathbb{E}((X - \mu)^2) = \int_{-\infty}^{\infty} (x - \mu)^2 f_X(x) dx$.

1.5.3 Limitní věty

Věta (Zákon velkých čísel). *Nechť $X_1, \dots, X_n$ jsou stejné rozdělené n.n.v. se střední hodnotou μ a rozptylem σ^2 . Označme $S_n = (X_1 + \dots + X_n)/n$. Pak pro každé $\varepsilon > 0$ platí*

$$\lim_{n \rightarrow \infty} P(|S_n - \mu| \geq \varepsilon) = 0.$$

Říkáme, že posloupnost S_n konverguje k μ v pravděpodobnosti, píšeme $S_n \xrightarrow{P} \mu$.

Věta (Centrální limitní věta). *Nechť $X_1, \dots, X_n$ jsou stejně rozdělené n.n.v. se střední hodnotou μ a rozptylem σ^2 . Označme*

$$Y_n = ((X_1 + \dots + X_n) - n\mu)/(\sqrt{n} \cdot \sigma).$$

Pak $Y_n \xrightarrow{d} N(0, 1)$. Neboli, pokud F_n je distribuční funkce Y_n , tak

$$\lim_{n \rightarrow \infty} F_n(x) = \Phi(x) \quad \forall x \in \mathbb{R}.$$

Říkáme, že posloupnost Y_n konverguje k $N(0, 1)$ v distribuci.

1.5.4 Bodové odhady

Chceme určit hodnotu nějakého parametru – často střední hodnoty neznámého rozdělení.

Definice (Bodový odhad). *Pro náhodný výběr $X_1, \dots, X_n \sim F_\theta$ a libovolnou funkci g nazveme bodový odhad $\hat{\theta}_n = g(X_1, \dots, X_n)$:*

- *nestranný, pokud $\mathbb{E}(\hat{\theta}_n) = g(\theta)$,*
- *asymptoticky nestranný, pokud $\lim_{n \rightarrow \infty} \mathbb{E}(\hat{\theta}_n) = g(\theta)$,*
- *konzistentní, pokud $\hat{\theta}_n \xrightarrow{P} g(\theta)$,*
- *vychýlení $\text{bias}(\hat{\theta}_n) = \mathbb{E}(\hat{\theta}_n) - \theta$*
- *střední kvadratickou chybu $\text{MSE} = \mathbb{E}((\hat{\theta}_n - \theta)^2)$*

Věta (Parametry výběrového momentu a rozptylu). *Máme:*

- $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ je konzistentní nestranný odhad $\mu = \mathbb{E}X_1 = \mathbb{E}X_2 = \dots$
- $\bar{S}_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X}_i)^2$ je konzistentní asymptoticky nestranný odhad σ^2
- $\hat{S}_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_i)^2$ je konzistentní nestranný odhad σ^2

Metoda momentů. Volíme takové θ , které řeší soustavu rovnic

$$m_r(\theta) = \widehat{m_r}(\theta) \quad r = 1, \dots, k$$

Nechť r -tý moment náhodné veličiny X je $\mathbb{E}(X^r)$ a označme:

- $m_r(\theta) := \mathbb{E}(X^r)$ pro $X \sim F_\theta$ jako r -tý moment (průměr z celé populace)
- $\widehat{m_r}(\theta) := \frac{1}{n} \sum_{i=1}^n X_i^r$ jako r -tý výběrový moment (průměr z náhod. výběru $X_1, \dots, X_n$ z F_θ)

Věta. $\widehat{m_r}(\theta)$ je nestranný konzistentní odhad pro $m_r(\theta)$.

Příklad. Nechť $X_1, \dots, X_n \sim \text{Bern}(p)$, kde $X_i =$ “ i -tý člověk je levák”. Potom $\theta = p \in [0, 1]$ a $m_1(\theta) = \mathbb{E}(X_1) = \theta$. Celkem

$$\widehat{m_r(\theta)} = \frac{1}{n}(X_1 + \dots + X_n) = \bar{X}_n.$$

1.5.5 Intervalové odhady

Místo jednoho čísla s nejistým významem vypočítáme z dat interval $[\hat{\theta}^-, \hat{\theta}^+]$.

Definice (Intervalový odhad). *Intervalový odhad (konfidenční interval) je interval $[D, H]$, který s předem zvolenou pravděpodobností $(1 - \alpha)$ pokrývá neznámý parametr θ populace. Číslo α se nazývá hladina významnosti (typicky $\alpha = 0.05$).*

Metoda založená na normální aproximaci. Tato metoda využívá centrální limitní větu (CLV) a je vhodná pro velké výběry ($n \geq 30$).

(1) **Bodový odhad:** Pro střední hodnotu $\theta = \mu$ použijeme $\hat{\theta} = \bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$.

(2) **Standardní chyba (SE):**

- Pokud je známý rozptyl σ^2 , tak $\text{SE} = \frac{\sigma}{\sqrt{n}}$.
- Pokud rozptyl neznáme, odhadneme ho pomocí výběrového rozptylu:

$$\widehat{\text{SE}} = \frac{\hat{S}_n}{\sqrt{n}}, \quad \text{kde} \quad \hat{S}_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2.$$

(3) **Kvantily rozdělení:**

- Pro známý rozptyl použijeme **normální rozdělení** $N(0, 1)$: $z_{\alpha/2} = \Phi^{-1}(1 - \frac{\alpha}{2})$
- Pro neznámý rozptyl a malé výběry použijeme **Studentovo t -rozdělení** s $n - 1$ stupni volnosti:

$$t_{\alpha/2} = \Psi_{n-1}^{-1}(1 - \frac{\alpha}{2}).$$

(4) **Interval spolehlivosti:**

$$\begin{aligned} &\hat{\theta} \pm z_{\alpha/2} \cdot \text{SE} \quad (\text{pro známý rozptyl}), \\ &\hat{\theta} \pm t_{\alpha/2} \cdot \widehat{\text{SE}} \quad (\text{pro neznámý rozptyl}). \end{aligned}$$

Aplikace. Používá se například pro odhady průměrné výšky, hmotnosti, doby života.

1.5.6 Testování hypotéz

Jedná se o metodu pro rozhodování o platnosti určitých hypotéz na základě pozorovaných dat.

Základní přístup. Postupujeme následovně:

- (1) Formulujeme **nulovou hypotézu** (H_0) - konzervativní tvrzení, které testujeme (např. “mince je spravedlivá”, “lék nemá účinek”).
- (2) Stanovíme **alternativní hypotézu** (H_1) - tvrzení, které platí, pokud H_0 neplatí (např. “mince je falešná”, “lék má účinek”).
- (3) Zvolíme **testovou statistiku** $T = h(X_1, \dots, X_n)$, která měří odchylku dat od H_0 .
- (4) Určíme **kritický obor** W - množinu hodnot statistiky, pro které zamítneme H_0 .
- (5) Vypočteme pozorovanou hodnotu statistiky $t = h(x_1, \dots, x_n)$.
- (6) Rozhodneme: pokud $t \in W$, zamítneme H_0 ve prospěch H_1 ; jinak H_0 nezamítáme.

Chyby v testování. Při rozhodování můžeme udělat dva typy chyb:

- **Chyba 1. druhu:** Zamítneme H_0 , i když platí.

$$P(\text{chyba 1. druhu}) = P(T \in W | H_0) = \alpha$$

- **Chyba 2. druhu:** Nezamítneme H_0 , i když neplatí.

$$P(\text{chyba 2. druhu}) = P(T \notin W | H_1) = \beta$$

- **Síla testu:** Pravděpodobnost správného zamítnutí H_0 , když neplatí.

$$1 - \beta = P(T \in W | H_1)$$

Hladina významnosti. Hladina významnosti α je pravděpodobnost chyby 1. druhu, kterou jsme ochotni akceptovat. Typické volby jsou $\alpha = 0.05$ nebo $\alpha = 0.01$. Kritický obor W pak volíme, aby:

$$P(T \in W | H_0) = \alpha$$

Základní princip: Zamítneme H_0 , pokud p -hodnota $= P(T \geq t | H_0) < \alpha$.

1.6 Logika

1.6.1 Syntaxe

Definice (Jazyk VL). Jazyk výrokové logiky je určený neprázdnou množinou výrokových proměnných $\mathbb{P}$. Patří do něj také symboly pro logické spojky $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ a závorky $(,)$.

Definice (Výroková formule). Výrok v jazyce $\mathbb{P}$ je prvek množiny $\text{VF}_{\mathbb{P}}$ definované následovně: $\text{VF}_{\mathbb{P}}$ je nejmenší množina splňující

- pro každý prvovýrok $p \in \mathbb{P}$ platí $p \in \text{VF}_{\mathbb{P}}$,
- pro každý výrok $\varphi \in \text{VF}_{\mathbb{P}}$ je $(\neg\varphi)$ také prvek $\text{VF}_{\mathbb{P}}$
- pro každé $\varphi, \psi \in \text{VF}_{\mathbb{P}}$ jsou $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi)$, a $(\varphi \leftrightarrow \psi)$ také prvky $\text{VF}_{\mathbb{P}}$.

Definice (Signatura). Signatura je dvojice $\langle \mathcal{R}, \mathcal{F} \rangle$, kde $\mathcal{R}, \mathcal{F}$ jsou disjunktní množiny symbolů (relační a funkční, ty zahrnují konstantní) spolu s danými aritami (tj. danými funkcí $\text{ar}: \mathcal{R} \cup \mathcal{F} \rightarrow \mathbb{N}$) a neobsahující symbol '=' (ten je rezervovaný pro rovnost).

Definice (Jazyk PL). Jazyk predikátové logiky je dán signaturou $\langle \mathcal{R}, \mathcal{F} \rangle$ a uvedením, zda jde o jazyk s rovností, či bez. Patří do něj spočetně mnoho proměnných, relační, funkční a konstantní symboly ze signatury (případně "="), univerzální a existenční kvantifikátory $\forall, \exists$, symboly pro logické spojky $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ a závorky $(,)$.

Definice (Termy). Termy jazyka L jsou konečné nápisy definované induktivně:

- každá proměnná a každý konstantní symbol z L je term,
- je-li f funkční symbol z L arity n a jsou-li $t_1, \dots, t_n$ termy, potom nápis $f(t_1, t_2, \dots, t_n)$ je také term.

Množinu všech termů jazyka L označíme Term_L .

Definice (Formule). Formule jazyka L jsou konečné nápisy definované induktivně. Platí stejné tři body jako u výroku, plus: Je-li φ formule a x proměnná, potom $(\forall x)\varphi$ a $(\exists x)\varphi$ jsou také formule.

Definice (Otevřená formule). Formule je otevřená, pokud neobsahuje žádný kvantifikátor.

Definice (Sentence). Formule je uzavřená (sentence), pokud nemá žádnou volnou proměnnou.

Definice (Literál). Literál ℓ je buď prvovýrok p nebo negace prvovýroku $\neg p$.

Definice (Klauzule). Klauzule je disjunkce literálů $C = \ell_1 \vee \ell_2 \vee \dots \vee \ell_n$. Jednotková klauzule je samotný literál ($n = 1$) a prázdnou klauzulí ($n = 0$) myslíme $\perp$.

Definice (CNF). Výrok je v konjunktivní normální formě (v CNF) pokud je konjunkcí klauzulí. Prázdný výrok v CNF je $\top$.

Definice (Elementární konjunkce). Elementární konjunkce je konjunkce literálů $E = \ell_1 \wedge \ell_2 \wedge \dots \wedge \ell_n$. Jednotková elementární konjunkce je samotný literál ($n = 1$). Prázdná ($n = 0$) je $\top$.

Definice (DNF). Výrok je v disjunktivní normální formě (v DNF) pokud je disjunkcí elementárních konjunkcí. Prázdný výrok v DNF je $\perp$.

Věta (Převod CNF DNF). Mějme konečný jazyk $\mathbb{P}$ a libovolnou množinu modelů $M \subseteq \text{M}_{\mathbb{P}}$. Potom existuje výrok φ_{DNF} v DNF a výrok φ_{CNF} v CNF takový, že $M = \text{M}_{\mathbb{P}}(\varphi_{\text{DNF}}) = \text{M}_{\mathbb{P}}(\varphi_{\text{CNF}})$. Konkrétně:

$$\begin{aligned}\varphi_{\text{DNF}} &= \bigvee_{v \in M} \bigwedge_{p \in \mathbb{P}} p^{v(p)} \\ \varphi_{\text{CNF}} &= \bigwedge_{v \in \overline{M}} \bigvee_{p \in \mathbb{P}} \overline{p^{v(p)}} = \bigwedge_{v \notin M} \bigvee_{p \in \mathbb{P}} p^{1-v(p)}\end{aligned}$$

- Implikace a ekvivalence:

$$\varphi \rightarrow \psi \sim \neg\varphi \vee \psi$$

$$\varphi \leftrightarrow \psi \sim (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)$$

- Negace:

$$\neg(\varphi \wedge \psi) \sim \neg\varphi \vee \neg\psi$$

$$\neg(\varphi \vee \psi) \sim \neg\varphi \wedge \neg\psi$$

$$\neg\neg\varphi \sim \varphi$$

- Konjunkce (převod do DNF):

$$\varphi \wedge (\psi \vee \chi) \sim (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$$

$$(\varphi \vee \psi) \wedge \chi \sim (\varphi \wedge \chi) \vee (\psi \wedge \chi)$$

- Disjunkce (převod do CNF):

$$\varphi \vee (\psi \wedge \chi) \sim (\varphi \vee \psi) \wedge (\varphi \vee \chi)$$

$$(\varphi \wedge \psi) \vee \chi \sim (\varphi \vee \chi) \wedge (\psi \vee \chi)$$

Definice (SAT). Výrok φ je v k -CNF, pokud je v CNF a každá klauzule má nejvýše k literálů. Problém k -SAT se ptá, zda je daný k -CNF výrok splnitelný. Pro $k \geq 3$ je k -SAT nadále NP-úplný, každou CNF formuli lze zakódovat do 3-CNF výroku.

Definice (PNF). Formule φ je v prenexní normální formě (PNF), je-li tvaru

$$(Q_1x_1) \dots (Q_nx_n)\varphi'$$

kde Q_i je kvantifikátor ($\forall$ nebo $\exists$) a formule φ' je otevřená. Formulí φ' potom říkáme otevřené jádro φ a $(Q_1x_1) \dots (Q_nx_n)$ je kvantifikátorový prefix.

Věta (Existence PNF). Ke každé formuli φ existuje ekvivalentní formule v prenexní normální formě.

Věta (Převod do PNF). Označme jako $\overline{Q}$ kvantifikátor opačný ke Q . Nechť φ a ψ jsou formule, a proměnná x nechť není volná ve formuli ψ . Potom platí:

$$\begin{aligned} \neg(Qx)\varphi &\sim (\overline{Q}x)\neg\varphi \\ (Qx)\varphi \wedge \psi &\sim (Qx)(\varphi \wedge \psi) \\ (Qx)\varphi \vee \psi &\sim (Qx)(\varphi \vee \psi) \\ (Qx)\varphi \rightarrow \psi &\sim (\overline{Q}x)(\varphi \rightarrow \psi) \\ \psi \rightarrow (Qx)\varphi &\sim (Qx)(\psi \rightarrow \varphi) \end{aligned}$$

Definice (Rezoluční pravidlo). Mějme klauzule C_1 a C_2 a literál ℓ takový, že $\ell \in C_1$ a $\bar{\ell} \in C_2$. Potom rezolventa klauzulí C_1 a C_2 přes literál ℓ je klauzule

$$C = (C_1 \setminus \{\ell\}) \cup (C_2 \setminus \{\bar{\ell}\}).$$

Definice (Rezoluční důkaz). Rezoluční důkaz (odvození) klauzule C z CNF formule S je konečná posloupnost klauzulí $C_0, C_1, \dots, C_n = C$ taková, že pro každé i buď $C_i \in S$ nebo C_i je rezolventou nějakých C_j, C_k kde $j < i$ a $k < i$.

Pokud rezoluční důkaz existuje, říkáme, že C je rezolucí dokazatelná z S , a píšeme $S \vdash_R C$. (Rezoluční) zamítnutí CNF formule S je rezoluční důkaz $\square$ z S , v tom případě je S (rezolucí) zamítnutelná.

Definice (Rezoluční strom). Rezoluční strom klauzule C z CNF formule S je konečný binární strom s vrcholy označenými klauzulemi, kde

- v kořeni je C ,
- v listech jsou klauzule z S ,
- v každém vnitřním vrcholu je rezolventa klauzulí ze synů tohoto vrcholu.

Definice (Rezoluční uzávěr). Rezoluční uzávěr $\mathcal{R}(S)$ formule S je definován induktivně jako nejmenší množina klauzulí splňující:

- $C \in \mathcal{R}(S)$ pro všechna $C \in S$,
- jsou-li $C_1, C_2 \in \mathcal{R}(S)$ a je-li C rezolventa C_1, C_2 , potom také $C \in \mathcal{R}(S)$.

1.6.2 Sémantika

Definice (Teorie). *Teorie v jazyce $\mathbb{P}$ je libovolná množina výroků v $\mathbb{P}$. Jednotlivým výročkům $\varphi \in T$ říkáme také axiomy.*

Definice (Pravdivostní funkce). *Pravdivostní funkce výroku φ v konečném jazyce $\mathbb{P}$ je funkce $f_{\varphi, \mathbb{P}}: \{0, 1\}^{|\mathbb{P}|} \rightarrow \{0, 1\}$ definovaná induktivně:*

- *je-li φ i -tý prvovýrok z $\mathbb{P}$, potom $f_{\varphi, \mathbb{P}}(x_0, \dots, x_{n-1}) = x_i$,*
- *je-li $\varphi = (\neg\varphi')$, potom $f_{\varphi, \mathbb{P}}(x_0, \dots, x_{n-1}) = f_{\neg}(f_{\varphi', \mathbb{P}}(x_0, \dots, x_{n-1}))$,*
- *je-li $\varphi = (\varphi' \square \varphi'')$ kde $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, potom*

$$f_{\varphi, \mathbb{P}}(x_0, \dots, x_{n-1}) = f_{\square}(f_{\varphi', \mathbb{P}}(x_0, \dots, x_{n-1}), f_{\varphi'', \mathbb{P}}(x_0, \dots, x_{n-1})).$$

Definice (Model jazyka VL). *Model jazyka $\mathbb{P}$ je libovolné pravdivostní ohodnocení $v: \mathbb{P} \rightarrow \{0, 1\}$. Množinu (všech) modelů jazyka $\mathbb{P}$ označíme $M_{\mathbb{P}}$:*

$$M_{\mathbb{P}} = \{v \mid v: \mathbb{P} \rightarrow \{0, 1\}\} = \{0, 1\}^{\mathbb{P}}$$

Definice (Model jazyka PL). *Model jazyka L , nebo také L -struktura, je libovolná struktura v signatuře jazyka L . Třidu všech modelů jazyka označíme M_L .*

Definice (Struktura). *Struktura v signatuře $\langle \mathcal{R}, \mathcal{F} \rangle$ je trojice $\mathcal{A} = \langle A, \mathcal{R}^{\mathcal{A}}, \mathcal{F}^{\mathcal{A}} \rangle$, kde*

- *A je neprázdná množina, říkáme jí doména (také univerzum),*
- *$\mathcal{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathcal{R}\}$ kde $R^{\mathcal{A}} \subseteq A^{\text{ar}(R)}$ je interpretace relačního symbolu R ,*
- *$\mathcal{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathcal{F}\}$ kde $f^{\mathcal{A}}: A^{\text{ar}(f)} \rightarrow A$ je interpretace funkčního symbolu f (speciálně pro konstantní symbol $c \in \mathcal{F}$ máme $c^{\mathcal{A}} \in A$).*

Definice (Platnost výroku v modelu, model výroku). *Mějme výrok φ v jazyce $\mathbb{P}$ a model $v \in M_{\mathbb{P}}$. Pokud platí $f_{\varphi, \mathbb{P}}(v) = 1$, potom říkáme, že výrok φ platí v modelu v , v je modelem φ , a píšeme $v \models \varphi$. Množinu všech modelů výroku φ označujeme $M_{\mathbb{P}}(\varphi)$.*

Definice (Platnost teorie, model teorie). *Je-li T teorie v jazyce $\mathbb{P}$, potom T platí v modelu v , pokud každý axiom $\varphi \in T$ platí ve v . V tom případě říkáme také, že v je modelem T , a píšeme $v \models T$. Množinu všech modelů teorie T v jazyce $\mathbb{P}$ označíme $M_{\mathbb{P}}(T)$.*

Definice (Sémantické pojmy). *Říkáme, že výrok φ (v jazyce $\mathbb{P}$) je*

- *pravdivý, tautologie, platí (v logice/logicky), a píšeme $\models \varphi$, pokud platí v každém modelu (jazyka $\mathbb{P}$), $M_{\mathbb{P}}(\varphi) = M_{\mathbb{P}}$,*
- *lživý, sporný, pokud nemá žádný model, $M_{\mathbb{P}}(\varphi) = \emptyset$.*
- *nezávislý, pokud platí v nějakém modelu, a neplatí v nějakém jiném modelu, tj. není pravdivý ani lživý, $\emptyset \subsetneq M_{\mathbb{P}}(\varphi) \subsetneq M_{\mathbb{P}}$,*
- *splnitelný, pokud má nějaký model, tj. není lživý, $M_{\mathbb{P}}(\varphi) \neq \emptyset$.*

Dále říkáme, že výroky φ, ψ (ve stejném jazyce $\mathbb{P}$) jsou (logicky) ekvivalentní, píšeme $\varphi \sim \psi$ pokud mají stejné modely, tj.

$$\varphi \sim \psi \text{ právě když } M_{\mathbb{P}}(\varphi) = M_{\mathbb{P}}(\psi).$$

Definice (Sémantické pojmy vzhledem k teorii). *Mějme teorii T v jazyce $\mathbb{P}$. Říkáme, že výrok φ v jazyce $\mathbb{P}$ je*

- pravdivý v T , důsledek T , platí v T , a píšeme $T \models \varphi$, pokud φ platí v každém modelu teorie T , neboli $M_{\mathbb{P}}(T) \subseteq M_{\mathbb{P}}(\varphi)$,
- lživý v T , sporný v T , pokud neplatí v žádném modelu T , neboli $M_{\mathbb{P}}(\varphi) \cap M_{\mathbb{P}}(T) = M_{\mathbb{P}}(T, \varphi) = \emptyset$.
- nezávislý v T , pokud platí v nějakém modelu T , a neplatí v nějakém jiném modelu T , tj. není pravdivý v T ani lživý v T , $\emptyset \subsetneq M_{\mathbb{P}}(T, \varphi) \subsetneq M_{\mathbb{P}}(T)$,
- splnitelný v T , konzistentní s T , pokud platí v nějakém modelu T , tj. není lživý v T , $M_{\mathbb{P}}(T, \varphi) \neq \emptyset$.

A říkáme, že výroky φ, ψ (ve stejném jazyce $\mathbb{P}$) jsou ekvivalentní v T , T -ekvivalentní, píšeme $\varphi \sim_T \psi$ pokud platí v týchž modelech T , tj.

$$\varphi \sim_T \psi \text{ právě když } M_{\mathbb{P}}(T, \varphi) = M_{\mathbb{P}}(T, \psi).$$

Definice. Řekneme, že teorie T v jazyce $\mathbb{P}$ je

- sporná, jestliže v ní platí $\perp$ (spor), ekvivalentně, jestliže nemá žádný model,
- bezesporná (splnitelná), pokud není sporná, tj. má nějaký model,
- kompletní, jestliže není sporná a každý výrok je v ní pravdivý nebo lživý, ekvivalentně, pokud má právě jeden model.

1.6.3 Extenze teorií

Definice (Extenze teorie). Mějme teorii T v jazyce $\mathbb{P}$.

- Extenze teorie T je libovolná teorie T' v jazyce $\mathbb{P}' \supseteq \mathbb{P}$ splňující $\text{Cs}_{\mathbb{P}}(T) \subseteq \text{Cs}_{\mathbb{P}'}(T')$,
- je to jednoduchá extenze, pokud $\mathbb{P}' = \mathbb{P}$,
- je to konzervativní extenze, pokud $\text{Cs}_{\mathbb{P}}(T) = \text{Cs}_{\mathbb{P}}(T') = \text{Cs}_{\mathbb{P}'}(T') \cap \text{VF}_{\mathbb{P}}$.

Skolemizace. Redukujeme otázku splnitelnosti dané teorie T na otázku splnitelnosti otevřené teorie T' . Celá konstrukce sestává z následujících kroků:

1. Převod do *prenexní normální formy* (vytýkání kvantifikátorů).
2. Nahrazení formulí jejich generálními uzávěry (abychom získali sentence – přidáme “ $\forall$ ”).
3. Odstranění existenčních kvantifikátorů (nahrazení sentencí *Skolemovými variantami* – odstraníme “ $\exists y_i$ ” a substituujeme “ $f_i(x_1, \dots, x_{n_i})$ ”).
4. Odstranění zbývajících univerzálních kvantifikátorů (výsledkem jsou otevřené formule).

1.6.4 Dokazatelnost

Pokud důkaz existuje, lze ho nalézt ‘algoritmicky’. Navíc musíme být schopni algoritmicky ověřit, že je daný objekt opravdu korektní důkaz. Existuje-li důkaz, říkáme, že φ je *dokazatelný* z T , a píšeme $T \vdash \varphi$. Po dokazovacím systému požadujeme dvě vlastnosti:

- *korektnost*: je-li výrok dokazatelný z teorie, je v ní pravdivý ($T \vdash \varphi \Rightarrow T \models \varphi$)
- *úplnost*: je-li výrok pravdivý v teorii, je z ní dokazatelný ($T \models \varphi \Rightarrow T \vdash \varphi$)

Definice (Tablo). Konečné tablo z teorie T je *uspořádaný, položkami označovaný strom zkonstruovaný aplikací konečně mnoha následujících pravidel*:

- *jednoprvkový strom označovaný libovolnou položkou je tablo z teorie T ,*
- *pro libovolnou položkou P na libovolné větvi V , můžeme na konec větve V připojit atomické tablo pro položku P ,*
- *na konec libovolné větve můžeme připojit položku $T\alpha$ pro libovolný axiom teorie $\alpha \in T$.*

Tablo z teorie T je buď konečné, nebo i nekonečné: v tom případě vzniklo ve spočetně mnoha krocích. Můžeme ho formálně vyjádřit jako sjednocení $\tau = \bigcup_{i \geq 0} \tau_i$, kde τ_i jsou konečná tabla z T , τ_0 je jednoprvkové tablo, a τ_{i+1} vzniklo z τ_i v jednom kroku.

Tablo pro položku P je tablo, které má položku P v kořeni.

Definice (Tablo důkaz). Tablo důkaz výroku φ z teorie T je sporné tablo z teorie T s položkou $F\varphi$ v kořeni. Pokud existuje, je φ (tablo) dokazatelný z T , píšeme $T \vdash \varphi$. (Definujme také tablo zamítnutí jako sporné tablo s $T\varphi$ v kořeni. Pokud existuje, je φ (tablo) zamítnutelný z T , tj. platí $T \vdash \neg\varphi$.)

- *Tablo je sporné, pokud je každá jeho větev sporná.*
- *Větev je sporná, pokud obsahuje položky $T\psi$ a $F\psi$ pro nějaký výrok ψ , jinak je bezesporná.*
- *Tablo je dokončené, pokud je každá jeho větev dokončená.*
- *Větev je dokončená, pokud*
 - *je sporná, nebo*
 - *je každá její položka na této větvi redukována a zároveň obsahuje položku $T\alpha$ pro každý axiom $\alpha \in T$.*
- *Položka P je redukována na větvi V procházející touto položkou, pokud*
 - *je tvaru Tp resp. Fp pro nějakou výrokovou proměnnou $p \in \mathbb{P}$, nebo*
 - *vyskytuje se na V jako kořen atomického tabla (tj., typicky, při konstrukci tabla již došlo k jejímu rozvoji na V).*

Věta (Konečnost sporu). Je-li $\tau = \bigcup_{i \geq 0} \tau_i$ sporné tablo, potom existuje $n \in \mathbb{N}$ takové, že τ_n je sporné konečné tablo.

Důsledek (Konečnost důkazu). Pokud $T \vdash \varphi$, potom existuje i konečný tablo důkaz φ z T .

	$\neg$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
True	$T\neg\varphi$ $F\varphi$	$T\varphi \wedge \psi$ $T\varphi$ $T\psi$	$T\varphi \vee \psi$ / \ $T\varphi$ $T\psi$	$T\varphi \rightarrow \psi$ / \ $F\varphi$ $T\psi$	$T\varphi \leftrightarrow \psi$ / \ $T\varphi$ $F\varphi$ $T\psi$ $F\psi$
	$F\neg\varphi$ $T\varphi$	$F\varphi \wedge \psi$ / \ $F\varphi$ $F\psi$	$F\varphi \vee \psi$ $F\varphi$ $F\psi$	$F\varphi \rightarrow \psi$ $T\varphi$ $F\psi$	$F\varphi \leftrightarrow \psi$ / \ $T\varphi$ $F\varphi$ $F\psi$ $T\psi$

Tabulka 1.1 Atomická tabla

1.6.5 Věty o kompaktnosti a úplnosti výrokové a predikátové logiky

Věta (O kompaktnosti). *Teorie má model, právě když každá její konečná část má model.*

Důsledek. *Spočetně nekonečný graf je bipartitní, právě když je každý jeho konečný podgraf bipartitní.*

Věta (O úplnosti VL). *Je-li výrok φ pravdivý v teorii T , potom je tablo dokazatelný z T , tj.*

$$T \models \varphi \implies T \vdash \varphi.$$

Věta (O úplnosti PL). *Je-li sentence φ pravdivá v teorii T , potom je tablo dokazatelná z T , tj.*

$$T \models \varphi \implies T \vdash \varphi.$$

1.6.6 Rozhodnutelnost

Definice (Kompletní teorie ve VL). *Teorie je kompletní, je-li bezesporná a každý výrok je buď pravdivý, nebo lživý. Ekvivalentně, právě když má jeden model.*

Definice (Kompletní teorie v PL). *Teorie je kompletní, je-li bezesporná a každá sentence je v ní buď pravdivá, nebo lživá. Ekvivalentně, právě když má právě jeden model až na elementární ekvivalenci.*

Definice (Elementární ekvivalence). *Struktury $\mathcal{A}, \mathcal{B}$ (v témž jazyce) jsou elementárně ekvivalentní, pokud v nich platí tytéž sentence. Značíme $\mathcal{A} \equiv \mathcal{B}$.*

Definice (Teorie struktury). *Mějme L -strukturu $\mathcal{A}$. Teorie struktury $\mathcal{A}$, značíme $\text{Th}(\mathcal{A})$ je množina všech L -sentencí platných v $\mathcal{A}$:*

$$\text{Th}(\mathcal{A}) = \{\varphi \mid \varphi \text{ je } L\text{-sentence a } \mathcal{A} \models \varphi\}$$

Definice (Rekurzivní axiomatizace). *Teorie T je rekurzivně axiomatizovaná, pokud existuje algoritmus, který pro každou vstupní formuli φ doběhne a odpoví, zda $\varphi \in T$.*

Definice (Rozhodnutelnost). *O teorii T říkáme, že je*

- rozhodnutelná, *pokud existuje algoritmus, který pro každou vstupní formuli φ doběhne a odpoví, zda $T \models \varphi$,*
- částečně rozhodnutelná, *pokud existuje algoritmus, který pro každou vstupní formuli φ :*
 - *pokud $T \models \varphi$, doběhne a odpoví “ano”,*
 - *pokud $T \not\models \varphi$, buď nedoběhne, nebo doběhne a odpoví “ne”.*

Věta. *Nechť T je rekurzivně axiomatizovaná. Potom:*

- (i) *T je částečně rozhodnutelná,*
- (ii) *je-li T navíc kompletní, potom je rozhodnutelná.*

Definice (Rekurzivní axiomatizovatelnost). *Třída L -struktur $K \subseteq M_L$ je rekurzivně axiomatizovatelná, pokud existuje rekurzivně axiomatizovaná L -teorie T taková, že $K = M_L(T)$. Teorie T' je rekurzivně axiomatizovatelná, pokud je rekurzivně axiomatizovatelná třída jejích modelů, neboli pokud je T' ekvivalentní nějaké rekurzivně axiomatizované teorii.*

Příklady rozhodnutelných teorií. Pro následující struktury je $\text{Th}(\mathcal{A})$ rekurzivně axiomatizovatelná, a tedy i rozhodnutelná:

- $\langle \mathbb{Z}, \leq \rangle$, jde o tzv. teorii *diskrétních lineárních uspořádání*,
- $\langle \mathbb{N}, S, 0 \rangle$, teorie *následníka s nulou*,
- $\langle \mathbb{R}, +, -, \cdot, 0, 1 \rangle$, teorie *reálně uzavřených těles*,
- $\langle \mathbb{C}, +, -, \cdot, 0, 1 \rangle$, teorie *algebraicky uzavřených těles charakteristiky 0*.

Společná informatika

Z následujících 4 témat budou vybrána 2.

2.1 Automaty a jazyky

2.1.1 Regulární Jazyky

Definice (Regulární gramatika (Typ 3)). *Gramatika G je regulární, pokud obsahuje pouze pravidla:*

$$A \rightarrow wB \quad \text{nebo} \quad A \rightarrow w, \quad A, B \in V, w \in T^*.$$

Definice (DFA). Deterministický konečný automat (DFA) $A = (Q, \Sigma, \delta, q_0, F)$ sestává z:

- konečné množiny stavů Q ,
- konečné neprázdné množiny vstupních symbolů (abecedy) Σ ,
- přechodové funkce $\delta : Q \times \Sigma \rightarrow Q$,
- počátečního stavu $q_0 \in Q$,
- množiny koncových (přijímajících) stavů $F \subseteq Q$.

Definice (ϵ -NFA). Nedeterministický konečný automat s ϵ přechody (ϵ -NFA) $A = (Q, \Sigma, \delta, q_0, F)$:

- konečné množiny stavů Q ,
- konečné množiny vstupních symbolů Σ ,
- přechodové funkce, zobrazení $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$ vracející podmnožinu Q ,
- počátečního stavu $q_0 \in Q$,
- množiny koncových (přijímajících) stavů $F \subseteq Q$.

Definice (Rozšířená přechodová funkce). Mějme přechodovou funkci $\delta : Q \times \Sigma \rightarrow Q$. Rozšířenou přechodovou funkci $\delta^* : Q \times \Sigma^* \rightarrow Q$ definujeme induktivně:

- $\delta^*(q, \epsilon) = q$,
- $\delta^*(q, wx) = \delta(\delta^*(q, w), x)$, pro $x \in \Sigma, w \in \Sigma^*$.

Definice (Regulární výrazy (RegE), hodnota RegE $L(\alpha)$). **Regulární výrazy** $\alpha, \beta \in \text{RegE}(\Sigma)$ nad konečnou neprázdnou abecedou $\Sigma = \{a_1, a_2, \dots, a_n\}$ a jejich hodnota $L(\alpha)$ jsou definovány induktivně:

	výraz α	pro	hodnota $L(\alpha) \equiv [\alpha]$
• Základ:	ϵ	prázdný řetězec	$L(\epsilon) = \{\epsilon\}$
	$\emptyset$	prázdný výraz	$L(\emptyset) = \{\} \equiv \emptyset$
	a	$a \in \Sigma$	$L(a) = \{a\}$

	výraz	hodnota
• Indukce:	$\alpha + \beta$	$L(\alpha + \beta) = L(\alpha) \cup L(\beta)$
	$\alpha\beta$	$L(\alpha\beta) = L(\alpha)L(\beta)$
	α^*	$L(\alpha^*) = L(\alpha)^*$
	(α)	$L((\alpha)) = L(\alpha)$

Každý regulární výraz dostaneme indukcí výše, tj. třída $RegE(\Sigma)$ je nejmenší třída uzavřená na uvedené operace.

2.1.2 Bezkontextové jazyky a Zásobníkový automat

Definice (Formální (generativní) gramatika). *Formální gramatika je $G = (V, T, P, S)$ složena z:*

- konečné množiny neterminálů V ,
- neprázdné konečné množiny terminálů T ,
- počátečního symbolu $S \in V$,
- konečné množiny pravidel P reprezentující rekurzivní definici jazyka. Každé pravidlo má tvar:

$$\beta A \gamma \rightarrow \omega, \quad A \in V, \quad \beta, \gamma, \omega \in (V \cup T)^*,$$

tj. levá strana obsahuje alespoň jeden neterminální symbol.

Definice (Bezkontextová gramatika (CFG)). *CFG je $G = (V, T, P, S)$ gramatika, obsahující pouze pravidla tvaru:*

$$A \rightarrow \omega, \quad A \in V, \quad \omega \in (V \cup T)^*.$$

Definice (Derivace $\Rightarrow^*$). *Mějme gramatiku $G = (V, T, P, S)$.*

- Říkáme, že α se přímo přepíše na ω (píšeme $\alpha \Rightarrow_G \omega$ nebo $\alpha \Rightarrow \omega$), jestliže:

$$\exists \beta, \gamma, \eta, \nu \in (V \cup T)^* : \alpha = \eta\beta\nu, \omega = \eta\gamma\nu \text{ a } (\beta \rightarrow \gamma) \in P.$$

- Říkáme, že α se přepíše na ω (píšeme $\alpha \Rightarrow^* \omega$), jestliže:

$$\exists \beta_1, \dots, \beta_n \in (V \cup T)^* : \alpha = \beta_1 \Rightarrow \beta_2 \Rightarrow \dots \Rightarrow \beta_n = \omega,$$

- Posloupnost $\beta_1, \dots, \beta_n$ nazýváme derivací (odvozením).

Definice (Jazyk generovaný gramatikou G). *Jazyk $L(G)$ generovaný gramatikou $G = (V, T, P, S)$ je množina terminálních řetězců, pro které existuje derivace ze startovního symbolu:*

$$L(G) = \{w \in T^* \mid S \Rightarrow_G^* w\}.$$

Jazyk neterminálu $A \in V$ definujeme:

$$L(A) = \{w \in T^* \mid A \Rightarrow_G^* w\}.$$

Definice (Zásobníkový automat (PDA)). *PDA je $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde:*

- Q je konečná množina stavů,
- Σ je neprázdná konečná množina vstupních symbolů,
- Γ je neprázdná konečná zásobníková abeceda,

- δ je přechodová funkce:

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P_{FIN}(Q \times \Gamma^*),$$

platí $\delta(p, a, X) \ni (q, \gamma)$, kde ' p ' je aktuální stav, ' a ' je aktuálně čtený symbol, ' X ' je symbol na vrcholu zásobníku, ' q ' je nový stav, do kterého automat přejde a ' γ ' je řetězec zásobníkových symbolů, který nahradí X na vrcholu zásobníku,

- $q_0 \in Q$ je počáteční stav,
- $Z_0 \in \Gamma$ je počáteční zásobníkový symbol (na začátku je jediný symbol na zásobníku),
- F je množina přijímajících (koncových) stavů; může být nedefinovaná.

2.1.3 Rekurzivně spočetné jazyky

Definice (Turingův stroj (TM)). Turingův stroj (TM) je sedmice $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, kde:

- Q je konečná množina stavů,
- Σ je konečná neprázdná množina vstupních symbolů,
- Γ je konečná množina všech symbolů pro pásku ($\Gamma \supseteq \Sigma$, $Q \cap \Gamma = \emptyset$),
- δ je (částečná) přechodová funkce:

$$\delta : (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\},$$

kde $\delta(q, X) = (p, Y, D)$ znamená:

- $q \in (Q - F)$ je aktuální stav,
- $X \in \Gamma$ je aktuální symbol na pásku,
- p je nový stav,
- $Y \in \Gamma$ je symbol pro zapsání do aktuální buňky,
- $D \in \{L, R\}$ je směr pohybu hlavy (doleva, doprava).

- $q_0 \in Q$ je počáteční stav,
- $B \in \Gamma - \Sigma$ je symbol pro prázdné buňky (blank),
- $F \subseteq Q$ je množina koncových (přijímajících) stavů.

Definice (Problémem). Problémem P myslíme množinu otázek kódovatelnou řetězcí nad abecedou Σ s odpověďmi $\in \{\mathbf{ANO}, \mathbf{NE}\}$. Pokud problém definujeme jakožto množinu, jde o otázku, zda vstup kóduje prvek dané množiny (např. polynom s celočíselným kořenem).

Definice (Rozhodnutelný problém). Problém je (algoritmicky) rozhodnutelný, pokud existuje Turingův stroj takový, že pro každý vstup $w \in P$ zastaví a navíc přijme právě když

$$P(w) = \mathbf{ANO} \quad (\text{tj. pro } P(w) = \mathbf{NE}, \text{ zastaví v ne-přijímacím stavu}).$$

Problém, který není algoritmicky rozhodnutelný je nerozhodnutelný.

Definice (Problém zastavení). Instancí problému zastavení je dvojice řetězců $M, w \in \{0, 1\}^*$. Problém zastavení je najít algoritmus $\mathbf{Halt}(M, w)$, který vydá 1 právě když stroj M zastaví na vstupu w , jinak vydá 0.

Věta (Problém zastavení). Problém zastavení není rozhodnutelný.

2.1.4 Chomského Hierarchie

Definice (Chomského hierarchie). *Gramatiky rozdělujeme do čtyř typů:*

- **Gramatiky typu 0 (rekurzivně spočetné jazyky $\mathcal{L}_0$)**

Pravidla v obecné formě:

$$\alpha \rightarrow \omega, \quad \alpha, \omega \in (V \cup T)^*,$$

a α obsahuje alespoň jeden neterminální symbol.

- **Gramatiky typu 1 (kontextové gramatiky, jazyky $\mathcal{L}_1$)**

Pouze pravidla ve tvaru:

$$\gamma A \beta \rightarrow \gamma \omega \beta, \quad A \in V, \quad \gamma, \beta \in (V \cup T)^*, \quad \omega \in (V \cup T)^+.$$

Výjimkou je pravidlo $S \rightarrow \epsilon$, pak se S nesmí vyskytovat na pravé straně žádného pravidla.

- **Gramatiky typu 2 (bezkontextové gramatiky, jazyky $\mathcal{L}_2$)**

Pouze pravidla ve tvaru:

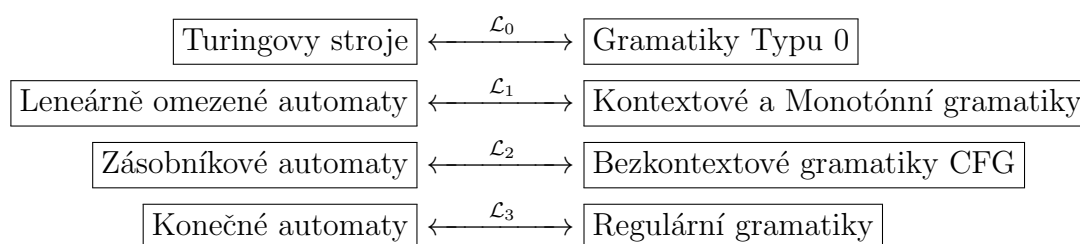
$$A \rightarrow \omega, \quad A \in V, \quad \omega \in (V \cup T)^*.$$

- **Gramatiky typu 3 (regulární gramatiky, jazyky $\mathcal{L}_3$)**

Pouze pravidla ve tvaru:

$$A \rightarrow \omega B \quad \text{nebo} \quad A \rightarrow \omega,$$

kde $A, B \in V$ a $\omega \in T^$.*



Převody

ϵ -NFA na DFA (konstrukce podmnožin)

1. Počáteční stav je zachován.
2. Přijímající stavy jsou všechny množiny obsahující nějaký přijímající stav.
3. Každý nový stav odpovídá množině stavů NFA. (Pro každý symbol z Σ spočítáme množinu stavů, do kterých se NFA může dostat, a ta tvoří nový stav v DFA.)

Regulární gramatika na DFA

1. Každý neterminál odpovídá stavu automatu + vytvoříme koncový stav K .
2. Počáteční stav S gramatiky je počáteční stav automatu.
3. Tvar $A \rightarrow aB$ zapíšeme v automatu jako $A \xrightarrow{a} B$.
4. Tvar $A \rightarrow a$ zapíšeme v automatu jako $A \xrightarrow{a} K$.
5. Tvar $A \rightarrow \epsilon$ zapíšeme v automatu jako koncový stav.

DFA na Regulární gramatika

1. Každému stavu automatu $A, B, C, \dots$ přiřadíme neterminál $A, B, C, \dots$
2. Počátečnímu stavu přiřadíme počáteční neterminál.
3. Přejchod $A \xrightarrow{a} B$ zapíšeme v gramatice jako pravidlo $A \rightarrow aB$.
4. Pokud je stav koncový, přidáme pravidlo $A \rightarrow \varepsilon$.

DFA na RegEx

1. Převodeme automat na regulární gramatiku.
2. Součet koncových stavů DFA bude tvořit výslednou rovnici.
3. Následně dosazujeme do stavů a převádíme je na RegEx.
4. Převod na RegEx: výraz ve tvaru $A = A\alpha + \beta$ přepíšeme na $A = \beta\alpha^*$.

RegEx na NFA

1. Pro každý RegEx symbol vytvoříme NFA s jedním přechodem. (symbol 0 bude $\rightarrow \bigcirc \xrightarrow{0} \odot$.)
2. Pokud je RegEx ve tvaru 01, pak automaty skládáme.
3. Pokud je RegEx ve tvaru 0 + 1, pak přidáme ε počáteční stav, který jde do automatů 0 a 1 a následně jsou oba svedeny do jednoho koncového stavu. (*Rozdvojení a sloučení*).
4. Pokud je RegEx ve tvaru 1^* , pak vytváříme cyklus.

CFG na Chomského normální tvar

1. Odstraníme ε -pravidla, jednotková pravidla ($A \rightarrow B$) a zbytečné symboly (nedosažitelné).
2. Pro každý terminál a vytvoříme nový neterminál A a přidáme pravidlo $A \rightarrow a$.
3. Převodeme pravidla do tvaru $A \rightarrow BC$ nebo $A \rightarrow a$.

CFG na PDA

Mějme CFG gramatiku $G = (V, T, P, S)$, a konstruuujeme PDA $P = (\{q\}, T, V \cup T, \delta, q, S)$.

1. Počáteční symbol gramatiky vložíme na zásobník.
2. Pro neterminály $A \in V : \delta(q, \varepsilon, A) = \{(q, \beta) \mid A \rightarrow \beta \text{ je pravidlo } G\}$.
3. Pro každý terminál $a \in T : \delta(q, a, a) = \{(q, \varepsilon)\}$. (*pop*)
4. Přijímání probíhá prázdným zásobníkem.

2.2 Algoritmy a datové struktury

2.2.1 Časová složitost algoritmu.

Definice (Časová složitost). *Nechť $t(x)$ je doba běhu pro vstup x . Potom časová složitost výpočtu:*

$$T(n) := \max\{t(x) \mid x \text{ vstup velikosti } n\}.$$

Definice (Prostorová složitost). *Nechť $s(x) := \text{“max adresa} - \text{min adresa} + 1\text{”}$ je prostor běhu pro vstup x . Potom prostorová složitost výpočtu je*

$$S(n) := \max\{s(x) \mid x \text{ vstup velikosti } n\}.$$

Asymptotická notace.

Mějme funkce $f, g : \mathbb{N} \rightarrow \mathbb{R}$.

Definice (Horní odhad). $f(n) \in \mathcal{O}(g(n)) \iff (\exists c > 0)(\exists n_0 \forall n > n_0) : 0 \leq f(n) \leq c \cdot g(n)$.

Definice (Dolní odhad). $f(n) \in \Omega(g(n)) \iff (\exists c > 0)(\exists n_0 \forall n > n_0) : 0 \leq c \cdot g(n) \leq f(n)$.

Definice (Těsný odhad). $f(n) \in \Theta(g(n)) \iff f(n) \in \mathcal{O}(g(n)) \quad \wedge \quad f(n) \in \Omega(g(n))$.

2.2.2 Třídy složitosti

Definice (Třída P). *Problém $L \in P \iff$ existuje algoritmus A a polynom f , že pro každý vstup x , algoritmus A doběhne v čase nejvýše $f(|x|)$ a vrátí výsledek $A(x) = L(x)$.*

Definice (Třída NP). *Problém $L \in NP \iff \exists K \in P$ problém a polynom g , že $\forall x : L(x) = 1$ právě tehdy, když pro nějaký řetězec y délky nejvýše $g(|x|)$ platí $K(x, y) = 1$.*

(“ K řeší L , ale krom vstupu x má nápovědu y . Pokud $L(x) = 1$, pak musí existovat nápověda y , kterou K schválí.”)

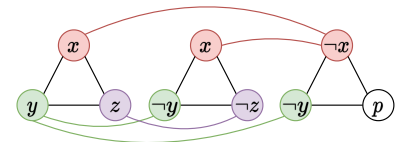
Definice (NP -těžkost). *Problém je NP -těžký, pokud na něj lze převést libovolný problém v NP .*

Definice (NP -úplnost). *Problém je NP -úplný, pokud je NP -těžký a je v NP .*

Definice (Převod). *Platí, že $A \rightarrow B \iff \exists f : \{0, 1\}^* \rightarrow \{0, 1\}^*$, že $\forall x : A(x) = B(f(x))$ a f běží v polytime.*

Příklady převodů:

- **SAT $\rightarrow$ 3-SAT:** přidáme literály: $(\alpha \vee \beta) \rightarrow (\alpha \vee x) \wedge (\neg x \vee \beta)$.
- **3-SAT $\rightarrow$ Nezávislá množina:** klauzule tvoří trojúhelníky. Spojujeme x s $\neg x$. Viz obrázek.
- **3-SAT $\rightarrow$ 3,3-SAT:** proměnné rozdělíme a zařídíme stejné ohodnocení kolečkem implikací.
- **3,3-SAT $\rightarrow$ 3D párování**



2.2.3 Metoda rozděl a panuj

Rekurzivně dělíme na podproblémy – problém velikosti n na dva $\frac{n}{2}$ apod.

Věta (Master). Rekurentní rovnice $T(n) = a \cdot T(\frac{n}{b}) + \Theta(n^c)$, $T(1) = 1$ má pro konstanty $a \geq 1, b > 1, c \geq 0$ řešení:

(1) pokud $\frac{a}{b^c} = 1$, pak $T(n) = \Theta(n^c \log n)$,

(2) pokud $\frac{a}{b^c} < 1$, pak $T(n) = \Theta(n^c)$,

(3) pokud $\frac{a}{b^c} > 1$, pak $T(n) = \Theta(n^{\log_b a})$.

Neboli, problém rozložíme na a podproblémů velikosti $\frac{n}{b}$. Výsledky složíme v $\Theta(n^c)$.

Merge sort.

Na vstupu n prvků, rozdělíme na dvě části $\lfloor \frac{n}{2} \rfloor$, $\lceil \frac{n}{2} \rceil$, na konci slijeme. Použijeme Master theorem: Pro $T(1) = 1$, triviálně. Pro $T(n)$ problém dělíme na $a = 2$ podproblémy velikosti $\frac{n}{2}$. Slijeme nakonec dohromady v $\Theta(n)$, $c = 1$. Celkem tedy $T(n) = 2T(\frac{n}{2}) + n$, což vyhovuje případu (1) a máme:

$$T(n) = \Theta(n \log n).$$

Algorithm 5: MergeSort($a_1, \dots, a_n$)

```

1 if  $n = 1$  then return  $a_1 = b_1$ 
2  $x_1, \dots, x_{\lfloor n/2 \rfloor} \leftarrow \text{MergeSort}(a_1, \dots, a_{\lfloor n/2 \rfloor})$ 
3  $y_1, \dots, y_{\lceil n/2 \rceil} \leftarrow \text{MergeSort}(a_{\lfloor n/2 \rfloor + 1}, \dots, a_n)$ 
4  $b_1, \dots, b_n \leftarrow \text{Merge}(x_1, \dots, x_{\lfloor n/2 \rfloor}, y_1, \dots, y_{\lceil n/2 \rceil})$ 

```

Algorithm 6: Merge($x_1, \dots, x_m, y_1, \dots, y_n$)

```

1  $i \leftarrow 1, j \leftarrow 1, k \leftarrow 1$   $\triangleright$  zbývá slít  $x_1, \dots, x_m, y_1, \dots, y_n$ , výsledek  $z_k, \dots, z_{m+n}$ 
2 while  $i \leq m \wedge j \leq n$  do
3   if  $x_i \leq y_j$  then
4     | přesuneme prvek z  $x$ :  $z_k \leftarrow x_i, i \leftarrow i + 1$ 
5   end
6   else
7     | přesuneme prvek z  $y$ :  $z_k \leftarrow y_j, j \leftarrow j + 1$ 
8   end
9    $k \leftarrow k + 1$ 
10 end
11 if  $i \leq m$ , zkopírujeme zbylá  $x$ :  $z_k, \dots, z_{m+n} \leftarrow x_i, \dots, x_m$ 
12 if  $j \leq n$ , zkopírujeme zbylá  $y$ :  $z_k, \dots, z_{m+n} \leftarrow y_j, \dots, y_n$ 
13 return  $z_1, \dots, z_{m+n}$ 

```

Karacubovo násobení.

Máme dvě čísla X, Y a chceme jejich součin $X \cdot Y$. Rozděláme horních $\frac{n}{2}$ a dolních $\frac{n}{2}$ cifer.

Platí $\begin{cases} X = A \cdot 10^{n/2} + B \\ Y = C \cdot 10^{n/2} + D \end{cases}$, pro čísla A, B, C, D . Součin pak můžeme spočítat jako:

$$XY = AC \cdot 10^n + (AD + BC) \cdot 10^{n/2} + BD.$$

Což je ovšem stále asymptoticky n^2 . Můžeme to ale ještě vylepšit – o jeden součin méně – na:

$$XY = AC \cdot 10^n + ((A + B)(C + D) - AC - BD) \cdot 10^{n/2} + BD.$$

Máme tak $T(n) = 3T(\frac{n}{2}) + \Theta(n)$, což vyhovuje případu (3) a dostáváme tak $\Theta(n^{\log_2 3}) \approx n^{1.53}$.

2.2.4 Binární vyhledávací stromy

Definice (Vyhledávací strom). *Pro vyhledávací strom platí, že klíč v každém vrcholu je větší než všechny klíče v levém podstromu a je menší než všechny klíče v pravém podstromu.*

Definice (AVL). *Rozdíl v hloubkách stromů je maximálně jedna. (vyvážený strom)
Hloubka je $\Theta(\log n)$.*

Definice (Binární strom). *Strom mající nejvýše dva syny.*

Základní operace.

Notace: $v \equiv$ kořen/vrchol, $l(v) \equiv$ levý syn v , $r(v) \equiv$ pravý syn v , $k(v) \equiv$ klíč v .

Show – vypíše všechny prvky	Find – porovná $k(v)$ s předem hledaným klíčem x
1. if $v = \emptyset$ then return 2. Show($l(v)$) 3. Vypiš $k(v)$ 4. Show($r(v)$)	1. if $v = \emptyset$ then return $\emptyset$ 2. if $x = k(v)$ then return v 3. if $x < k(v)$ then return Find($l(v), x$) 4. if $x > k(v)$ then return Find($r(v), x$)
Min – minimum z prvků	Insert – vkládáme prvek do stromu
1. if $l(v) = \emptyset$ then return v 2. return Min($l(v)$)	1. if $v = \emptyset$ then vytvoříme nový v s klíčem x 2. if $x = k(v)$ then pass $\triangleright x$ je již ve stromu 3. if $x < k(v)$ then $l(v) \leftarrow$ Insert($l(v), x$) 4. if $x > k(v)$ then $r(v) \leftarrow$ Insert($r(v), x$)
Delete – odstraníme prvek ze stromu	
1. if $l(v) = r(v) = \emptyset$ then return $\emptyset$ $\triangleright$ list 2. if $l(v) = \emptyset$ then return $r(v)$ $\triangleright$ existuje pravý syn 3. if $r(v) = \emptyset$ then return $l(v)$ $\triangleright$ existuje levý syn 4. $s \leftarrow$ Min($r(v)$) $\triangleright$ nahradíme následníka 5. $k(v) \leftarrow s$ 6. $r(v) \leftarrow$ Delete($r(v), s$)	

2.2.5 Třídění

BubbleSort.

Primitivní třídící algoritmus. Časová složitost je $\mathcal{O}(n^2)$ v nejhorším i průměrném případě.

Algorithm 7: BubbleSort ($A = [a_1, \dots, a_n]$)

```

1 for  $i \leftarrow 1$  to  $n - 1$  do
2   for  $j \leftarrow 1$  to  $n - i$  do
3     if  $A[j] > A[j + 1]$  then
4        $\text{swap}(A[j], A[j + 1])$ 
5     end
6   end
7 end
8 return  $A$ 

```

InsertionSort.

Je efektivní pro malá pole a téměř seřazená data. Časová složitost je ale stále $\mathcal{O}(n^2)$.

Algorithm 8: InsertionSort ($A = [a_1, \dots, a_n]$)

```
1 for  $i \leftarrow 2$  to  $n$  do
2    $key \leftarrow A[i]$ 
3    $j \leftarrow i - 1$ 
4   while  $j > 0$  and  $A[j] > key$  do
5      $A[j + 1] \leftarrow A[j]$ 
6      $j \leftarrow j - 1$ 
7   end
8    $A[j + 1] \leftarrow key$ 
9 end
10 return  $A$ 
```

Poznámka. Insertionsort je vhodný pro malá pole (typicky $n < 20$) a dosahuje lineární časové složitosti pro téměř seřazená data.

QuickSort.

Typický příklad algoritmu rozděl a panuj. Průměrná časová složitost: $\mathcal{O}(n \log n)$, v případě špatného výběru pivotu ale můžeme dosáhnout $\mathcal{O}(n^2)$. Paměťová složitost je $\mathcal{O}(\log n)$ (rekurzivní zásobník).

Algorithm 9: QuickSort ($X = [x_1, \dots, x_n]$)

```
1 if  $n \leq 1$  then return  $Y = X$ 
2  $p \leftarrow$  vyber pivotu z  $X$ 
3  $L \leftarrow$  prvky  $X$ , které  $< p$ 
4  $P \leftarrow$  prvky  $X$ , které  $> p$ 
5  $S \leftarrow$  prvky  $X$ , které  $= p$ 
6 Rekurzivně třídíme části
7    $L \leftarrow \text{Quicksort}(L)$ 
8    $P \leftarrow \text{Quicksort}(P)$ 
9 return Slepíme  $Y = (L, S, P)$ 
```

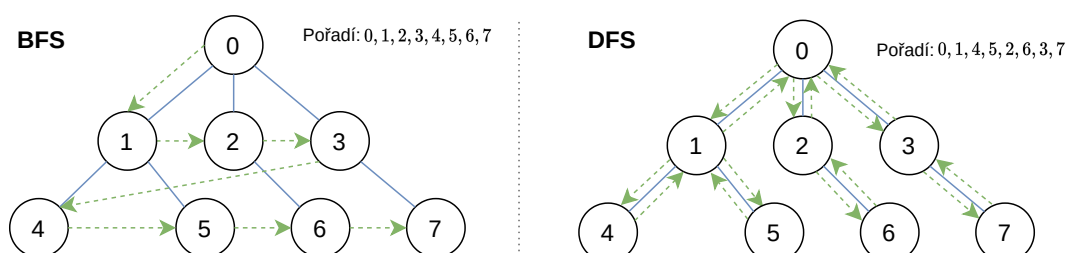
Věta. Quicksort je randomizovaný algoritmus, jehož očekávaná časová složitost je $\mathcal{O}(n \log n)$ při volbě pivotu náhodným výběrem.

Věta (O složitosti třídění). Každý deterministický algoritmus v porovnávacím modelu, který třídí n -prvkovou posloupnost, použije v nejhorším případě $\Omega(n \log n)$ porovnání.

2.2.6 Grafové algoritmy

Prohledávání do hloubky (BFS) a do šířky (DFS).

Na vstupu máme graf $G = (V, E)$ s počátečním vrcholem $v_0 \in V$. Během průchodu grafem rozlišujeme tři možné stavy vrcholů; *nenalezené* – ještě jsme je nepotkali, *otevřené* – potkali, ale neprozkoumali hrany, které z nich vedou, a *uzavřené* – už jsme prozkoumali i hrany, nadále se nezabýváme.



Algorithm 10: BFS(G, v_0)

```
1 for každý  $v \in V$  do
2   | stav( $v$ )  $\leftarrow$  nenalezený
3 end
4 stav( $v_0$ )  $\leftarrow$  otevřený
5  $Q \leftarrow v_0$  ▷ fronta
6 while  $Q \neq \emptyset$  do
7   |  $v \leftarrow Q.\text{pop}()$ 
8   | for každý  $w$  následník vrcholu  $v$  do
9     | if stav( $w$ ) = nenalezený then
10    |   | stav( $w$ )  $\leftarrow$  otevřený
11    |   |  $Q.\text{push}(w)$ 
12    | end
13  | end
14  | stav( $v$ )  $\leftarrow$  uzavřený
15 end
```

Pro DFS ještě definujeme pomocná pole $\text{in}(v)$, $\text{out}(v)$, do kterých zaznamenejeme, v jakém pořadí jsme vrcholy otevírali a zavírali.

Algorithm 11: DFS(G)

```
1 for každý  $v \in V$  do
2   | stav( $v$ )  $\leftarrow$  nenalezený
3   |  $\text{in}(v), \text{out}(v) \leftarrow$  nedefinováno
4 end
5  $T \leftarrow 0$  ▷ globální počítadlo kroků
6 DFS2( $v_0$ )
```

Algorithm 12: DFS2(v)

```
1 stav( $v$ )  $\leftarrow$  otevřený
2  $T \leftarrow T + 1$ ,  $\text{in}(v) \leftarrow T$ 
3 for každý následník  $w$  vrcholu  $v$  do
4   | if stav( $w$ ) = nenalezený then
5     |   | DFS2( $w$ )
6   | end
7 end
8 stav( $v$ )  $\leftarrow$  uzavřený
9  $T \leftarrow T + 1$ ,  $\text{out}(v) \leftarrow T$ 
```

Poznámka (Složitost BFS a DFS). Oba algoritmy pracují v čase $\mathcal{O}(n + m)$ a prostoru $\Theta(n + m)$.

Poznámka. Algoritmus DFS detekuje cykly a klasifikuje hrany

Topologické třídění.

Acyklické orientované grafy (DAG) lze uspořádat, aby vedly hrany po směru uspořádání.

Definice (Topologické uspořádání). Lineární uspořádání $\prec$ na vrcholech grafu je topologické, pokud $\forall \{x, y\} \in E : x \prec y$.

Věta. Orientovaný graf má topologické uspořádání $\iff$ je DAG.

Věta. DFS opouští vrcholy v pořadí opačné topologickému.

Nejkratší cesty v ohodnocených grafech.

Pro hledání nejkratší cesty v neohodnoceném grafu můžeme použít BFS/DFS, ale pro ohodnocené grafy otřebujeme větší kanón – Dijkstrův/Bellman-Fordův algoritmus. Dijkstrův algoritmus funguje pouze pro nezáporné hrany, Bellman-Ford i pro záporné.

Algorithm 13: Dijkstra (G, v_0)

```

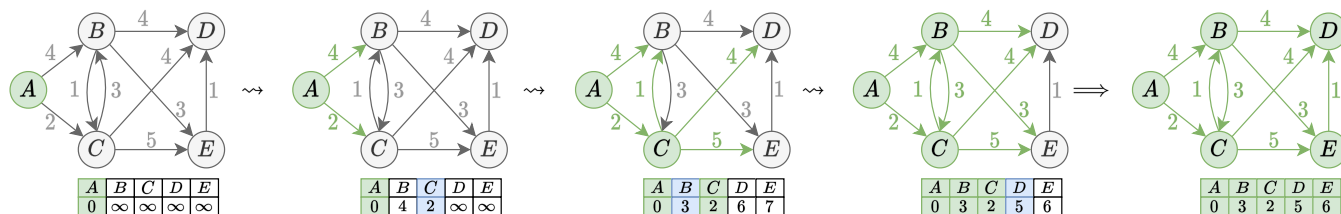
1 for každý  $v \in V$  do
2   stav( $v$ )  $\leftarrow$  nenalezený
3    $h(v) \leftarrow +\infty$ 
4    $P(v) \leftarrow$  nedefinováno
5 end
6 stav( $v_0$ )  $\leftarrow$  otevřený
7  $h(v_0) \leftarrow 0$ 
8 while  $\exists v$  otevřený do
9    $v \leftarrow$  vrchol s nejmenší hodnotou  $h(v)$ 
10  for každý následník  $w$  vrcholu  $v$  do
11    if  $h(w) > h(v) + l(v, w)$  then
12       $h(w) \leftarrow h(v) + l(v, w)$ 
13      stav( $w$ )  $\leftarrow$  otevřený
14       $P(w) \leftarrow v$ 
15    end
16  end
17  stav( $v$ )  $\leftarrow$  uzavřený
18 end

```

▷ pole vzdáleností/ ohodnocení
▷ pole předchůdců

▷ $l(v, w)$ značí délku cesty mezi v a w

Poznámka (Složitost Dijkstrova algoritmu). Časová složitost je $\mathcal{O}(n^2)$ nebo $\mathcal{O}(m + n \log n)$ s prioritní frontou.



Obrázek 2.1 Vždy vybereme vrchol s nejmenší cenou, který ještě nebyl vybráný.

Algoritmus (Bellman-Ford). Čas $\mathcal{O}(mn)$.

- Hledání nejkratší cesty v ohodnoceném orientovaném grafu včetně záporných hran.
- Dívám se na hrany a aktualizuji vrcholy.
- Fronta podle pořadí otevřených vrcholů $\rightsquigarrow$ můžu otevírat už jednou zavřené vrcholy.

Poznámka. Dijkstra pracuje pouze s nezáporným ohodnocením, Bellman-Ford i se záporným.

Minimální kostry grafu.

Definice (Váha). Váha je funkce $w : E \rightarrow \mathbb{R}$. Váha $w(H)$ podgrafu $H \subseteq G$ je součet vah jeho hran.

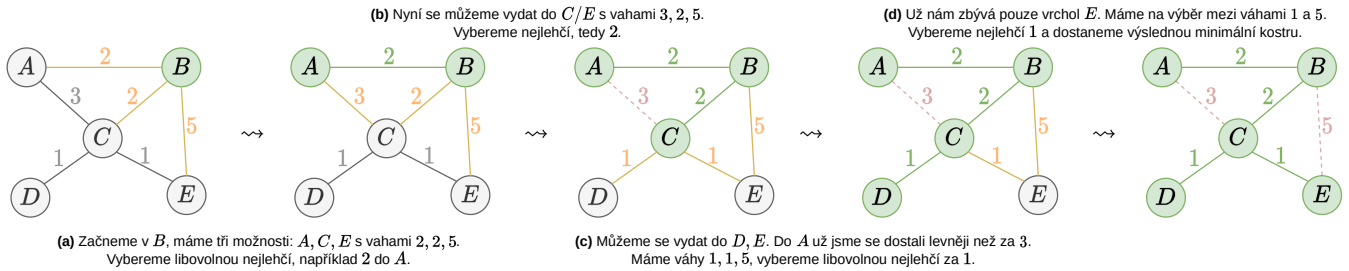
Definice (Minimální kostra). Kostra grafu je minimální, pokud má mezi kostrami nejmenší váhu.

Algorithm 14: Jarník (G, r) — G je souvislý s unikátními vahami. Vráť minimální kostru.

```

1  $v_0 \leftarrow$  libovolný vrchol grafu
2  $T \leftarrow$  strom obsahující  $v_0$  a žádné hrany
3 while  $\exists \{u, v\} \in E : u \in V_T \wedge v \notin V_T$  do
4   | hranu  $e = \{u, v\}$  s nejmenší takovou vahou přidáme do  $T$ 
5 end
6 return  $T = (V_T, E_T)$ 

```



Poznámka (Složitost Jarníkova algoritmu). V každém kroku Jarníkův algoritmus přidá nejlevnější hranu spojující strom s novým vrcholem. Časová složitost je $\mathcal{O}(|V|^2)$ nebo $\mathcal{O}(|E| + |V| \log |V|)$ s prioritní frontou

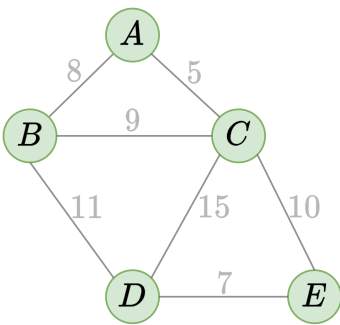
Algorithm 15: Borůvka (G)

```

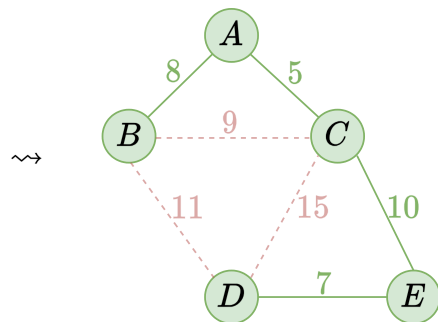
1  $T \leftarrow (V, \emptyset)$  ▷ triviální les
2 while  $T$  není souvislý do
3   | Rozklad  $T$  na komponenty souvislosti  $T_1, \dots, T_k$ 
4   | Pro každý strom  $T_i$  najdeme nejlehčí hranu  $e_i$  mezi  $T_i$  a zbytkem grafu
5   |  $T \leftarrow T \cup \{e_1, \dots, e_k\}$ 
6 end
7 return  $T$ 

```

(a) Každý vrchol je komponenta souvislosti.



(c) Propojíme nejlevnější hranou vzniklé dvě komponenty.



(B) Pro každý vrchol najdeme souseda s nejmenší vahou.

$A \xrightarrow{5} C, B \xrightarrow{8} A, C \xrightarrow{5} A, D \xrightarrow{7} E, E \xrightarrow{7} D$

Poznámka (Složitost Borůvkova algoritmu). Borůvkův algoritmus v každé iteraci přidá nejlevnější hranu z každé komponenty. Zároveň pracuje i s nesouvislými grafy. Časová složitost je $\mathcal{O}(|E| \log |V|)$. Vhodný pro řídké grafy.

Poznámka (Jarník VS Borůvka). Jarník přidává v každém kroku jeden vrchol a pracuje s jedním stromem (souvislý graf). Borůvka naproti tomu přidává více hran najednou a pracuje s více komponentami (nesouvislý graf)

Toky v sítích.

Definice jednotlivých pojmů viz kapitola Teorie grafů (1.4.9).

Algorithm 16: Ford-Fulkerson – *Princip hledání maximálního toku v síti.*

```
1  $f \leftarrow$  nulový tok  
2 while existuje zlepšující cesta  $P$  ze  $z \rightarrow s$  do  
3    $\varepsilon \leftarrow \min_{e \in E(P)} r(e).$   $\triangleright$  Rezerva  $r(uv) = c(uv) - f(uv) + f(vu)$   
4   Zvětšíme tok  $f$  podél  $P$  o  $\varepsilon$ .  $\triangleright \forall e$  po směru zvětšíme a proti směru zmenšíme  $f(e)$   
5 end  
6 return tok  $f$ .
```

2.3 Programovací jazyky (Java)

2.3.1 Základy

Interface specifikuje kontrakty třídy – vícenásobná dědičnost.

Abstrakce oddělení interface od implementace.

Polymorfismus schopnost jedné reference odkazovat na objekty různých typů.

Dědičnost opětovné použití existující třídy – “**extends**”.

implements implementuje interface – třída bude mít definované metody.

static člen třídy není vázán na instanci/ objekt, ale na celou třídu.

protected přístupu v rámci třídy a všech podtříd, ale mimo třídu neviditelná.

super() volání konstruktora předka (rodiče) – když dědí třída.

```
1 public interface Animal {
2     void makeSound();           // veřejná abstraktní metoda
3     ...
4 }
5
6 public class Pet {
7     protected String name;      // přístupná v rámci třídy a v podtřídách
8     public void greet() {
9         System.out.println("Hello-" + name);
10    }
11 }
12
13 public class Dog extends Pet implements Animal {    // dedení a implementace rozhraní
14     public Dog(String name) {
15         super(name);           // volání předka, přístup k chráněnému atributu z nadtridy
16     }
17
18     @Override
19     public void makeSound() {    // implementace metody z interface
20         System.out.println("Woof");
21     }
22
23     @Override
24     public void greet() {        // přepsání metody z předka/nadtridy (polymorfismus)
25         System.out.println("Dog-" + name + "-says:-Woof!");
26     }
27 }
28
29 public class Main {
30     public static void main(String[] args) {
31         Animal myDog = new Dog("Bob");    // polymorfismus: použití rozhraní jako typu
32         myDog.makeSound();                // volání metody z interface
33
34         Pet myPet = (Pet) myDog;          // přetypování na nadtridu
35         myPet.greet();                    // volání přepsané metody z předka
36     }
37 }
```

2.3.2 Vlákna

Používají se primárně dva způsoby – `Thread` a `Runnable`.

Thread.

```
1 Thread t = new MyThread(); // vytvořime instanci tridy MyThread
2 t.start();
3
4 public class MyThread extends Thread { // trida MyThread dedi od tridy Thread
5     @Override
6     public void run() { // prepisujeme metodu run
7         System.out.println("Thread-is-running");
8     }
9 }
```

Runnable.

```
1 Runnable r = () -> System.out.println("Runnable-is-running");
2 Thread t = new Thread(r); // vytvořime instanci Thread a predame ji instanci Runnable
3 t.start(); // spustime vlakno
```

Synchronizace.

Snaha zajistit, aby “něco” probíhalo pouze v jednom vlákně najednou.

```
1 public void increment() {
2     synchronized(this) { // k teto casti kodu se dostane jen jeden thread najednou
3         this.count++;
4     }
5 }
6
7 // nebo
8
9 public synchronized void increment() { // synchronizace cele metody
10     this.count++;
11 }
```

Terminologie vláken:

- **volatile** $\equiv$ viditelnost hodnoty mezi vlákny.
- **wait()** $\equiv$ vlákno se uspí a čeká na probuzení.
- **notify()** $\equiv$ probudí čekající vlákno.
- **notifyAll()** $\equiv$ probudí všechna čekající vlákna.
- **AtomicInteger** $\equiv$ `synchronized` na steroidech (rychlé, bez blokace).

```
1 private volatile boolean running = true;
2 public void stop() {
3     running = false; // bez synchronized, ale s volatile
4 }
5
6 private AtomicInteger count = new AtomicInteger(0);
7 public void increment() {
8     count.incrementAndGet(); // atomicka operace
9 }
```

2.3.3 Lambda funkce

```
1 // bez lambda funkce:
2 isEven(x) {
3     return x % 2 == 0;
4 }
5
6 // lambda funkce:
7 isEven = x % 2 == 0;
8
9
10 // bez lambda funkce:
11 for (String name : names) {
12     System.out.println(name);
13 }
14
15 // lambda funkce:
16 names.forEach(name -> System.out.println(name));
```

2.3.4 Race condition a Deadlock

Race condition $\equiv$ dvě nebo více vláken přistupuje k datům současně a výsledek závisí na načasování (přepíše se v běhu).

Například: `increment() {count++;}` – více vláken najednou upravuje `count`.

Řešení: použít `synchronized` nebo `AtomicInteger`.

Deadlock $\equiv$ dvě a více vláken čekání na zámky, které se vzájemně drží.

2.4 Architektura počítačů a operačních systémů

BigEndian (MSB) $0x1A2B \rightsquigarrow$

1A	2B
----	----

LittleEndian (LSB) $0x1A2B \rightsquigarrow$

2B	1A
----	----

Operace:

OR ... “|” ... *alespoň 1*

AND ... “&” ... *oboje*

XOR ... “^” ... *právě 1*

SHL/SHR ... “<<” ... *posun k MSB doleva (resp. >> doprava)*

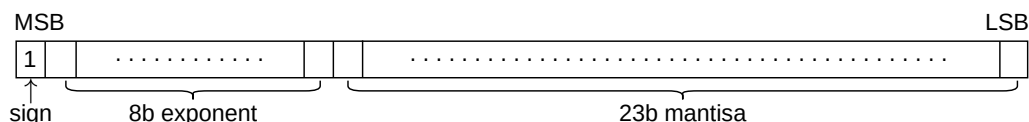
NOT ... “~” ... *opak*

Reprezentace záporných čísel.

- Stejně jako celá, ale 1. bit znaménka. (*v praxi se nepoužívá*)
- **Jedničkový doplněk:** u záporných se prohodí 1 a 0. Funguje porovnávání i sčítání, akorát máme dvě nuly (0^+ , 0^-).
- **Dvojkový doplněk:** MSB je znaménkový bit. Řeší dvě nuly. Negace je doplněk bitů +1.

Reprezentace desetinných čísel.

Za pomoci *znaménka*, *exponentu* a *mantisy*.



Příklad: Reprezentace čísla -234.75

- Číslo je záporné, proto *sign* bude 1.
- Převédeme $234_{10} = 11101010_2$.
- Převédeme $0.75_{10} = .11_2$ na dvojkovou soustavu:

$$0.75 \cdot 2 = 1.5 \rightsquigarrow 1 + 0.5$$

$$0.5 \cdot 2 = 1 \rightsquigarrow 1 + 0$$

$$0 \cdot 2 = 0 \quad \text{stop}$$

- Výsledek je tedy: 11101010.11_2

Sdílení procesoru.

- **Proces:** nezávislá běhová entita v OS. Má vlastní memory space.
- **Thread:** v rámci procesu, sdílí procesovou paměť a “další věci”.

Virtuální paměť.

- **Stránkování:** *Virtual Address Space* je rozdělena do stejných částí (*stránek*) velikosti 2^n . Fyzická *Address Space* je také rozdělena na stejné velikosti (*frame*). Platí “ $1\text{ page} = 1\text{ frame}$ ”. Také “ $1\text{ level page} = 32\text{ bit}$ ”.

Race condition více vláken najednou přistupuje ke stejným datům.
Pomůže *protected* nebo *scheduling*.

Synchronizace.

- **Active:** instrukce spuštěné během čekání na přístup.
- **Passive:** část procesu je zablokována, dokud není povolen přístup.
- **SpinLock:** *active* za použití TSL/CAS. Ideální pro krátké čekání – malá latence.
- **Semaphore:** protected counter a fronta na čekání (threads). Kontroluje přístup ke zdroji mezi více vlákny. Příkazy UP ($++\text{counter}$), DOWN ($-\text{counter}$).
- **Mutex:** mutual exclusion. V podstatě semafor s counterem 1 – kontrola, že jen jeden thread přistupuje v čase. Příkazy LOCK, UNLOCK.

Deadlock Vzájemné čekání, které vede k zacyklení.
Nutné podmínky k zacyklení:

1. Mutex – alespoň jeden je exclusive.
2. Hold & wait – jeden drží žádost pro ten druhý.
3. No preemption – resources nemůžou být znovu přijaty bez újmy.
4. Circular wait.

```
1 mutex m1, m2;
2
3 // Thread1      // Thread2
4 m1.lock();      m2.lock();
5 m2.lock();      m1.lock();
6 m2.unlock();    m1.unlock();
7 m1.unlock();    m2.unlock();
```

Obecná informatika

Z následujících 4 témat budou vybrána 4.

3.1 Kombinatorika

3.1.1 Vytvořující funkce

Definice (Vytvořující funkce). Vytvořující funkce posloupnosti $a_0, a_1, \dots = (a_n)_{n=0}^\infty \in \mathbb{R}$ je funkce proměnné x definována jako součet $f(x) = \sum_{n=0}^\infty a_n x^n$.

Gen. funkce	Posloupnost	Vzorec
$\sum_{n=0}^\infty x^n$	$(1, 1, 1, 1, \dots)$	$\frac{1}{1-x}$
$\sum_{n=0}^\infty a^n x^n$	$(1, a, a^2, a^3, \dots)$	$\frac{1}{1-ax}$
$\sum_{n=0}^\infty (x^2)^n$	$(1, 0, 1, 0, \dots)$	$\frac{1}{1-x^2}$
$\sum_{n=0}^\infty (-1)^n x^n$	$(1, -1, 1, -1, \dots)$	$\frac{1}{1+x}$
$\sum_{n=0}^\infty n x^n$	$(0, 1, 2, 3, 4, \dots)$	$\frac{x}{(1-x)^2}$

Tabulka 3.2 Typické vytvořující funkce a jejich vlastnosti.

Příklad (Použití vytvořujících funkcí k řešení lineárních rekurencí). V několika krocích:

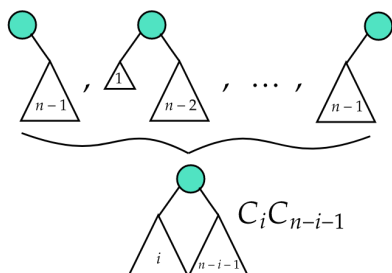
- Na vstupu máme rekurentní rovnici pro a_n .
- Rovnici vynásobíme x^n . (dostaneme tak $a_n x^n$)
- Přepíšeme jako sumu, tedy $\sum_{n=0}^\infty a_n x^n$.
- Sumy si pro zjednodušení rozepíšeme: $S_i = a_2^2 + a_3 x^3 + \dots = f(x) - a_1 x - a_0$, apod.
- Sumy dáme opět do rovnosti (např. $S_1 = S_2 - S_3$) a vyjádříme funkci $f(x)$.
- Pokud je to potřeba, rozložíme $f(x)$ na parciální zlomky.
- Rozepíšeme $f(x)$ jako mocninnou řadu a získáme explicitní vzorec pro a_n .

Věta (Zobecněná binomická věta). Pro $d \in \mathbb{R}$ platí $(1+x)^d = \sum_{n=0}^{\infty} \binom{d}{n} x^n$, pro $|x| < 1$.

Definice (Catalanova čísla). Catalanova čísla $(C_n)_{n=0}^{\infty}$ udávají počet binárních stromů s n vnitřními vrcholy. Platí $\frac{1}{n+1} \binom{2n}{n}$, neboli počet korektního uzávorkování.

Příklad (Odvození Catalanova čísla). Mějme funkci:

$$C(x) := \sum_{n=0}^{\infty} C_n x^n, \quad C_0 = 1, \quad \text{a} \quad C_n = C_0 C_{n-1} + C_1 C_{n-2} + \dots + C_{n-1} C_0 = \sum_{i=0}^{n-1} C_i C_{n-i-1}$$



$$\sum_{n=1}^{\infty} C_n x^n = \sum_{n=1}^{\infty} \left(\sum_{i=0}^{n-1} C_i C_{n-i-1} \right) x^n \Rightarrow$$

$$C(x) - 1 = x \sum_{n=0}^{\infty} \left(\sum_{i=0}^n C_i C_{n-i} \right) x^n = x \cdot C^2(x)$$

Dostáváme tak: $C(x) = 1 + xC^2(x)$, což si můžeme zapsat jako kvadratickou rovnici:

$$xC^2(x) - C(x) + 1 \Rightarrow \begin{cases} \frac{1+\sqrt{1-4x}}{2x} = C^+(x) & \text{není řešením - diverguje} \\ \frac{1-\sqrt{1-4x}}{2x} = C^-(x) & \text{konverguje k 1 při } x \rightarrow 0 \end{cases}$$

Počítáme tak dál a vyjádříme vzorec pro n -tý člen:

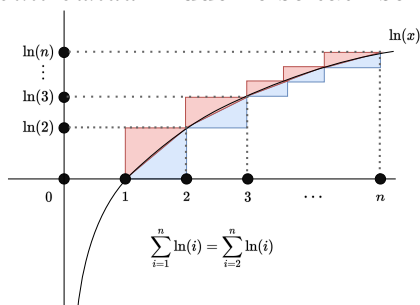
$$\begin{aligned} C_n &:= [x^n] \frac{1 - \sqrt{1-4x}}{2x} = [x^{n+1}] \frac{1 - \sqrt{1-4x}}{2} = -\frac{1}{2} [x^{n+1}] \sqrt{1-4x} = \\ &= -\frac{1}{2} (-4)^{n+1} [x^{n+1}] (1-x)^{\frac{1}{2}} \stackrel{ZBV}{=} -\frac{1}{2} (-4)^{n+1} [x^{n+1}] \binom{\frac{1}{2}}{n+1} = \\ &= (-1)^n 2^{2n+1} \cdot \frac{\frac{1}{2}(\frac{1}{2}-1)(\frac{1}{2}-2) \dots (\frac{1}{2}-n)}{(n+1)!} = (-1)^n 2^{2n+1} \cdot \frac{\frac{1}{2}(-\frac{1}{2})(-\frac{3}{2}) \dots (-\frac{2n-1}{2})}{(n+1)!} = \\ &= 2^n \cdot \frac{1 \cdot 3 \cdot \dots \cdot (2n-1)}{(n+1)!} = \frac{1 \cdot 3 \cdot \dots \cdot (2n-1)}{(n+1)n!} \cdot \frac{2 \cdot 4 \cdot \dots \cdot 2n}{n!} = \frac{1}{n+1} \frac{(2n)!}{(n!)^2} = \frac{1}{n+1} \binom{2n}{n}. \end{aligned}$$

3.1.2 Odhady faktoriálu a kombinačních čísel

Věta (Odhad faktoriálu). Platí odhad: $e \left(\frac{n}{e}\right)^n \leq n! \leq en \left(\frac{n}{e}\right)^n$.

Důkaz. Rozepíšeme $n!$ za pomoci vlastností logaritmu jako součet: $\ln(n!) = \sum_{i=1}^n \ln(i) = \sum_{i=2}^n \ln(i)$.

- *Dolní odhad:* Budeme sčítat “schody” nad křivkou:



$$\ln(n!) \geq \int_1^n \ln(x) dx = [x \ln(x) - x]_1^n = n \ln(n) - n + 1$$

$$n! \geq e^{n \ln n - n + 1} = e \left(\frac{n}{e}\right)^n$$

- *Horní odhad:* Podobně jako dolní odhad, jen budeme sčítat “schody” pod křivkou:

$$\sum_{i=1}^{n-1} \ln(i) = \ln((n-1)!) \leq n \ln(n) - n + 1$$

Ve výsledku dostaneme:

$$n \ln n - n + 1 \geq \ln((n-1)!)$$

$$e^{n \ln n - n + 1} \geq (n-1)!$$

$$n \cdot e^{n \ln n - n + 1} \geq n!$$

$$n \cdot e \left(\frac{e}{n}\right)^n \geq n!$$

□

Věta (Odhad kombinačního čísla). Pro $1 \leq k \leq n$ platí: $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$.

Důkaz. Budeme využívat vztahu $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

- *Dolní odhad:*

$$\binom{n}{k} = \frac{n(n-1)(n-2) \cdot \dots \cdot (n-k+1)}{k(k-1) \cdot \dots \cdot 1} = \frac{n}{k} \cdot \frac{n-1}{k-1} \cdot \frac{n-2}{k-2} \cdot \dots \cdot \frac{n-k+1}{1} \geq \left(\frac{n}{k}\right)^k$$

Nerovnost platí, protože $\frac{n}{k}$ je nejmenší a zbytek je rostoucí posloupnost.

- *Horní odhad:*

$$\binom{n}{k} = \frac{n(n-1)(n-2) \cdot \dots \cdot (n-k+1)}{k!} \leq \frac{n^k}{\left(\frac{k}{e}\right)^k} = \left(\frac{e \cdot n}{k}\right)^k$$

Nerovnost platí, protože $\left(\frac{k}{e}\right)^k$ je dolní odhad $k!$.

□

Věta (Odhad binomického čísla $\binom{2m}{m}$). Platí odhad: $\forall m \in \mathbb{N}_0 : \frac{4^m}{2\sqrt{m}} \leq \binom{2m}{m} \leq \frac{4^m}{\sqrt{2m}}$.

Důkaz. Definujme $P := \frac{\binom{2m}{m}}{2^{2m}}$ a dokažme, že $\frac{1}{2\sqrt{m}} \leq P \leq \frac{1}{\sqrt{2m}}$.

$$P := \frac{\binom{2m}{m}}{2^{2m}} = \frac{\frac{(2m)!}{m! \cdot m!}}{\underbrace{2 \cdot 2 \cdot \dots \cdot 2}_{2m}} = \frac{1 \cdot 2 \cdot 3 \cdot \dots \cdot 2m}{(2 \cdot 4 \cdot \dots \cdot 2m)(2 \cdot 4 \cdot \dots \cdot 2m)} = \frac{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2m-1)}{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2m}$$

- *Horní odhad:*

$$\begin{aligned} P^2 &= \frac{1 \cdot 1 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot \dots \cdot (2m-1) \cdot (2m-1)}{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdot \dots \cdot (2m) \cdot (2m)} = \\ &= 1 \cdot \frac{1 \cdot 3}{2 \cdot 2} \cdot \frac{3 \cdot 5}{4 \cdot 4} \cdot \frac{5 \cdot 7}{6 \cdot 6} \cdot \dots \cdot \frac{(2m-3) \cdot (2m-1)}{(2m-2) \cdot (2m-2)} \cdot \frac{2m-1}{(2m) \cdot (2m)} \leq \\ &\leq \frac{2m-1}{(2m) \cdot (2m)} < \frac{1}{2m}, \text{ a proto tedy } P \leq \frac{1}{\sqrt{2m}}. \end{aligned}$$

- *Dolní odhad:*

$$\begin{aligned}
 P^2 &= \frac{1 \cdot 1 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot \dots \cdot (2m-1) \cdot (2m-1)}{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdot \dots \cdot (2m) \cdot (2m)} = \\
 &= \frac{1}{2} \cdot \frac{3 \cdot 3}{2 \cdot 4} \cdot \frac{5 \cdot 5}{4 \cdot 6} \cdot \dots \cdot \frac{(2m-1) \cdot (2m-1)}{(2m-2) \cdot (2m)} \cdot \frac{1}{2m} \geq \\
 &\geq \frac{1}{4m}, \text{ a proto tedy } P^2 \geq \frac{1}{4m} \text{ a } P \geq \frac{1}{2\sqrt{m}}.
 \end{aligned}$$

□

3.1.3 Ramseyovy věty

Definice (Klika). Klika v grafu $G = (V, E)$ je množina vrcholů, t.ž. každé dva jsou spojené hranou. Největší kliku G značíme $\omega(G)$.

Definice (Nezávislá množina). Nezávislá množina v grafu $G = (V, E)$ je množina vrcholů, t.ž. žádné dva nejsou spojené hranou. Největší nezávislou množinu G značíme $\alpha(G)$.

Definice (Ramseyovo číslo).

$$R(k, l) = \min\{N \in \mathbb{N} \mid \text{každý graf } G \text{ na alespoň } N \text{ vrcholech obsahuje } \omega(G) \geq k \text{ nebo } \alpha(G) \geq l\}.$$

Věta (Ramseyova grafová dvoubarevná). Necht $k, l \in \mathbb{N}$, potom existuje $N \in \mathbb{N}$ takové, že ve všech grafech s alespoň N vrcholy, platí $\omega(G) \geq k$ nebo $\alpha(G) \geq l$.

Věta (Ramseyova – nekonečná pro dvojice). Pro každé $t \in \mathbb{N}$ (počet barev) a obarvení $c: \binom{\mathbb{N}}{2} \rightarrow \{1, 2, \dots, t\}$ existuje nekonečná množina $A \subseteq \mathbb{N}$ taková, že pro všechny dvojice $\{a_1, a_2\} \in \binom{A}{2}$ má $c(\{a_1, a_2\})$ stejnou hodnotu.

Důkaz (nekonečnou indukcí). Postupným rozebíráním množiny $\mathbb{N}$ dojdeme až k hledané množině A . Indukci začneme tím, že označíme $A_1 = \mathbb{N}$ a nastavíme $i = 1$.

V množině A_i zafixujeme libovolný vrchol v_i , zbylé vrcholy $v \in A_i \setminus \{v_i\}$ rozdělíme do množin $B_i^1, B_i^2, \dots, B_i^t$ podle barvy $c(\{v_i, v\})$ hrany, která je spojuje s v_i . Víme, že množina A_i je nekonečná, proto alespoň jedna z uvedených stejnobarevných podmnožin B_i^j musí být také nekonečná. Tuto množinu označíme za A_{i+1} a pokračujeme v indukci.

Nemáme zaručeno, že pokaždé vybereme podmnožinu stejné barvy, podle které jsme vybrali předchozí množinu. Protože je ale indukce nekonečná, alespoň jednu barvu musíme zvolit nekonečněkrát. Vrcholy v_i , které odpovídají této barvě, tvoří hledanou množinu A . □

Věta (Ramseyova – nekonečná pro p -tice). Pro každé $t \in \mathbb{N}$ (počet barev), $p \in \mathbb{N}$ a obarvení $c: \binom{\mathbb{N}}{p} \rightarrow \{1, 2, \dots, t\}$ existuje nekonečná množina $A \subseteq \mathbb{N}$, taková že pro všechny p -tice $\{a_1, \dots, a_p\} \in \binom{A}{p}$ má $c(\{a_1, \dots, a_p\})$ stejnou hodnotu.

Věta (Ramseyova – konečná pro p -tice). Pro každé $t \in \mathbb{N}$ (počet barev), $p \in \mathbb{N}$, existuje $N \in \mathbb{N}$ t.ž. $\forall n \geq N$ a obarvení $c: \binom{\{1, \dots, n\}}{p} \rightarrow \{1, 2, \dots, t\}$ existuje konečná množina $A \subseteq \{1, \dots, n\}$ velikosti k , taková, že pro všechny p -tice $\{a_1, \dots, a_p\} \in \binom{A}{p}$ má $c(\{a_1, \dots, a_p\})$ stejnou hodnotu.

Odhady Ramseyova čísla.

Věta. Necht $R(k, 1) = R(1, l) = 1$. Pak platí $R(k, l) \leq R(k-1, l) + R(k, l-1) - 1$ a z toho:

$$R(k, l) \leq \binom{k+l-2}{k-1} = \binom{k+l-2}{l-1}.$$

Věta. Necht $\binom{n}{k} \cdot \left(\frac{1}{2}\right)^{\binom{k}{2}-1} < 1$. Potom $R(k, k) > n$.

Důsledek. Pro $\forall k \geq 3: R(k, k) > 2^{k/2}$.

3.1.4 Extremální kombinatorika

Extremální kombinatorika studuje *maximální* nebo *minimální velikosti množin* (nebo struktur), které splňují určité vlastnosti, často za podmínek *vylučujících jiné konfigurace*. Například *kolik hran může mít graf bez trojúhelníku*.

Definice. Pro $n \in \mathbb{N}$ a graf H definujeme $\text{ex}(n, H)$ jako největší $|E|$ na n vrcholech v grafu $G = (V, E)$, neobsahující H jako podgraf.

$$\text{ex}(n, H) = \max\{|E|; G = (V, E) : |V| = n, H \subseteq G\}.$$

Příklad. Například $\text{ex}(n, K_3) = \left| E \left(K_{\lfloor \frac{n}{2} \rfloor, \lceil \frac{n}{2} \rceil} \right) \right| = \lfloor \frac{n}{2} \rfloor \cdot \lceil \frac{n}{2} \rceil = \frac{n^2}{4}$

Definice (Turánův graf). Turánův graf na n vrcholech s r partitami, značený $T_{n,r}$, je úplný r -partitní graf na n vrcholech, jehož partity mají velikosti $\lfloor \frac{n}{r} \rfloor$ a $\lceil \frac{n}{r} \rceil$. Potom $t_{n,r}$ = počet hran $T_{n,r}$.

Věta (Turán). $\forall r \geq 2 : \text{ex}(n, K_{r+1}) = t_{n,r} = \left(1 - \frac{1}{r}\right) \frac{n^2}{2}$.

(Neboli, jak by měl vypadat graf s co nejvíce hranami, aby neobsahoval K_r jako podgraf).

Definice (k -uniformní hypergraf). k -uniformní hypergraf je dvojice (V, E) , kde E je množina k -prvkových podmnožin V .

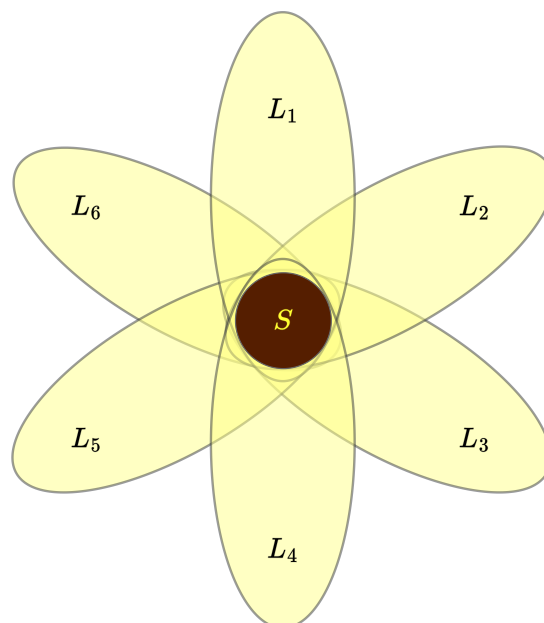
Definice. $f(n, k) :=$ Největší počet hyperhran v k -uniformním hypergrafu na n vrcholech, v němž žádné dvě hyperhrany nejsou disjunktní.

Poznámka. Pro $n < k : f(n, k) = 0$, protože neexistují hyperhrany.

Poznámka. Pro $k \leq n < 2k : f(n, k) = \binom{n}{k}$, protože každé dvě množiny z $\binom{V}{k}$ se protínají.

Definice (Slunečnice). Slunečnice se středem S a l lístky je l -tice množin $L_1, \dots, L_l$ taková, že $\forall i \neq j : L_i \cap L_j = S$. Tedy $E = \left\{ \{1\} \cup e' \mid e' \in \binom{\{2, \dots, n\}}{k-1} \right\}$.

Věta (Erdős-Ko-Rado). Pro libovolné k a $n \geq 2k$ platí $f(n, k) = \binom{n-1}{k-1}$.



Obrázek 3.2 Příklad slunečnice pro $l = 6$, s množinami $L_1, \dots, L_6$ a středem $S = L_1 \cap L_2 \cap L_3 \cap L_4 \cap L_5 \cap L_6$.

3.1.5 Samoopravné kódy

Definice (Abeceda a slovo). Abeceda Σ je konečná množina symbolů. Slovo délky n je posloupnost n symbolů. Všechna slova délky n je Σ^n .

Definice (Hammingova vzdálenost). Pro $x, y \in \mathbb{Z}_2^n$ je Hammingova vzdálenost

$$d(x, y) := |\{i \in [n] : x_i \neq y_i\}|.$$

Definice (Hammingova váha). Hammingova váha $\|x\| := |\{i \in [n] : x_i \neq 0\}|$.

Definice (Minimální vzdálenost). Minimální vzdálenost pro kód $C \in \mathbb{Z}_2^n$ je minimum z Hammingových vzdáleností:

$$\Delta(C) := \min_{\substack{x, y \in C \\ x \neq y}} d(x, y).$$

Definice $((n, k, d)$ -kód). (n, k, d) -kód je množina $C \in \mathbb{Z}_2^n$ taková, že $|C| = 2^k$ a $\Delta(C) = d$.

Definice (Lineární kód). Lineární kód je kód $C \in \mathbb{Z}_2^n$, který je vektorový podprostor $\mathbb{Z}_2^n$.

Definice (Generující matice kódu C). Generující matice kódu C pro lineární (n, k, d) -kód je matice $G \in \mathbb{Z}_2^{k \times n}$, jejíž řádky tvoří bázi C .

Definice (Kódování). Nechť C je (n, k, d) -kód pro $k \in \mathbb{N}$, tak kódování pro C je bijekce $\mathbb{Z}_2^k \rightarrow C$.

Definice (Dekódování). Dekódování (n, k, d) -kódu C je funkce $g : \mathbb{Z}_2^n \rightarrow C$ taková, že

$$\forall x \in \mathbb{Z}_2^n : d(x, g(x)) = \min_{y \in C} d(x, y)$$

. Neboli přiřazujeme nejbližší slovo; slovo s nejmenší vzdáleností.

Definice (Duální kód k C “ortogonální doplněk”). Duální kód k C je

$$C^\perp := \{\langle x, y \rangle = 0 \mid y \in \mathbb{Z}_2^n, \forall x \in C\}.$$

Definice (Kontrolní matice). Nechť C je lineární (n, k, d) -kód. Kontrolní matice kódu C je matice, jejíž řádky tvoří bázi $C^\perp$.

Definice (Hammingovy kódy). Nechť $r \in \mathbb{N}, r \geq 2$, nechť K_r je matice s r řádky a $2^r - 1$ sloupci, jejíž sloupce jsou nenulové a různé. Potom Hammingovy kódy H_r jsou kódy s kontrolní maticí K_r .

Definice (Koule). Koule $B(x, t) := \{d(x, y) \leq t \mid y \in \mathbb{Z}_2^n\}$ je okolí poloměru t kolem x v $\mathbb{Z}_2^n$.

Věta (Hammingův odhad). Pokud existuje (n, k, d) -kód C , tak pro $t = \lfloor \frac{d-1}{2} \rfloor$ a $V(n, t) = \sum_{i=0}^t \binom{n}{i}$:

$$2^k \leq \frac{2^n}{V(n, t)}.$$

Důkaz. Plyne z toho, že $x, y \in C$, kde $x \neq y$: $B(x, \lfloor \frac{d-1}{2} \rfloor) \cap B(y, \lfloor \frac{d-1}{2} \rfloor) = \emptyset$. □

Definice (Perfektní kód). Kód C je perfektní, pokud pro něj platí Hammingův odhad s rovností.

Příklad (Příklady perfektních kódů). Perfektní kódy jsou například Hammingovy kódy nebo “opakovací kód s lichou délkou”

3.2 Diferenciální a integrální počet ve více rozměrech

3.2.1 Riemannův integrál

Jednorozměrný

Definice (Dělení). Dělení intervalu $[a, b]$ je posloupnost $P = (t_0, \dots, t_n)$, kde

$$a = t_0 < t_1 < \dots < t_{n-1} < t_n = b.$$

Definice (Horní/dolní Riemannovy součty). Pro omezenou funkci $f : J = [a, b] \rightarrow \mathbb{R}$ a dělení P definujeme dolní a horní součty:

$$s(f, P) = \sum_{j=1}^n m_j(t_j - t_{j-1}), \quad \text{resp.} \quad S(f, P) = \sum_{j=1}^n M_j(t_j - t_{j-1}),$$

kde

$$m_j = \inf\{f(x) : t_{j-1} \leq x \leq t_j\}, \quad a \quad M_j = \sup\{f(x) : t_{j-1} \leq x \leq t_j\}.$$

Definice (Horní/dolní Riemannův integrál). Horní a dolní Riemannův integrál f přes $[a, b]$ je:

$$\int_a^b f(x) \, dx = \sup\{s(f, P) : P \text{ dělení}\} \quad a \quad \overline{\int}_a^b f(x) \, dx = \inf\{S(f, P) : P \text{ dělení}\}.$$

Definice (Riemannův integrál funkce). Riemannův integrál funkce f přes $[a, b]$ je:

$$(R) \int_a^b f(x) \, dx, \text{ pokud } \int_a^b f(x) \, dx = \overline{\int}_a^b f(x) \, dx.$$

Vícerozměrný

Definice (n -rozměrný kompaktní interval). n -rozměrný kompaktní interval ($v \mathbb{E}_n$) je

$$J = [a_1, b_1] \times \dots \times [a_n, b_n].$$

Definice (Dělení intervalu). Dělení intervalu J je posloupnost dělení $P = (P^1, \dots, P^n)$:

$$P^j : a_j = t_{j,0} < t_{j,1} < \dots < t_{j,n_j-1} < t_{j,n_j} = b_j.$$

Definice (Cihly intervalu). Intervalům

$$[t_{1,i_1}, t_{1,i_1+1}] \times \dots \times [t_{n,i_n}, t_{n,i_n+1}]$$

říkáme cihly dělení P . Množinu všech cihel značíme $\mathcal{B}(P)$.

Je to skoro disjunktní dělení intervalu J . Různé cihly z $\mathcal{B}(P)$ se totiž setkávají jen v podmnožinách okrajů, tedy v množinách objemu 0, díky čemuž platí:

$$\text{vol}(J) = \sum \{\text{vol}(B) : B \in \mathcal{B}(J)\}.$$

Definice (“Supremum/infimum” na kompaktním intervalu). Je dána omezená $f : J \rightarrow \mathbb{R}$ na n -rozměrném kompaktním intervalu J a $B \subseteq J$ je n -rozměrný kompaktní podinterval intervalu J . Položme

$$m(f, B) = \inf\{f(\mathbf{x}) : \mathbf{x} \in B\} \quad a \quad M(f, B) = \sup\{f(\mathbf{x}) : \mathbf{x} \in B\}.$$

Definice (Horní/dolní součty). Pro dělení P intervalu J a omezenou funkci $f : J \rightarrow \mathbb{R}$ definujeme

$$s_J(f, P) = \sum \{m(f, B) \cdot \text{vol}(B) : B \in \mathcal{B}(P)\},$$

$$S_J(f, P) = \sum \{M(f, B) \cdot \text{vol}(B) : B \in \mathcal{B}(P)\}.$$

Definice ((Horní/dolní) Riemannův integrál). Množiny $\{s(f, P) \mid P \text{ dělení}\}$ a $\{S(f, P) \mid P \text{ dělení}\}$ jsou shora/zdola omezené a můžeme definovat dolní/horní Riemannův integrál funkce f přes J jako

$$\int_J f(\mathbf{x}) d\mathbf{x} = \sup\{s(f, P) \mid P \text{ dělení}\} \quad a \quad \overline{\int}_J f(\mathbf{x}) d\mathbf{x} = \inf\{S(f, P) \mid P \text{ dělení}\}.$$

Jsou-li si rovný, máme Riemannův integrál funkce f přes J , značíme:

$$\int_J f(\mathbf{x}) d\mathbf{x} \quad \text{nebo prostě} \quad \int_J f$$

Věta (Kritérium existence Riemannova integrálu). Riemannův integrál $\int_J f(\mathbf{x}) d\mathbf{x}$ existuje právě když $\forall \varepsilon > 0$ existuje dělení P takové, že

$$S_J(f, P) - s_J(f, P) < \varepsilon$$

Věta (Riemannův integrál pro spojitě funkce). Každá spojitá funkce $f : J \rightarrow \mathbb{R}$ na n -rozměrném kompaktním intervalu má Riemannův integrál $\int_J f$.

Věta (Fubiniova věta). Vezměme součin $J = J' \times J'' \subseteq \mathbb{E}_{m+n}$ intervalů $J' \subseteq \mathbb{E}_m$, $J'' \subseteq \mathbb{E}_n$. Nechť existuje

$$\int_J f(\mathbf{x}, \mathbf{y}) d\mathbf{x}\mathbf{y}$$

a nechť pro každé $\mathbf{x} \in J'$, resp. $\mathbf{y} \in J''$, existuje

$$\int_{J'} f(\mathbf{x}, \mathbf{y}) d\mathbf{x} \quad a \quad \int_{J''} f(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

Potom je

$$\int_J f(\mathbf{x}, \mathbf{y}) d\mathbf{x}\mathbf{y} = \int_{J'} \left(\int_{J''} f(\mathbf{x}, \mathbf{y}) d\mathbf{y} \right) d\mathbf{x} = \int_{J''} \left(\int_{J'} f(\mathbf{x}, \mathbf{y}) d\mathbf{x} \right) d\mathbf{y}$$

3.2.2 Funkce více proměnných

Definice (Parciální derivace). Pro funkci $f(x_1, \dots, x_n)$ a bod $\mathbf{a}$ definujeme funkci

$$\phi_k(t) = f(a_1, \dots, a_{k-1}, t, a_{k+1}, \dots, a_n)$$

Parciální derivace⁴ funkce f podle proměnné x_k v bodě $\mathbf{a}$ je (obvyklá) derivace funkce ϕ_k v bodě a_k

$$\lim_{h \rightarrow 0} \frac{\phi_k(a_k + h) - \phi_k(a_k)}{h} = \lim_{h \rightarrow 0} \frac{f(a_1, \dots, a_{k-1}, a_k + h, a_{k+1}, \dots, a_n) - f(\mathbf{a})}{h}.$$

Značíme ji: $\frac{\partial f(\mathbf{a})}{\partial x_k}$ nebo: $\frac{\partial f}{\partial x_k}(\mathbf{a})$.

Definice (Totální diferenciál). Funkce f má totální diferenciál v bodě $\mathbf{a}$, existuje-li funkce μ spojitá v okolí U bodu $\mathbf{o} \in \mathbb{R}^n$ taková, že $\mu(\mathbf{o}) = 0$ a čísla $A_1, \dots, A_n$ pro která

$$f(\mathbf{a} + \mathbf{h}) - f(\mathbf{a}) = \sum_{k=1}^n A_k h_k + \|\mathbf{h}\| \mu(\mathbf{h}).$$

S použitím skalárního součinu jde též zapsat jako

$$f(\mathbf{a} + \mathbf{h}) - f(\mathbf{a}) = \mathbf{A}\mathbf{h} + \|\mathbf{h}\| \mu(\mathbf{h})$$

Pravidlo	Vzorec pro parciální derivaci	Příklad
Součet	$\frac{\partial}{\partial x}(f(x) + g(x)) = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial x}$	$\frac{\partial}{\partial x}(x^2 + y) = 2x + 0$
Násobení	$\frac{\partial}{\partial x}(f(x)g(x)) = \frac{\partial f}{\partial x}g(x) + f(x)\frac{\partial g}{\partial x}$	$\frac{\partial}{\partial x}(x \cdot y) = 1 \cdot y + x \cdot 0$
Dělení	$\frac{\partial}{\partial x}\left(\frac{f(x)}{g(x)}\right) = \frac{\frac{\partial f}{\partial x}g(x) - f(x)\frac{\partial g}{\partial x}}{(g(x))^2}$	$\frac{\partial}{\partial x}\left(\frac{x}{y}\right) = \frac{1 \cdot y - x \cdot 0}{y^2}$
Chain rule	$\frac{\partial z}{\partial x} = \sum_{k=1}^n \frac{\partial f}{\partial u_k} \cdot \frac{\partial u_k}{\partial x}$	$z = u^2, \quad u = x + y \rightsquigarrow \frac{\partial z}{\partial x} = 2u \cdot \frac{\partial u}{\partial x} = 2(x + y) \cdot 1$

Tabulka 3.3 Pravidla pro parciální derivace

Lemma (Spojitost, parciální derivace a totální diferenciál). *Nechť má funkce f totální diferenciál v bodě $\mathbf{a}$. Potom je f spojitá v $\mathbf{a}$, a má všechny parciální derivace v $\mathbf{a}$ s hodnotami $\frac{\partial f(\mathbf{a})}{\partial x_k} = A_k$.*

Věta (Spojité parciální derivace a totální diferenciál). *Nechť má f spojité parciální derivace v okolí bodu $\mathbf{a}$. Potom má v $\mathbf{a}$ totální diferenciál.*

Věta (O záměnnosti). *Mějme funkci $f(x, y)$ takovou, že existují parciální derivace $\frac{\partial^2 f}{\partial x \partial y}$ a $\frac{\partial^2 f}{\partial y \partial x}$, které jsou spojité v nějakém okolí bodu (x, y) . Potom:*

$$\frac{\partial^2 f(x, y)}{\partial x \partial y} = \frac{\partial^2 f(x, y)}{\partial y \partial x}$$

Věta (Lagrangeova věta ve více proměnných). *Nechť má f spojité parciální derivace v konvexní otevřené množině $U \subseteq \mathbb{E}_n$. Potom pro libovolné dva body $\mathbf{x}, \mathbf{y} \in U$ $\exists 0 \leq \theta \leq 1$ takové, že:*

$$f(\mathbf{y}) - f(\mathbf{x}) = \sum_{j=1}^n \frac{\partial f(\mathbf{x} + \theta(\mathbf{y} - \mathbf{x}))}{\partial x_j} (y_j - x_j), \quad \text{resp.} \quad f(\mathbf{y}) - f(\mathbf{x}) = \nabla f(\mathbf{c})(\mathbf{y} - \mathbf{x}),$$

kde $\mathbf{c} = \mathbf{x} + \theta(\mathbf{y} - \mathbf{x})$ (konvexní kombinace $\mathbf{x}, \mathbf{y}$) a $\nabla f(p) = (\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n})^T$ je **gradient** funkce f .

Výpočet extrémů pomocí parciálních derivací.

Mějme funkci $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $f(x_1, \dots, x_n)$.

1. Najdeme stacionární body tak, že položíme první parciální derivace rovny nule:

$$\frac{\partial f}{\partial x_1}(x_1, \dots, x_n) = 0, \quad \frac{\partial f}{\partial x_2}(x_1, \dots, x_n) = 0, \quad \dots, \quad \frac{\partial f}{\partial x_n}(x_1, \dots, x_n) = 0.$$

2. V těchto bodech vyšetříme druhé derivace pomocí *Hessiánu*:

$$H_f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}, \quad \text{pro } f(x, y) \text{ jen jednoduše: } H_f = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}.$$

3. Typ extrému určíme pomocí kritéria pozitivní/negativní definitnosti Hessiánu:

- H_f pozitivně definitní $\implies$ lokální minimum, $(v \mathbb{R}^2 \text{ pokud } \det(H) > 0 \text{ a } f_{xx} > 0)$
- H_f negativně definitní $\implies$ lokální maximum, $(v \mathbb{R}^2 \text{ pokud } \det(H) > 0 \text{ a } f_{xx} < 0)$
- H_f indefinitní $\implies$ sedlový bod. $(v \mathbb{R}^2 \text{ pokud } \det(H) < 0)$

⁴Geometricky odpovídá tečně funkce v daném bodě a příslušné ose.

Existence extrémů pro funkce více proměnných.

Věta. Pokud je f spojitá na kompaktní množině $K \subset \mathbb{R}^n$, pak dle Weierstrassovy věty nabývá f na K svého maxima i minima.

Věta (Nutná podmínka pro lokální extrém). Mějme funkci $f: D \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$. Pokud má f v bodě $a \in D$ lokální extrém, potom platí jedna z podmínek:

(1) $\forall i = 1, \dots, m: \frac{\partial f(a)}{\partial x_i} = 0$, nebo derivace neexistuje.

(2) Bod a je na hranici D .

Věta (Postačující podmínka pro extrém). Mějme funkci $f: D \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$, bod a uvnitř D a dále platí, že $\frac{\partial^2 f}{\partial x_i \partial x_j}$ jsou spojitě v a , a $\frac{\partial f(a)}{\partial x_i} = 0$ (podezřelý z extrému). Pak:

(1) Pokud $D^2 f(a)$ je pozitivně definitní, pak f nabývá v a lokálního minima.

(2) Pokud $D^2 f(a)$ je negativně definitní, pak f nabývá v a lokálního maxima.

(3) Pokud $D^2 f(a)$ je indefinitní, pak f v a nenabývá lokálního extrému.

Tato matice $D^2 f(a)$ se nazývá Hessova matice a značí se $H(a)$.

Vázané extrémy.

Cílem je hledat extrémy nějakého zobrazení $f(x)$ na nějaké omezené množině $\{x: g(x) = 0\}$.

Věta (O hledání extrému funkcí s vazbami). Buďte $f, g_1, \dots, g_k$ reálné funkce definované na otevřené množině $D \subseteq \mathbb{E}_n$. Necht' mají spojitě parciální derivace. Necht' je hodnota matice

$$M = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_k}{\partial x_1} & \dots & \frac{\partial g_k}{\partial x_n} \end{pmatrix}$$

maximální, tedy $k \leq n$, v každém bodě oboru D .

Jestliže funkce f nabývá v bodě $\mathbf{a} = (a_1, \dots, a_n)$ lokálního extrému podmíněného vazbami

$$\forall i \in \{1, \dots, k\} \quad g_i(x_1, \dots, x_n) = 0$$

Pak existují čísla $\lambda_1, \dots, \lambda_k$, Lagrangeovy multiplikátory, taková, že $\forall i \in 1, \dots, n$ platí

$$\frac{\partial f(\mathbf{a})}{\partial x_i} + \sum_{j=1}^k \lambda_j \cdot \frac{\partial g_j(\mathbf{a})}{\partial x_i} = 0$$

nebo ekvivalentně přes gradienty jako $\nabla f(\mathbf{a}) + \boldsymbol{\lambda} \cdot \nabla g(\mathbf{a}) = 0$

Ilustrační příklad: Necht' $f(x, y) = xy$ a $g(x, y) = x^2 + y^2 - 1$. Podle věty hledáme

$$\nabla f(x, y) + \lambda \nabla g(x, y) = \mathbf{0}, \quad g(x, y) = 0.$$

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} + \lambda \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix} = \mathbf{0}$$

$$\begin{bmatrix} y \\ x \end{bmatrix} + \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix} = \mathbf{0} \quad \rightsquigarrow \quad \begin{aligned} y &= -2\lambda x, \\ x &= -2\lambda y. \end{aligned}$$

Máme tři rovnice o třech neznámých. Určíme $x = 4\lambda^2 x$, a máme $x = 0$, nebo $4\lambda^2 = 1$ (pro $\lambda = \pm \frac{1}{2}$). Dosadíme do $g(x, y) = x^2 + y^2 - 1$ a dopočítáme y . Následně dosadíme body do $f(x, y)$ a zjistíme, že v bodech $\left(\pm \frac{1}{\sqrt{2}}, \pm \frac{1}{\sqrt{2}}\right)$ nabývá $f = \frac{1}{2}$ maxima, v bodech $\left(\pm \frac{1}{\sqrt{2}}, \mp \frac{1}{\sqrt{2}}\right)$ nabývá $f = -\frac{1}{2}$ minima a body $(0, \pm 1)$ dávají $f = 0$ a nejsou tudíž extrémy.

3.2.3 Metrický prostor

Definice (Metrický prostor). *Nechť X je množina, $d : X \times X \rightarrow \mathbb{R}$ funkce taková, že platí:*

- $\forall x, y \in X : d(x, y) \geq 0, \quad d(x, y) = 0 \iff x = y$
- $\forall x, y \in X : d(x, y) = d(y, x)$
- $\forall x, y, z \in X : d(x, z) \leq d(x, y) + d(y, z)$ (trojúhelníková nerovnost)

pak (X, d) je metrický prostor.

Příklad. *Metrické prostory jsou například: $(\mathbb{R}, |x - y|)$, $(\mathbb{C}, |x - y|)$.*

Definice (Euklidovský prostor $\mathbb{E}_n$). *Definujeme jako metrický prostor $(\mathbb{R}^n, d)$, kde d :*

$$d((x_1, \dots, x_n), (y_1, \dots, y_n)) = \sqrt{\sum_i (x_i - y_i)^2}$$

Pro nás zvlášť důležitý, známý v podobě vektorového prostoru $\mathbb{R}^n$ se skalárním součinem $\langle \mathbf{u} | \mathbf{v} \rangle$ a normou $\|\mathbf{u}\| = \sqrt{\mathbf{u} | \mathbf{u}}$ a vzdáleností $d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|$

Definice (Diskrétní prostor). *Definujeme jako (X, d) , kde $d(x, y) = 1$ pro $x \neq y$*

Definice (Podprostor). *Bud' (X, d) metrický prostor. Pak (Y, d') je podprostor, kde $Y \subseteq X$ a $\forall x, y \in Y : d'(x, y) = d(x, y)$.*

Uzavřené a otevřené množiny

Definice (Okolí). *Nechť (X, d) je metrický prostor, pak $\forall x \in X, \exists \varepsilon > 0$:*

$$\Omega(x, \varepsilon) = \{y \in X \mid d(x, y) < \varepsilon\}.$$

Formulaci $\Omega(x, \varepsilon)$ se říká otevřená koule s poloměrem ε okolo x .

Definice (Otevřená množina). *$U \subseteq (X, d)$ je otevřená, pokud je okolím každého svého bodu, neboli*

$$\forall x \in U \exists \delta > 0 : \Omega(x, \delta) \subseteq U = (x - \delta, x + \delta) \cap U.$$

Definice (Uzavřená množina). *Nechť $V \subseteq (X, d)$. Pokud každá posloupnost $(a_n) \subseteq V$ má $\lim a_n = a \in X$, pak $a \in V$ a V nazveme uzavřenou množinou. (Alternativě, pokud je $X \setminus V$ otevřená.)*

Příklad. *Uvážíme-li interval $(0, \infty)$ s Euklidovskou metrikou, pak*

- $[1, 2]$ je pouze uzavřený $\dots (\forall x \in [1, 2] \nexists \delta > 0 : \Omega(x, \delta) \subseteq [1, 2])$
- $(0, 1]$ je také pouze uzavřený $\dots (\forall \text{ lim na } (0, 1] \text{ konvergující v } (0, \infty) \text{ mají lim v } (0, 1])$
- $(1, \infty)$ je pouze otevřený. $\dots (\text{Např } \lim a_n = \lim 1 + \frac{1}{n} = 1 \in (0, \infty), \text{ ale } 1 \notin (1, \infty).)$
- $(0, \infty)$ je jak otevřený, tak uzavřený

Spojitosť funkce na metrickém prostoru

Definice (Spojité zobrazení). *$f : (X, d) \rightarrow (Y, d')$ je spojité zobrazení, pokud*

$$(\forall x \in X)(\forall \varepsilon > 0)(\exists \delta > 0)(\forall y \in X) : d(x, y) < \delta \implies d'(f(x), f(y)) < \varepsilon.$$

Definice (Konvergence). *Posloupnost $(x_n)_n$ v metrickém prostoru (X, d) konverguje k $x \in X$, pokud*

$$\forall \varepsilon > 0 \exists n_0 : \forall n \geq n_0 : d(x_n, x) < \varepsilon.$$

Věta (Složení spojitých zobrazení je spojitý). *Pokud jsou $f : (X_1, d_1) \rightarrow (X_2, d_2)$ a $g : (X_2, d_2) \rightarrow (X_3, d_3)$ spojitý, pak je spojitý i $g \circ f : (X_1, d_1) \rightarrow (X_3, d_3)$.*

Věta (Věta o konvergenci). *Zobrazení $f : (X_1, d_1) \rightarrow (X_2, d_2)$ je spojitý $\iff$ pro každou $(x_n)_n$ konvergentní v (X_1, d_1) , posloupnost $(f(x_n))_n$ konverguje v (X_2, d_2) a platí $\lim_n f(x_n) = f(\lim_n x_n)$.*

Kompaktnost

Definice (Kompaktní množina). *Množina je kompaktní, pokud je uzavřená a omezená.*

Definice (Kompaktní metrický prostor). *Metrický prostor (X, d) je kompaktní, pokud každá posloupnost v něm obsahuje konvergentní podposloupnost.*

Věta (Extrémy spojitě funkce na kompaktním prostoru). *Bud' (X, d) kompaktní. Potom každá spojitá funkce $f : (X, d) \rightarrow \mathbb{R}$ nabývá minima i maxima (t.j. nejsou nekonečné).*

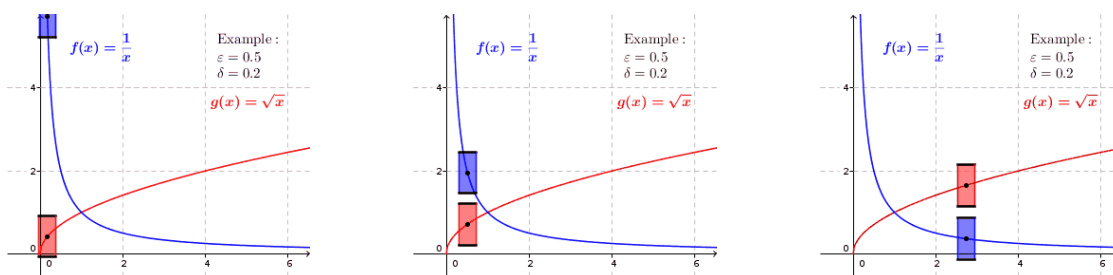
Stejnomořná spojitost

Definice (Stejnomořná spojitost). *Řekneme, že $f : (X, d) \rightarrow (Y, d')$ je stejnoměrně spojitá, pokud*

$$(\forall \varepsilon > 0)(\exists \delta > 0)(\forall x \in X)(\forall y \in X) : d(x, y) < \delta \implies d'(f(x), f(y)) < \varepsilon.$$

V porovnání s definicí normální spojitosti je rozdíl v pozici kvantifikátorů $\forall x, y$ (v definici spojitosti jsou na úplném začátku). To odpovídá tomu, že pro spojitost požadujeme okénko pouze *pro jeden bod* a ne pro celou funkci, jak je tomu u stejnoměrné spojitosti.

Intuice. Geometrický význam je ten, že pro libovolnou vzdálenost ε v y existuje δ t. ž. okénko o velikosti (δ, ε) umístěné do libovolného místa v grafu není grafem protnuto nahoře ani dole. Na obrázku 3.3 je vidět, že funkce $\sqrt{x}$ stejnosměrnou spojitost splňuje, kdežto funkce $\frac{1}{x}$ nikoliv.



Obrázek 3.3 Geometrický význam stejnosměrné spojitosti pro $f(x) = 1/x$ a $g(x) = \sqrt{x}$ na intervalu $\mathbb{R}^+$.

Věta (Spojitost zobrazení na kompaktním prostoru). *Pokud je (X, d) kompaktní, pak je každé spojitě zobrazení $f : (X, d) \rightarrow (Y, d')$ stejnoměrně spojitě.*

3.3 Pokročilé algoritmy a datové struktury

3.3.1 Dynamické programování

Princip. Začneme s rekurzivním algoritmem, který je exponenciálně pomalý. Odhalíme opakované výpočty stejných podproblémů. Uděláme si tabulku (cache), do které zapisujeme, které podproblémy jsme už vyřešili. Cache lze vyplňovat bez rekurze, zvolíme vhodné pořadí podproblémů. Získáme tak stejně rychlý, ale jednodušší algoritmus. (*Topologicky uspořádáme DAG*).

Nejdelší rostoucí podposloupnost.

Chceme najít podposloupnost která je ostře rostoucí. Problém si nejprve rozdělíme na hledání počtu prvků (největší možný).

Nechť $\text{NRP}(i)$ je délka Nejdelší Rostoucí Podposloupnosti začínající prvkem x_i .

Algorithm 17: $\text{NRP}(i)$: Nejdelší Rostoucí Podposloupnost $\Theta(n^2)$

```
1  $d \leftarrow 1$ 
2 for  $j = i + 1, \dots, n$  do
3   if  $x_j > x_i$  then
4      $d \leftarrow \max(d, 1 + \text{NRP}(j))$ 
5   end
6 end
7 return  $d$ 
```

Rekurze se můžeme zbavit a tabulku vyplňovat postupně od největšího i k nejmenšímu. Budeme tedy počítat $T[i]$, což bude délka nejdelší ze všech rostoucích podposloupností začínajících x_i .

Algorithm 18: $\text{NRP2}(i)$: Nejdelší Rostoucí Podposloupnost iterativně $\Theta(n^2)$

```
1  $x_0 \leftarrow -\infty$ 
2 for  $i = n, \dots, 0$  do
3    $T[i] \leftarrow 1$  ▷ tabulka (cache)
4   for  $j = i + 1, \dots, n$  do
5     if  $x_j > x_i$  &  $T[i] < 1 + T[j]$  then
6        $T[i] \leftarrow 1 + T[j]$  ▷ lepší řešení
7     end
8   end
9 end
10 return délka  $T[0]$ 
```

Editační vzdálenost – Levenstein $\mathcal{O}(n \cdot m)$.

Máme následující editační operace: změna/ vložení/ smazání jednoho znaku.

Definujme $L(x_1 \dots x_n, y_1 \dots y_m)$ jako délku nejkratší posloupnosti editačních operací, která převede $x_1 \dots x_n$ na $y_1 \dots y_m$.

Platí následující podmínky:

- (1) Pokud $x_1 = y_1$: $L(x_2 \dots x_n, y_2 \dots y_m)$
 - (2) **Změna** x_1 na y_1 : $1 + L(x_2 \dots x_n, y_2 \dots y_m)$
 - (3) **Vložení** y_1 : $1 + L(x_1 \dots x_n, y_2 \dots y_m)$
 - (4) **Smazání** x_1 : $1 + L(x_2 \dots x_n, y_1 \dots y_m)$
- Celkem tak máme $L(\epsilon, \epsilon) = 0$, a :
 $L(x_1 \dots x_n, y_1 \dots y_m) = \min((1)(2)(3)(4)).$

Otočíme směr výpočtu a tabulku T s výsledky podproblémů budeme vyplňovat bez použití rekurze. Představíme-li si ji jako matici, každý prvek závisí pouze na těch, které leží napravo a dolů od něj. Tabulku proto můžeme vyplňovat po řádcích zdola nahoru, zprava doleva. Tím získáme jednodušší algoritmus, který běží v čase $\Theta(nm)$.

Algorithm 19: Edit($x_1 \dots x_n, y_1 \dots y_m$):	Levenstein iterativně $\Theta(n^2)$
--	-------------------------------------

```

1 for  $i = 1, \dots, n + 1$  do  $T[i, m + 1] \leftarrow n - i + 1$ .
2 for  $j = 1, \dots, m + 1$  do  $T[n + 1, j] \leftarrow m - j + 1$ .
3 for  $i = n, \dots, 1$  do
4   for  $j = m, \dots, 1$  do
5     if  $x_i = y_j$  then  $\delta \leftarrow 0$  else  $\delta \leftarrow 1$ .
6      $T[i, j] \leftarrow \min(\delta + T[i + 1, j + 1], 1 + T[i + 1, j], 1 + T[i, j + 1])$     ▷ všechny způsoby
7   end
8 end
9 return  $L(x_1 \dots x_n, y_1 \dots y_m) = T[1, 1]$ .
```

3.3.2 Grafové algoritmy

Komponenty silné souvislosti orientovaných grafů.

Nechť $G(V)$ je orientovaný graf a $\rightarrow, \leftrightarrow$ jsou relace na V vyznačující:

- $u \rightarrow v$ je hrana z u do v ,
- $u \leftrightarrow v$ je $u \rightarrow v$ & $v \rightarrow u$.

Definice (Silná souvislost). Říkáme, že graf je silně souvislý $\iff$ relace $\leftrightarrow$ má 1 komponentu.

Definice (Komponenty silné souvislosti). Podgrafy indukované třídami $\leftrightarrow$.

Definice (Graf komponent). Pro graf G definujeme graf komponent $C(G)$ jako:

- $V(C(G)) :=$ komponenty G ,
- $C_1, C_2 \in E(C(G)) \iff \exists v_1 \in C_1, v_2 \in C_2 : v_1 v_2 \in E(G)$.

Algorithm 20: Rozklad grafu na komponenty silné souvislosti: $\Theta(n + m)$

```

1 Sestrojíme  $G^T$                                 ▷ otočení orientací všech hran, lze díky  $C(G^T) \cong C(G)^T$ 
2  $Z \leftarrow$  prázdný zásobník
3 Opakované DFS na  $G^T$ , při opouštění vrcholu přidáme do  $Z$ 
4 Pole  $\forall v : \text{komp}(v) \leftarrow \emptyset$                 ▷ komp je pole komponent
5 for  $v \in Z$  do
6   odebíráme  $v$  ze zásobníku  $Z$ 
7   if  $\text{komp}(v) = \emptyset$  then
8     DFS( $v$ ) v  $G$ , nechodíme do vrcholů s  $\text{komp} \neq \emptyset$ , navštíveným vrcholům nastavíme
       $\text{komp}(-) \leftarrow v$ .
9   end
10 end
```

Toky v sítích.

Definice (Průtok). Průtok $f^* : E \rightarrow \mathbb{R}$ definujeme pro tok f jako: $f^*(uv) = f(uv) - f(vu)$.

Definice (Síť rezerv). Síť rezerv k toku f v síti $S = (V, E, z, s, c)$ je síť $R(S, f) := (V, E, z, s, r)$, kde $r(e)$ je rezerva hrany e při toku f .

Definice (Blokující tok). Pokud na každé orientované zs -cestě P , $\exists e \in P : f(e) = c(e)$.

Definice (Vrstevnatá síť). Síť je vrstevnatá (pročištěná), pokud všechny její vrcholy a hrany leží na nejkratších cestách ze z do s .

Dinitzův algoritmus.

Začne s nulovým tokem a bude ho vylepšovat pomocí nějakých pomocných toků v síti rezerv, až se dostane k maximálnímu toku. Počet potřebných iterací přitom bude záviset na tom, jak vydatné pomocné toky seženeme – na jednu stranu bychom chtěli, aby byly podobné maximálnímu toku, na druhou stranu jejich výpočtem nechceme trávit příliš mnoho času. Vhodným kompromisem jsou blokující toky.

Algorithm 21: Dinitzův algoritmus: $\mathcal{O}(n^2m)$

Data: Síť (V, E, z, s, c)
Result: Maximální tok f

```
1  $f \leftarrow$  nulový tok
2 while existuje cesta  $\ell$  do
3    $R \leftarrow$  síť rezerv, smažeme z  $f$  hrany s nulovou rezervou.
4    $\ell \leftarrow$  délka nejkratší cesty ze  $z$  do  $s$  v  $R$  ▷ BFS
5   if žádná taková cesta neexistuje then
6     | zastavíme se
7   end
8   Pročistíme síť  $R$ .
9    $g \leftarrow$  blokující tok v  $R$ 
10  Zlepšíme tok  $f$  pomocí  $g$ .
11 end
12 return tok  $f$ .
```

Algorithm 22: Blokující Tok: $\mathcal{O}(nm)$

Data: Vrstevnatá síť R s rezervami r
Result: Blokující tok g

```
1  $g \leftarrow$  nulový tok
2 while existuje v  $R$  orientovaná cesta  $P$  ze  $z$  do  $s$  do
3    $\varepsilon \leftarrow \min_{e \in P} (r(e) - g(e))$ 
4   forall  $e \in P$  do
5     |  $g(e) \leftarrow g(e) + \varepsilon$ 
6     | if  $g(e) = r(e)$  then
7       | smažeme  $e$  z  $R$ .
8     | end
9   Dočistíme síť pomocí fronty.
10 end
11 end
12 return tok  $g$ .
```

Lemma (O korektnosti). *Pokud se algoritmus zastaví, vydá maximální tok.*

Lemma. *V každém průchodu Dinicova algoritmu vzroste l alespoň o 1.*

Goldbergův algoritmus

Hledá maximální toku v síti. Je jednodušší než Dinicův algoritmus a má po pár snadných úpravách stejnou, nebo dokonce lepší časovou složitost.

Goldbergův algoritmus začne s ohodnocením hran, které ani nemusí být tokem, a postupně ho upravuje a zmenšuje, až se z něj stane tok, a to dokonce tok maximální.

Definice (Vlna). *Funkce $f : E \rightarrow \mathbb{R}_0^+$ je vlna v síti (V, E, z, s, c) , splňuje-li obě následující podmínky:*

- $\forall e \in E : f(e) \leq c(e)$ (vlna nepřekročí kapacity hran),
- $\forall v \in V \setminus \{z, s\} : f^\Delta(v) \geq 0$ (přebytek ve vrcholech je nezáporný).

Každý tok je tedy vlnou, ale opačně tomu tak být nemusí. V průběhu výpočtu se tedy potřebujeme postupně zbavit nenulových přebytků ve všech vrcholech kromě zdroje a spotřebiče. K tomu bude sloužit následující operace:

Definice (Převedení přebytku). *Převedení přebytku po hraně uv můžeme provést, pokud $f^\Delta(u) > 0$ a $r(uv) > 0$. Proběhne tak, že po hraně uv pošleme $\delta = \min(f^\Delta(u), r(uv))$ jednotek toku, podobně jako v předchozích algoritmech buď přičtením po směru nebo odečtením proti směru.*

Rádi bychom postupným převáděním všechny přebytky přepravili do spotřebiče, nebo je naopak přelili zpět do zdroje. Chceme se ovšem vyhnout přelévání přebytků tam a zase zpět, takže vrcholům přiřadíme výšky — to budou nějaká přirozená čísla $h(v)$.

Přebytek pak budeme ochotni převádět pouze z vyššího vrcholu do nižšího. Pokud se stane, že nalezneme vrchol s přebytkem, ze kterého nevede žádná nenasycená hrana směrem dolů, budeme tento vrchol zvedat — tedy zvyšovat mu výšku po jedné, než se dostane dostatečně vysoko, aby z něj přebytek mohl odtéci.

Algorithm 23: Goldbergův algoritmus: $\mathcal{O}(n^2m)$

Data: Síť (V, E, z, s, c)

Result: Maximální tok f

```

1  Nastavíme počáteční výšky:                                ▷ zdroj ve výšce  $n$ , ostatní ve výšce 0
2       $h(z) \leftarrow n$ 
3       $h(v) \leftarrow 0$  pro všechny  $v \neq z$ 
4  Vytvoříme počáteční vlnu:                                ▷ všechny hrany ze  $z$  na maximum
5       $f \leftarrow$  všude nulová funkce
6       $f(zv) \leftarrow c(zv)$ , kdykoliv  $zv \in E$ 
7  while existuje vrchol  $u \notin \{z, s\}$  takový, že  $f^\Delta(u) > 0$  do
8      |   if existuje hrana  $uv$  s  $r(uv) > 0$  a  $h(u) > h(v)$  then
9      |   |   převedeme přebytek po hraně  $uv$ .
10     |   else
11     |   |    $h(u) \leftarrow h(u) + 1$ .                                ▷ zvedneme  $u$ 
12     |   end
13 end
14 return Maximální tok  $f$ .
```

Toky v celočíselně ohodnocených grafech

Mějme nějaký bipartitní graf (V, E) . Přetvoříme ho na síť (V', E', z, s, c) následovně:

- Nalezneme partity grafu, *levou* a *pravou*.
- Všechny hrany zorientujeme zleva doprava.
- Přidáme zdroj z a vedeme z něj hrany do všech vrcholů levé partity.
- Přidáme stok s a vedeme do něj hrany ze všech vrcholů pravé partity.
- Všem hranám nastavíme jednotkovou kapacitu.

Najdeme maximální celočíselný tok a jelikož $\forall e : c(e) = 1$, musí po každé hraně téci buď 0 nebo 1. Do výsledného párování vložíme právě ty hrany původního grafu, po kterých teče 1.

3.3.3 Algoritmy vyhledávání v textu

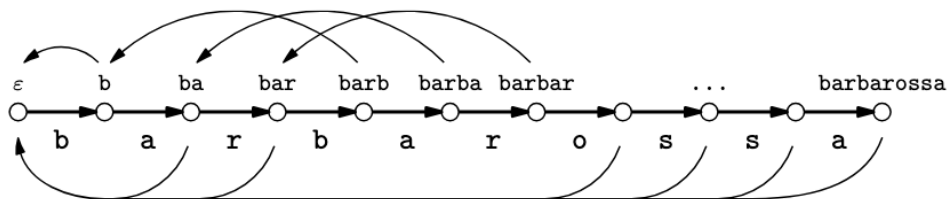
Definice. Necht J je délka jehly, S délka sena. Dále budeme pracovat s následujícími pojmy:

- **Podslovo** $\alpha[i : j] \equiv \alpha[i]\alpha[i+1]\dots\alpha[j-1]$.
- **Prefix** $\alpha[:j] \equiv \alpha[0:j]$, **Suffix** $\alpha[i:] \equiv \alpha[i:|\alpha|]$.
- **Stav algoritmu:** Nejdelší prefix jehly, který je suffixem sena.
- **Dopředná funkce:** rozšíření prefixu $s \rightarrow s+1$ přidáním jednoho symbolu $\iota[s]$.
- **Zpětná funkce** $Z(s)$ přiřadí α jeho nejdelší suffix různý od α , který je prefixem jehly.

Knuth-Morris-Prat. $\Theta(J + S)$

Aalgoritmus pro hledání jednoho vzoru v textu. Stavy očíslovíme $0, \dots, J$.

- Nejprve spočítáme prefix funkci (pole nejdelších vlastních prefixů, které jsou zároveň suffixy).
- Při prohledávání textu využíváme prefixovou funkci k tomu, abychom se při neúspěchu nemuseli vracet, ale abychom přeskočili části textu, kterou už jsme porovnali.



Algorithm 24: $\text{KmpKrok}(s, x)$ – jsme ve stavu s , přečetli jsme znak x .

```

1 while  $\iota[s] \neq x \wedge s \neq 0$  do  $s \leftarrow Z[s]$ . ▷ zpětné hrany
2 if  $\iota[s] = x$  then  $s \leftarrow s + 1$ . ▷ dopředná hrana
Result: Nový stav  $s$ .

```

Algorithm 25: $\text{KmpHledej}(\sigma)$

```

1  $s \leftarrow 0$ 
2 for znaky  $x \in \sigma$  do
3    $s \leftarrow \text{KmpKrok}(s, x)$ 
4   if  $s = J$  then Ohlásíme výskyt.
5 end

```

Algorithm 26: KmpKonstrukce(ι) – jehla ι délky J . (konstrukce automatu)

```
1  $Z[0] \leftarrow \emptyset$   $Z[1] \leftarrow 0$ 
2  $s \leftarrow 0$ 
3 for  $i = 2, \dots, J$  do
4    $s \leftarrow \text{KmpKrok}(s, \iota[i-1])$ 
5    $Z[i] \leftarrow s$ 
6 end
Result: Pole zpětných hran  $Z$ .
```

Aho-Corasick. $\Theta(J + S + V)$

Hledání více řetězců najednou v textu. V je počet výskytů v seně S .

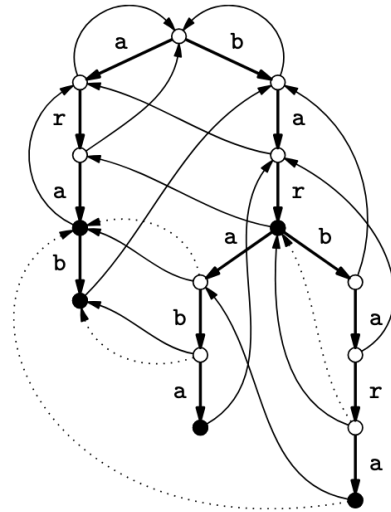
Máme $\iota_1, \dots, \iota_N$ s délkami J_i a seno σ délky S .

Definice (Zkratková hrana). Řekne, jaký je nejdelší vlastní suffix slova α , který je jehlou.

- $Zpet(s)$ – číslo stavu, kam vede zpětná hrana (nebo $\emptyset$, pokud žádná nevede),
- $Zkratka(s)$ – kam vede zkratková hrana (obdobně),
- $Slovo(s)$ – zda tu končí nějaké slovo (a pokud ano, tak které),
- $Dopředu(s, x)$ – kam vede dopředná hrana označená písmenem x .

Algoritmus probíhá následovně:

- (1) Nejprve vytvoříme trie (prefixový strom) všech vzorů.
- (2) Pak přidáme tzv. *fail* odkazy mezi uzly pro přechod při neúspěchu.
- (3) Nakonec procházíme text znak po znaku a sledujeme přechody v automatu, přičemž při každém kroku víme, zda právě skončil některý z hledaných vzorů.



Algorithm 27: AcKrok(s, x) – jsme ve stavu s , přečetli jsme znak x

```
1 while  $\text{Dopředu}(s, x) = 0 \wedge s \neq \text{koren}$  do
2    $s \leftarrow \text{Zpet}(s)$ 
3 end
4 if  $\text{Dopředu}(s, x) \neq 0$  then  $s \leftarrow \text{Dopředu}(s, x)$ 
Result: Nový stav  $s$ .
```

Algorithm 28: AcHledej(σ) – seno σ a zkonstruovaný automat

```
1  $s \leftarrow$  koren
2 for znaky  $x \in \sigma$  do
3    $s \leftarrow$  AcKrok( $s, x$ )
4    $q \leftarrow s$ 
5   while  $q \neq 0$  do
6     if Slovo( $q$ )  $\neq 0$  then
7       | Ohlasime Slovo( $q$ )
8     end
9      $q \leftarrow$  Zkratka( $q$ )
10  end
11 end
```

Algorithm 29: AcKonstrukce($\iota_1 \dots \iota_n$)

```
1 založime strom s korenem  $r$ 
2 vkladame do stromu slovo  $\iota_1 \dots \iota_n \rightsquigarrow$  Dopredu( $-$ ,  $-$ ),  $\rightsquigarrow$  Slovo( $-$ )
3 Zpet( $r$ )  $\leftarrow 0$ 
4  $F \leftarrow$  fronta, vložíme do ni syny korene
5 synum korene nastavíme zpetnou hranu na koren, neboli Zpet( $-$ )  $\leftarrow r$ 
6 while  $F \neq 0$  do
7   vybereme z fronry vrchol  $\rightarrow i$ 
8   for syny  $s$  vrcholu  $i$  do
9      $z \leftarrow$  AcKrok(Zpet( $i$ )), znak na hrane  $\{is\}$ 
10    Zpet( $s$ )  $\leftarrow z$ 
11    if Slovo( $z$ )  $\neq 0$  then
12      | Zkratka( $s$ )  $\leftarrow z$ 
13    else
14      | Zkratka( $s$ )  $\leftarrow$  Zkratka( $z$ )
15    end
16    Vložíme  $s$  do  $F$ 
17  end
18 end
```

3.3.4 Algebraické algoritmy

Poznámka. Pro připomenutí. Pracujeme s různými tvary komplexních čísel:

- **Algebraický tvar:** $z = a + bi$, $a, b \in \mathbb{R}$, dále $\bar{z} = a - bi$.
- **Goniometrický tvar:** $z = |z| (\cos \varphi + i \sin \varphi)$.
- **Exponenciální tvar:** Eulerova formule $e^{i\varphi} = \cos \varphi + i \sin \varphi$.

Definice (Primitivní n -tá odmocnina z 1). Číslo $\omega \in \mathbb{C}$ je primitivní n -tá odmocnina z 1, pokud $\omega^n = 1$ a žádné z čísel $\omega^1, \omega^2, \dots, \omega^{n-1}$ není rovno 1.

Poznámka. Pro sudé n je $\omega^{n/2} = -1$.

Platí totiž $(\omega^{n/2})^2 = \omega^n = 1$, takže $\omega^{n/2}$ je druhá odmocnina z 1. Takové odmocniny jsou jenom dvě: 1 a -1 , ovšem 1 to být nemůže, protože ω je primitivní.

Definice (Diskrétní Fourierova transformace (DFT)). *Diskrétní Fourierova transformace je zobrazení $\mathcal{F} : \mathbb{C}^n \rightarrow \mathbb{C}^n$, které vektoru $\mathbf{x} = (x_0, \dots, x_{n-1})$ přiřadí vektor $\mathbf{y} = (y_0, \dots, y_{n-1})$ daný předpisem*

$$y_j = \sum_{k=0}^{n-1} x_k \cdot \omega^{jk},$$

kde ω je pevně zvolená primitivní n -tá odmocnina z jedné. Vektor $\mathbf{y}$ se nazývá Fourierův obraz vektoru $\mathbf{x}$.

Algorithm 30: Algoritmus FFT – Rychlá Fourierova Transformace $\Theta(n \log n)$

Data: Číslo $n = 2^k$, primitivní n -tá odmocnina z jedničky ω a vektor $(p_0, \dots, p_{n-1})$ koeficientů polynomu P .

Result: Graf polynomu P , tedy vektor $(y_0, \dots, y_{n-1})$, kde $y_j = P(\omega^j)$.

```

1 if  $n = 1$  then
2    $y_0 \leftarrow p_0$ 
3   return  $(y_0)$ 
4 else
5    $(s_0, \dots, s_{n/2-1}) \leftarrow \text{FFT}(n/2, \omega^2, (p_0, p_2, p_4, \dots, p_{n-2}))$ 
6    $(\ell_0, \dots, \ell_{n/2-1}) \leftarrow \text{FFT}(n/2, \omega^2, (p_1, p_3, p_5, \dots, p_{n-1}))$ 
7   for  $j = 0, \dots, n/2 - 1$  do
8      $y_j \leftarrow s_j + \omega^j \cdot \ell_j$ 
9      $y_{j+n/2} \leftarrow s_j - \omega^j \cdot \ell_j$ 
10  end
11  return  $(y_0, \dots, y_{n-1})$ 
12 end
```

Definice (Fourierova Transformace). *Fourierova transformace vektoru $\vec{v} \in \mathbb{C}^n$ je dána maticovým násobením:*

$$\mathcal{F}(\vec{v})_k = (\Omega \cdot \vec{v})_k = \sum_{j=0}^{n-1} \vec{v}_j \cdot \omega^{jk}$$

$$\text{kde } \Omega = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^1 & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{pmatrix} \text{ je Fourierova matice.}$$

Definice (Inverzní Fourierova Transformace IFFT). *Inverzní Fourierova transformace je:*

$$\mathcal{F}^{-1}(\vec{v})_k = \frac{1}{n} \sum_{j=0}^{n-1} \vec{v}_j \cdot \omega^{-jk},$$

což odpovídá maticovému násobení s komplexně sdruženou maticí $\bar{\Omega}$:

$$\bar{\Omega} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \dots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \dots & \omega^{-(n-1)^2} \end{pmatrix}$$

Věta (Vlastnosti Fourierovy matice). *Fourierova matice splňuje $\Omega \cdot \bar{\Omega} = n \cdot I_n$, kde I_n je jednotková matice řádu n .*

Poznámka (Výpočet IFFT). *Pro praktický výpočet IFFT použijeme stejný algoritmus FFT, ale:*

- Místo ω použijeme ω^{-1} (nebo ekvivalentně $\bar{\omega}$).
- Výsledek nakonec vydělíme n .

3.3.5 RSA

Notace: Zdefinujeme si:

$p, q \in \mathbb{N}$		velká prvočísla, t.ž.: $p \neq q$
(N, e)		dvojice, veřejný klíč, kde $N = p \cdot q$
$\varphi(N) = (p-1)(q-1)$		Eulerova funkce
$e \in \mathbb{N}, 0 < e < \varphi(N)$		šifrovací exponent
$d \in \mathbb{N}$		dešifrovací exponent

Zároveň musí platit $\gcd(e, \varphi(N)) = 1$ a dále se hodí k výpočtům následující vztahy:

$y = x^e \pmod{N}$		zašifrování plaintextu, výsledkem je ciphertext
$x = y^d \pmod{N}$		dešifrování ciphertextu, výsledkem je plaintext
$d \cdot e \equiv 1 \pmod{\varphi(N)}$	..	získání d (Euklidovým algoritmem)

Dešifrování se dá lehce odvodit: $y^d \equiv x^{e \cdot d} \equiv x^1 \equiv x \pmod{N}$.

Popis algoritmu: Bob si vygeneruje náhodná p, q a vypočítá z nich $N = p \cdot q$. Dále vypočítá Eulerovu funkci $\varphi(N)$ a následně vygeneruje číslo $e \in \mathbb{N}$, pro které platí $\gcd(e, \varphi(N)) = 1$. Tímto číslem zašifruje plaintext x vztahem $y = x^e \pmod{N}$. Pak už jen nalezneme číslo $d \in \mathbb{N}$ euklidovým algoritmem $d \cdot e \equiv 1 \pmod{\varphi(N)}$. Veřejný klíč, dvojici (N, e) , pošle Alici spolu s ciphertextem y .

Alice přijme veřejný klíč (N, e) a ciphertext y . Pouze Alici je znám soukromý klíč (N, d) , využije ho k dešifrování y . To udělá vztahem $x = y^d \pmod{N}$.

Eva nemá možnost si zprávu přečíst, protože nezná dešifrovací exponent d . Musela by ho uhádnout, což není pravděpodobné, nebo by musela znát prvočísla p, q . Kdyby znala p, q mohla by si jednoduše dopočítat $\varphi(N)$ a následně d tak, jak jsme to udělali my. Bezpečnost RSA tedy stojí na tom, že útočník není schopen rozložit $N = p \cdot q$ na p, q , proto je potřeba je volit dostatečně velká.

3.3.6 Aproximační algoritmy

Máme nějakou množinu přípustných řešení a každé z nich ohodnotíme cenou $c(x)$. Mezi nimi hledáme optimální řešení s minimální cenou c^* . Vystačíme si s α -aproximací – přípustné řešení má cenu $c' \leq \alpha c^*$ pro $\alpha > 1$.

Definice (Poměrová chyba). *Poměrová chyba je poměr mezi výstupem a optimem je nejvýše α .*

Definice (Relativní chyba). *Relativní chyba nepřekročí $(c' - c^*)/c^*$.*

Aproximační schémata.

Definice (PTAS - Polynomial-Time Approximation Scheme). *Algoritmus je PTAS, pokud $\forall \epsilon > 0$ najde v čase polynomiálním v n řešení s relativní chybou nejvýše ϵ . Formálně:*

$$\frac{|\text{ALG} - \text{OPT}|}{\text{OPT}} \leq \epsilon,$$

kde OPT je optimální řešení a ALG je řešení algoritmu.

Definice (FPTAS - Fully Polynomial-Time Approximation Scheme). *Algoritmus je FPTAS, pokud je PTAS a navíc jeho časová složitost je polynomiální jak v n , tak v $\frac{1}{\epsilon}$. Formálně:*

$$\text{Časová složitost} = \mathcal{O}\left(p\left(n, \frac{1}{\epsilon}\right)\right)$$

kde p je nějaký polynom.

Příklad. Problém batohu má FPTAS. Problém TSP v obecných grafech má PTAS, ale ne FPTAS.

Poznámka. FPTAS je silnější požadavek než PTAS, protože vyžaduje polynomiální závislost na $1/\epsilon$, ne jen na n .

Problém obchodního cestujícího (TSP).

Definice (Problém TSP). Mějme neorientovaný graf G s nezáporně ohodnocenými hranami $d(e) \geq 0$. Cílem je najít nejkratší hamiltonovskou kružnici v G . Předpokládáme, že graf je úplný a splňuje trojúhelníkovou nerovnost:

$$\forall x, y, z \in V : d(x, y) + d(y, z) \geq d(x, z)$$

Algorithm 31: TSP – 2-aproximace

- 1 Najdi minimální kostru T grafu G .
 - 2 Projdi kostru Eulerovským sledem (délka $2T$).
 - 3 Odstraň duplicity ve vrcholech pomocí “zkratek” (využitím trojúhelníkové nerovnosti).
-

Poznámka. Výsledná délka kružnice je $\leq 2T \leq 2\text{OPT}$, kde OPT je délka optimálního řešení.

Věta. Bez trojúhelníkové nerovnosti nelze TSP aproximovat (pokud $P \neq NP$).

Problém batohu.

Definice (Problém batohu). Mějme batoh s kapacitou H a množinu předmětů $1, \dots, n$ s:

- Hmotnostmi $h_1, \dots, h_n$,
- Cenami $c_1, \dots, c_n$.

Cílem je najít podmnožinu $P \subseteq \{1, \dots, n\}$ takovou, že:

$$h(P) = \sum_{i \in P} h_i \leq H \quad \text{a} \quad c(P) = \max \sum_{i \in P} c_i.$$

Algorithm 32: Aproximace problému batohu

- 1 Odstraň všechny předměty s $h_i > H$.
 - 2 Spočítej $c_{\max} = \max_i c_i$ a zvol $M = \lceil \frac{n}{\epsilon} \rceil$ pro libovolné $\epsilon > 0$.⁵
 - 3 Kvantujeme ceny: pro $i = 1, \dots, n$ nechť $\hat{c}_i = \lfloor c_i \cdot \frac{M}{c_{\max}} \rfloor$.
 - 4 Vyřešíme upravený problém batohu s $\hat{c}_1, \dots, \hat{c}_n$ dynamickým programováním.
 - 5 Vratíme předměty z optimálního řešení kvantovaného problému.
-

Poznámka. Zaokrouhlování hmotností by mohlo vést k nepřijatelným řešením, ale zaokrouhlování cen je bezpečné.

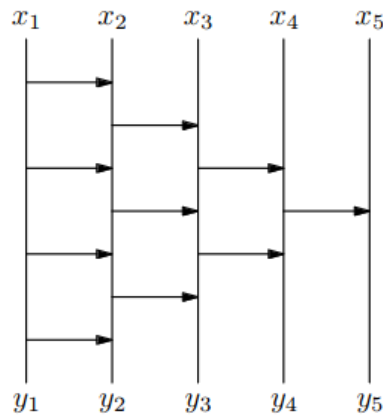
Věta. Algoritmus poskytuje $(1 - \epsilon)$ -aproximaci v čase $O(n^3/\epsilon)$.

⁵Zvolíme přirozené $M < c_{\max}$ a zobrazíme $[0, c_{\max}] \rightarrow \{0, \dots, M\}$. Tedy každou cenu c_i znásobíme poměrem $\frac{M}{c_{\max}}$. Každé c_i jsme změnil nejvýše o $\frac{c_{\max}}{M}$, proto celkovou cenu nejvýše o $n \cdot \frac{c_{\max}}{M}$. Po odstranění všech $h_i > H$ platí $c^* > c_{\max}$, proto chyba aproximace $\leq n \cdot \frac{c^*}{M}$. Chybu omezíme shora $\epsilon > 0$ jako $\epsilon \cdot c^*$ a musíme tak zvolit $M \geq \frac{n}{\epsilon}$.

3.3.7 Paralelní třídění komparátorových sítí

Definice (Komparátorová síť). Komparátorová síť je hradlová síť složená z komparátorů. Na vstupu má 2 čísla a výstupem je vlevo menší číslo, vpravo větší číslo.

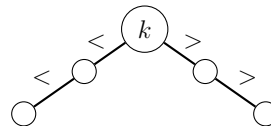
Příklad (Bubble sort $\mathcal{O}(n)$). Příkladem je Bubble sort, který jsme schopni udělat lineární:



Bitonické třídění.

Definice (Čistá bitonická posloupnost). Posloupnost $x_0, \dots, x_{n-1}$ je čistě bitonická, pokud existuje index k takový, že:

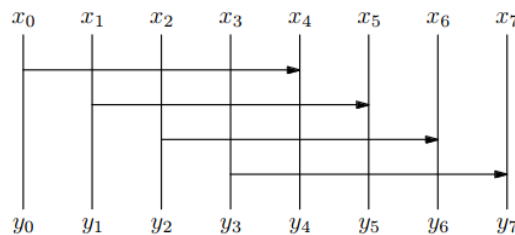
- $x_0 \leq x_1 \leq \dots \leq x_k$ (rostoucí část)
- $x_k \geq x_{k+1} \geq \dots \geq x_{n-1}$ (klesající část)



Definice (Bitonická posloupnost). Posloupnost je bitonická, pokud je cyklickou rotací nějaké čistě bitonické posloupnosti.

Definice (Separátor). Separátor je obvod, který:

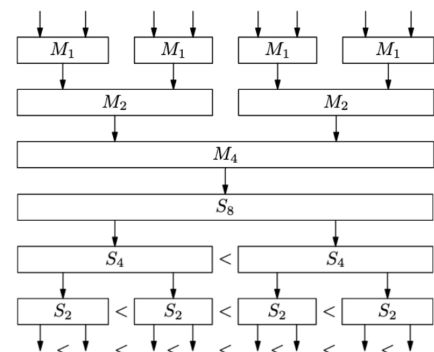
- Rozdělí bitonickou posloupnost délky n na dvě poloviční posloupnosti.
- Zajistí, že všechny prvky v první polovině jsou menší než všechny prvky v druhé polovině.



Vlastnosti. Platí následující vlastnosti:

- Hloubka obvodu: 1 (jedna vrstva komparátorů)
- Počet hradel: $\frac{n}{2}$ ($\forall$ komparátor 2 vstupy a 2 výstupy)
- Časová složitost: $\mathcal{O}(1)$

Poznámka. Za pomoci bitonických “třídiček” a “slévaček” jsme takto schopni vytvořit jednoduše Merge sort. (viz. Obr.)



3.4 Pokročilá diskretní matematika

3.4.1 Barvení grafů

Definice (Vrcholová barevnost). Vrcholová barevnost $\chi(G)$ grafu G je nejmenší počet barev, kterými lze obarvit vrcholy G . (Tedy žádné dva sousední vrcholy nemají stejnou barvu).

Poznámka. Pro graf G značíme:

- Maximální stupeň jako $\Delta(G)$,
- Minimální stupeň jako $\delta(G)$,
- Velikost největší kliky jako $\omega(G)$,
- Velikost největší nezávislé množiny jako $\alpha(G)$.

Věta. Pro graf G platí vztah: $\chi(G) \geq \frac{|V(G)|}{\alpha(G)}$.

Definice (Degenerovanost). Graf G je d -degenerovaný $\equiv$ každý podgraf H grafu G má $\delta(H) \leq d$.

Poznámka. Pokud G je $\Delta(G)$ -degenerovaný, pak $\chi(G) \leq \Delta(G) + 1$.

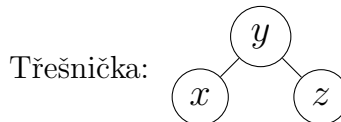
Definice (Hranové barvení). Hranové barvení je funkce $b : E(G) \rightarrow B$ (barvy) taková, že pokud $e \neq f \in E$ a e, f mají společný vrchol, pak $b(e) \neq b(f)$.

Definice (Hranová barevnost). Hranová barevnost G (“chromatický index”) $\chi'(G)$ je minimální počet barev pro hranové barvení G . Platí $\chi'(G) \leq \chi(G)$.

Věta (Brooks). Nechť G je souvislý graf který není úplný a není lichá kružnice. Pak

$$\chi(G) \leq \Delta(G).$$

Důkaz. Indukcí na počtu vrcholů: pokud má graf vrchol stupně $< \Delta$, odebereme ho a zpětně snadno dobarvíme. Pokud je Δ -regulární, najdeme “třešničku”:



Odebereme z , dobarvíme $G - z$ Δ barvami. Pokud x a y mají různé barvy, tak lze z dobarvit rovnou. Pokud mají stejnou barvu, barvy přeházíme tak, aby se uvolnila barva pro z . Výjimkou, kdy to selže, je jen úplný graf nebo lichá kružnice. $\square$

Věta (Vizing). Pro každý graf G platí, že

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1.$$

Důkaz. Postupujeme hladově – každou hranu barvíme postupně jednou z $1, \dots, \Delta + 1$ barev. Dále rozdělíme na případy; hledáme *volné* a *použitelné* vrcholy. Pokud u koncových vrcholů není volná barva, tak pomocí přehazování barev na alternativních řetězcích vždy uvolníme vhodnou barvu. $\square$

Poznámka. Platí, že $\omega(G) \leq \chi(G)$. A také $\omega(\bar{G}) = \alpha(G)$.

Definice (Perfektní graf). Graf G je perfektní, pokud $\forall H \leq G : \chi(H) = \omega(H)$.

Věta (Slabá o perfektních grafech). Graf G je perfektní $\iff \bar{G}$ je perfektní.

Věta (Silná o perfektních grafech). Graf G je perfektní $\iff G$ ani $\bar{G}$ neobsahuje jako indukovaný podgraf lichý cyklus délky ≥ 5 .

Poznámka. Perfektní grafy jsou například K_n nebo bipartitní grafy.

Definice (Chordální graf). Graf je chordální, pokud neobsahuje $C_k, k \geq 4$ jako indukovaný podgraf. (každá kružnice má chordu (tětivu))

3.4.2 Párování v grafech

Definice (Párování). Párování je množina hran M taková, že každý vrchol je incidentní s nejvýše jednou hranou v M . Tedy $\forall e \neq e' \in M : e \cap e' = \emptyset$.

Definice (Největší párování). Největší párování je párování největší velikosti. Velikost je nejvýše $\frac{n}{2}$.

Definice (Perfektní párování). Perfektní párování je párování, kde každý vrchol je incidentní s právě jednou hranou. (Neboli pokud v párování neexistuje žádný volný vrchol. $|M| = \frac{|V|}{2}$.)

Poznámka. Grafy s lichým počtem vrcholů nemají perfektní párování.

Tutteho a Petersenova věta.

Definice (Tutteho podmínka). Platí $\forall S \subseteq V : \text{odd}(G - S) \leq |S|$, kde odd značí počet lichých komponent grafu.

Věta (Tutte). Graf G má perfektní párování $\iff$ platí Tutteova podmínka.

Proof. Obměnou " $\Rightarrow$ ": Říkáme, že pokud neplatí Tutteho podmínka, pak nemá perfektní párování.

Nechť $\exists S \subseteq V$ t. ž. $\text{odd}(G - S) > |S|$. V perfektním párování se alespoň 1 vrchol z každé liché komponenty musí spárovat s nějakým z S , ale těch není dostatek. $\square$

Věta (Petersen). Každý 3-regulární 2-souvislý⁶ graf má perfektní párování.

Důkaz. Vezměme si libovolnou $S \subseteq V$. Ukážeme, že splňuje Tutteho podmínku.

Pokud $\text{odd}(G - S) = 0$, jsme hotovi.

Předpokládejme, tedy že v $G - S$ existuje lichá komponenta K . Potom platí:

$$\sum_{v \in V(K)} \deg_G(v) = 3|V(K)| = 2|E(K)| + \text{počet hran vycházející mimo } K, \text{ což je liché číslo.}$$

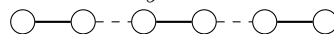
Počet hran mezi K a S musí být alespoň 2, jinak máme most, je to tedy alespoň 3.

Maximální počet hran z S je $3|S|$, tudíž lichých komponent může být nejvýše $\frac{3|S|}{3}$, tedy $|S|$. $\square$

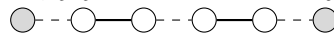
Edmondsův kytičkový algoritmus.

Definice (Volný vrchol). Volný vrchol je vrchol, který nevidí žádnou hranu párování.

Definice (Střídavá cesta). Střídavá cesta je cesta, kde jsou všechny vnitřní vrcholy incidentní s párovací hranou a navíc tato hrana leží na cestě.

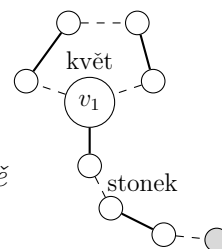


Definice (Volná střídavá cesta). Volná střídavá cesta je taková střídavá cesta, jejíž koncové vrcholy jsou volné.



Definice (Kytka). Kytka je tvořena stonkem a květem:

- stonk je střídavá cesta z v_1 (i nulové) délky končící volným vrcholem.
- květ je lichá "střídavá" kružnice s vrcholem v_1 , ke kterému přiléhají dvě hrany $\notin M$.



Algoritmus. (Edmondsův). Na vstupu párování M a zlepší jej nebo jej prohlásí za největší.

- Hledáme střídavé cesty, které zvyšují velikost M .
- Pokud narazíme na kytku, zkontrahuje ji do jednoho supervrcholu.
- Kytka se rekurzivně zpracuje a po nalezení párování ve sbaleném grafu se rozbalí zpět do původní struktury.

⁶Vrcholově i hranově, pro 3-regulární grafy je to to samé; alternativně můžeme říct graf bez mostů a artikulací.

3.4.3 Kreslení grafů na plochách.

Definice (Homeomorfismus). Necht' $X \subseteq \mathbb{R}^n, Y \subseteq \mathbb{R}^m$. Potom homeomorfismus z X na Y je funkce $f : X \mapsto Y$, která je spojitá, bijekce a f^{-1} je spojitá.

Říkáme, že X, Y jsou homeomorfní, $X \cong Y$, pokud mezi nimi existuje homeomorfismus.

Definice (Plocha). Plocha je kompaktní (uzavřená, omezená), souvislá, 2-rozměrná varieta bez hranice (dostatečně malé okolí každého bodu je homeomorfní otevřenému okolí v $\mathbb{R}^2$).

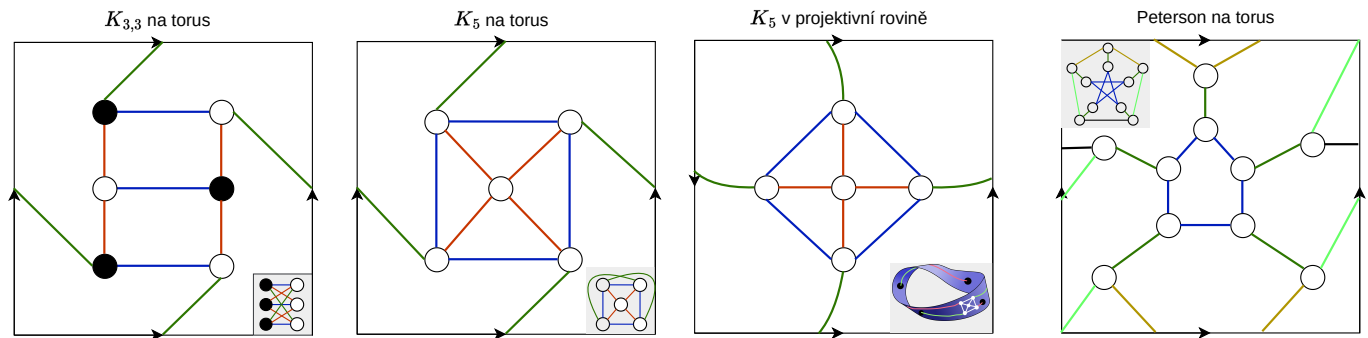
Konstrukce ploch.

Přidání ucha. “Teleport, do kterého když vejde, tak na druhé straně vyjde opačně (otočeně).” Vyřízneme dva kruhy, vezmeme plášť válce bez dna a vrchu, ohneme a přilepíme na díry po kruzích.

Přidání křížítka. “Teleport, do kterého když vejde, tak nás to přesune naproti.”

Definice (Orientovaná plocha). Pro $g \in \{0, 1, \dots\}$ necht' $\sum_g$ značí plochu vzniklou ze sféry přidáním g uší, tak říkáme, že $\sum_g$ je orientovatelná plocha rodu g .

Definice (Neorientovaná plocha). Pro $g \in \{1, 2, \dots\}$ necht' $\prod_g$ značí plochu vzniklou ze sféry přidáním g křížíttek, tak říkáme, že $\prod_g$ je neorientovatelná plocha rodu g .



Eulerova charakteristika.

Definice (Nakreslení grafu). Nakreslení grafu $G = (V, E)$ na plochu Γ je zobrazení φ t. ž.:

- každému vrcholu $v \in V$ přiřadí bod $\varphi(v) \in \Gamma$
- každé hraně $e \in E$ přiřadí prostou (neprotínající se) křivku $\varphi(e) \in \Gamma$ spojující konce $\varphi(x), \varphi(y)$
- vrcholy se nepřekrývají: $x, y \in V : x \neq y \implies \varphi(x) \neq \varphi(y)$
- hrany se překrývají nejvýše ve sdílených vrcholech: $e, f \in E : e \neq f \implies \varphi(e) \cap \varphi(f) = \{\varphi(x) \mid x \in e \cap f\}$
- vrcholy, které neleží na hraně se s ní neprotínají: $e \in E, x \in V : x \notin e \implies \varphi(x) \notin \varphi(e)$

Definice (Stěna nakreslení). Souvislá komponenta $\Gamma \setminus \left(\left(\bigcup_{e \in E} \varphi(e) \right) \cup \left(\bigcup_{x \in V} \varphi(x) \right) \right)$.

Definice (Buňkové nakreslení). Každá stěna je homeomorfní otevřenému kruhu v $\mathbb{R}^2$.

Definice (Eulerova charakteristika). Eulerova charakteristika plochy Γ je

$$\begin{aligned} \chi(\Gamma) &= \begin{cases} 2 - g & \Gamma \cong \prod(g \geq 1) \\ 2 - 2g & \Gamma \cong \sum(g \geq 0) \end{cases} \\ &= 2 - \# \text{ křížíttek} - 2 \cdot \# \text{ uší.} \end{aligned}$$

Věta (Zobecněná Eulerova formule). *Nechť máme nakreslení grafu $G = (V, E)$ na ploše Γ , které má S stěn. Pak $|V| - |E| + |S| \geq \chi(\Gamma)$. Pokud je buňkové, tak dokonce $|V| - |E| + |S| = \chi(\Gamma)$.*

Důsledek. *Každý graf G nakreslitelný na plochu Γ splní $|E| \leq 3|V| - 3\chi(\Gamma)$, pokud $|V| \geq 4$.*

Věta. *Nechť Γ je plocha, $\Gamma \neq \Sigma_0$ a nechť G je graf nakreslený na Γ .*

Potom G obsahuje vrchol stupně $\delta(G) \leq \left\lfloor \frac{5 + \sqrt{49 - 24\chi(\Gamma)}}{2} \right\rfloor$.

Poznámka. *Platí vztah $\sum_{f \in S} \text{delka}(f) = 2|E| = \sum_{v \in V} \deg(v)$.*

3.4.4 Grafové minory

Definice (Minor). *Nechť H, G jsou grafy. Pak H je minor G (resp. G obsahuje H jako minor), značíme $H \preceq G$, pokud H lze získat z G posloupností mazání vrcholů, mazání hran nebo kontrakcí hran.*

Vlastnosti. Pro grafové minory platí následující vlastnosti:

- $\preceq$ je transitivní (prostě spojíme posloupnosti operací),
- H podgraf $G \implies H$ minor G ,
- G rovinný $\implies$ jeho minory jsou také rovinné.

Věta (Kuratowski). G rovinný $\iff$ neobsahuje dělení K_5 ani $K_{3,3}$ jako podgraf.

Věta (Kuratowski-Warner). G rovinný $\iff$ neobsahuje K_5 ani $K_{3,3}$ jako minor.

3.4.5 Množiny a zobrazení

Definice (Třída). *Pokud je $\varphi(x)$ formule, tak výraz $\{x \mid \varphi(x)\}$ nazýváme třídový term. Definuje “soubor” všech množin x , pro které platí $\varphi(x)$. Tomuto souboru říkáme třída určená formulí $\varphi(x)$.*

Poznámka. *Každá množina je zároveň i třídou, protože $x = \{z \mid z \in x\}$.*

Definice (Vlastní třída). *Třidu, která není množinou, nazýváme vlastní třída.*

Definice (Třídové operace). *Pro třídy A a B definujeme*

- $A \cap B = \{x \mid x \in A \wedge x \in B\},$
- $A \cup B = \{x \mid x \in A \vee x \in B\},$
- $A \setminus B = \{x \mid x \in A \wedge x \notin B\},$... případně $A - B$
- $\bigcup A = \{x \mid (\exists a)(a \in A \wedge x \in a)\},$
- $\bigcap A = \{x \mid (\forall a)(a \in A \rightarrow x \in a)\}$
- $A \subseteq B \equiv (\forall x)(x \in A \rightarrow x \in B),$... A je podtřídou B
- $A \subset B \equiv A \subseteq B \wedge A \neq B,$... A je vlastní podtřídou B
- $\mathcal{P}(A) = \{x \mid x \subseteq A\}.$

Definice (Kartézský součin). Kartézský součin tříd A a B je třída

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

Kartézský součin tříd $A_1, A_2, \dots, A_n$ definujeme induktivně jako

$$A_1 \times \dots \times A_n = (A_1 \times \dots \times A_{n-1}) \times A_n = \{(a_1, \dots, a_n) \mid \bigwedge_{i=1}^n a_i \in A_i\}.$$

Lemma. Jsou-li x a y množiny, potom je $x \times y$ také množina.

Definice (Binární relace). Binární relace je libovolná třída $R \subseteq V \times V$. Namísto $(x, y) \in R$ píšeme $x R y$. Podobně lze definovat n -ární relaci jako $R \subseteq V^n$.

Definice. Pro (binární) relaci R a třídu X definujeme třídy

- $R^{-1} = \{(v, u) \mid u R v\},$... inverzní relace k relaci R
- $\text{Dom}(R) = \{u \mid (\exists v)(u R v)\},$... definiční obor relace R
- $\text{Rng}(R) = \{v \mid (\exists u)(u R v)\},$... obor hodnot relace R
- $R \upharpoonright X = R \cap (X \times V) \subseteq R,$... zúžení relace R na třídu X
- $R[X] = \text{Rng}(R \upharpoonright X) \subseteq \text{Rng}(R).$... obraz třídy X relací R

Definice. Relace odpovídající relačním symbolům jazyka teorie množin jsou

- $E = \{(x, y) \mid x \in y\},$... náležení
- $\text{Id} = \{(x, y) \mid x = y\}.$... identita
- $\Delta_X = \text{Id} \upharpoonright X$... identita na třídě X

Definice (Skládání relací). Pro relace R a S definujeme relaci

$$R \circ S = \{(u, w) \mid (\exists v)(u R v \wedge v S w)\}$$

Definice (Zobrazení). Relace F je zobrazení (funkce) $\equiv$

$$(\forall u)(\forall v)(\forall w)((u, v) \in F \wedge (u, w) \in F \rightarrow v = w).$$

Namísto $u F v$ píšeme $F(u) = v$, při definování píšeme $F : u \mapsto v$.

Poznámka. F je zobrazení $\iff (\forall x \in \text{Dom}(F))(\exists! y \in \text{Rng}(F))(F(x) = y)$.

Definice (Prosté zobrazení, bijekce). Řekneme, že zobrazení F je

- prosté $\equiv F^{-1}$ je zobrazení, tedy $\forall x, y \in X : f(x) = f(y) \implies x = y,$... injekce
- zobrazení třídy X do třídy $Y \equiv \text{Dom}(F) = X \wedge \text{Rng}(F) \subseteq Y$, píšeme $F : X \rightarrow Y$,
- na $\equiv F : X \rightarrow Y \wedge \text{Rng}(F) = Y$, tedy $\forall y \in Y \exists x \in X : f(x) = y,$... surjekce
- bijekce mezi třídami X a $Y \equiv$ je to prosté zobrazení X na Y .

Definice (Suma). Pro každou množinu x existuje množina s , která je sjednocením všech množin uvnitř x . Této množině říkáme suma množiny x a značíme ji $\bigcup x$.

$$(\forall x)(\exists s) : (\forall z)(z \in s \iff (\exists y)(y \in x \wedge z \in y)).$$

Definice (Potenční množina). Pro každou množinu x existuje množina všech jejích podmnožin. Této množině říkáme potenční množina množiny x a značíme ji $\mathcal{P}(x)$.

$$(\forall x)(\exists p) : (\forall z)(z \in p \iff z \subseteq x).$$

3.4.6 Subvalence a ekvivalence množin

Definice (Ekvivalence). *Relace R je na třídě X ekvivalence $\equiv$ je reflexivní, symetrická a tranzitivní.*

Definice (Třídy ekvivalence). *Nechť R je ekvivalence na třídě X . Pro $x \in X$ definujeme třídu*

$$[x]_R = \{y \mid y \in X \wedge x R y\} = R[\{x\}] \cap X,$$

kterou nazýváme ekvivalenční třída prvku x .

Definice. *Pro množiny x a y definujeme relace*

1. $x \approx y \equiv$ existuje bijekce $f : x \rightarrow y$,
2. $x \preceq y \equiv$ existuje prosté zobrazení $f : x \rightarrow y$,
3. $x \prec y \equiv x \preceq y \wedge x \not\approx y$.

Věta (Cantor, Schröder, Bernstein). $x \approx y \iff (x \preceq y \wedge y \preceq x)$.

Konečné a spočetné množiny

Definice (Tarski – konečnost). *Množina x je konečná $\equiv$ každá neprázdná $a \subseteq \mathcal{P}(x)$ má maximální prvek vůči inkluzi. Pokud je x konečná, tak píšeme $\text{Fin}(x)$.*

Poznámka. *Množina x je konečná $\iff$ každá neprázdná $a \subseteq \mathcal{P}(x)$ má minimální prvek vůči inkluzi.*

Lemma. *Je-li x konečná a y nekonečná, potom $x \prec y$.*

Definice (Dedekendovská konečnost). *Množina x je dedekendovsky konečná $\equiv (\forall y)(y \subset x \rightarrow y \prec x)$.*

Lemma. *Je-li x konečná, potom je i dedekendovsky konečná.*

Definice (Spočetnost). *Množina x je*

- spočetná $\equiv x \approx \omega$,
- nejvýše spočetná $\equiv$ je spočetná nebo konečná,
- nespočetná $\equiv$ není nejvýše spočetná.

Věta. *Platí*

1. *každá shora omezená podmnožina $A \subseteq \omega$ je konečná,*
2. *každá shora neomezená podmnožina $A \subseteq \omega$ je spočetná.*

Věta. *Jsou-li A, B spočetné množiny, pak $A \cup B$ a $A \times B$ jsou také spočetné.*

Mohutnosti množin různých čísel

Věta. Množiny $\mathbb{Q}, \mathbb{Z}$ jsou spočetné.

Věta (Cantorova). Pro každou množinu x platí $x \prec \mathcal{P}(x)$.

Důsledek. Množina $\mathcal{P}(\omega)$ je nespočetná.

Věta. ${}^\omega 2 \approx \mathcal{P}(\omega) \approx \mathbb{R} \approx (0, 1)$.

Věta. Množiny $\mathbb{Q}$ a $\mathbb{R}$ nejsou ekvivalentní.

Důkaz. (Cantorovou diagonální metodou).

Víme, že $\mathbb{R} \approx (0, 1)$. Pro spor předpokládejme, že existuje bijekce $\mathbb{N} \approx (0, 1)$. Tedy všechna reálná čísla v tomto intervalu lze zapsat jako očíslovaný seznam prvků $a_{ij} \in \{0, 1, \dots, 9\}$.

$$r_1 = 0.a_{11}a_{12}a_{13}a_{14}\dots$$

$$r_2 = 0.a_{21}a_{22}a_{23}a_{24}\dots$$

$$r_3 = 0.a_{31}a_{32}a_{33}a_{34}\dots$$

...

Nyní sestrojíme nové reálné číslo $r = 0.b_1b_2b_3b_4\dots$ takto: $b_i = \begin{cases} 1, & \text{je-li } a_{ii} \neq 1 \\ 2, & \text{je-li } a_{ii} = 1 \end{cases}$.

Tímto jsme zajistili, že r se liší od každého r_i alespoň v i -té cifře. Tedy $r \neq r_i$ pro všechna i . Přitom ale $r \in (0, 1)$, takže mělo být v seznamu, což je spor. $\square$

3.4.7 Dobré uspořádání

Definice (Uspořádání). Relace $R \subseteq V \times V$ je na třídě X

- trichotomická $\equiv (\forall x, y \in X)(x R y \vee y R x \vee x = y)$, ... porovnatelnost
- ostré uspořádání $\equiv$ je ireflexivní a tranzitivní na X ,
- uspořádání $\equiv$ je reflexivní, slabě antisymetrická a tranzitivní na X ,
- lineární uspořádání $\equiv$ je trichotomická a uspořádání na X .

Pokud je R uspořádání, tak místo $x R y$ píšeme $x \leq_R y$. Pro ostré uspořádání píšeme $x <_R y$.

Definice. Nechť R je uspořádání na třídě A a $X \subseteq A$. Prvek $a \in A$ je

- horní mez třídy $X \equiv (\forall x \in X)(x \leq_R a)$, ... také majoranta
- maximální prvek třídy $X \equiv a \in X \wedge (\nexists x \in X)(a <_R x)$,
- největší prvek třídy $X \equiv a \in X$ a je to horní mez X ,
- supremum třídy $X \equiv$ je nejmenší prvek třídy všech horní mezí X .

Největší prvek, resp. supremum značíme $\max_R(X)$, resp. $\sup_R(X)$; pokud existují. Obdobně definujeme minorantu, minimální prvek, nejmenší prvek a infimum.

Definice. Řekneme, že uspořádání R je na množině A

- husté $\equiv (\forall x, y \in A)(x < y \rightarrow (\exists z \in A)(x < z < y))$,
- dobré $\equiv$ každá neprázdná $B \subseteq A$ má nejmenší prvek,
- úplné $\equiv$ každá neprázdná, shora omezená $B \subseteq A$ má supremum.

Ordinální a kardinální čísla

Ordinální čísla jsou zobecnění přirozených čísel a udávají typy dobře uspořádaných množin. Pomocí ordinálních čísel pak lze nadefinovat kardinální čísla, což jsou speciální ordinální čísla, která měří mohutnosti množin. Ordinální čísla definujeme tak, aby byla dobře uspořádána relací $\in$, takže na nich budeme moci provádět takzvanou transfinite indukci.

Definice. *Třída X je tranzitivní $\equiv y \in x \in X \rightarrow y \in X$.*

Definice (Ordinální číslo). *Množina x je ordinální číslo $\equiv$*

- 1. x je tranzitivní,*
- 2. relace $\in$ je dobré ostré uspořádání na x .*

Třidu všech ordinálních čísel značíme O_n

Definice (Kardinální čísla). *Ordinální číslo α nazveme kardinálním číslem, pokud každé menší ordinální číslo $\beta < \alpha$ má i menší mohutnost (tj. α nelze bijektivně zobrazit na žádnou podmnožinu β). Označíme-li jako C_n třídu všech kardinálních čísel, můžeme zapsat tuto definici ve tvaru:*

$$\alpha \in C_n \iff (\forall \beta)(\beta < \alpha \implies \neg(\beta \approx \alpha)).$$