



FIT SZZ – NI-SPOL 2024

Zpracované otázky více či méně inspirované zdroji níže a ověřené podle obsahu přednášek platných k LS 2023/2024.

Navazuje stylem na  BI-SPOL 2022 .

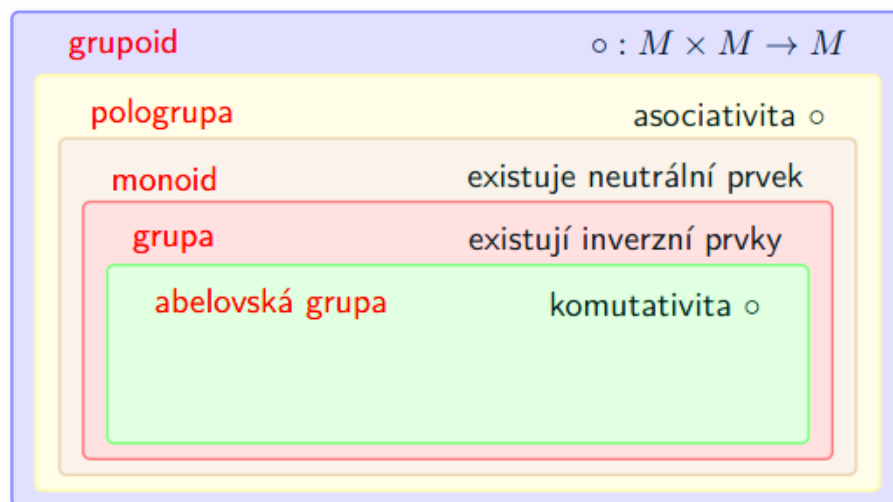
Zdroje

- https://fit-wiki.cz/_media/%C5%A1kola/st%C3%A1tnice/mi-spol-revision23.pdf - inspirace pro otázky 1-5 z NI-MPI
- https://fit-wiki.cz/_media/%C5%A1kola/szz-2022-common.pdf - inspirace skoro pro všechno
-  BI-SPOL 2022 - otázky 13-15 z NI-KOP
-  NI-PDP 2024 - otázky 16-20 z NI-PDP
- Přednášky platné k LS 2023/2024

1 – Teorie grup: Grupoidy, pologrupy, monoidy a grupy. Podgrupy, cyklické grupy a jejich generátory.

Grupoidy, pologrupy, monoidy a grupy

- Matematická struktura složená z množiny a binární operace na ní
 - Neprázdná množina M (nosič)
 - Binární operace $\cdot : M \cdot M \rightarrow M$
- Jednotlivé struktury dědí své vlastnosti



- Grupoid: neprázdná množina M s binární operací uzavřenou na M
 - Např. $(\mathbb{Q}, -)$: uzavřená ale ne asociativní
- Pologrupa: grupoid s asociativní operací: $(x \cdot x) \cdot x = x \cdot (x \cdot x)$
 - Např. $(\mathbb{N}_+, +)$: sčítání kladných přirozených čísel, chybí neutrální 0
- Monoid: pologrupa s neutrálním prvkem: $x \cdot e = e \cdot x = x$
 - Např. $(\mathbb{R}_0^+, +)$: existuje neutrální prvek, ale chybí inverze
- Grupa: monoid, kde ke každému prvku a existuje inverzní b : $a \cdot b = e$
- Abelovská grupa: grupa, kde je operace komutativní: $a \cdot b = b \cdot a$
- Speciální abelovské grupy
 - \mathbb{Z}_n^+ : Aditivní grupa modulo n , $\{0, \dots, n-1\}$, 0 neutrální, 1 generátor
 - \mathbb{Z}_n^\times : Multiplikativní grupa modulo n , $\{1, \dots, n-1\}$ nesoudělná s n , 1 neutrální
- Vlastnosti neutrálního prvku a inverze
 - Neutrální prvek existuje je v monoidu právě jeden: $e = e' \cdot e = e'$
 - Inverze je jednoznačně určena: pokud bychom měli k prvku a inverze b, c , pak
$$c = c \cdot e = c \cdot (a \cdot b) = (c \cdot a) \cdot b = e \cdot b = b$$
- Cayleyho tabulka
 - Tabulka se všemi prvky v záhlaví a prvním sloupci, výsledky operace složení dvou prvků v příslušném políčku
 - Poznáme uzavřenost, najdeme neutrální prvek a inverzi
 - Komutativitu poznáme podle diagonální symetrie

- Nepoznáme snadno asociativitu
- Tabulka grupy tvoří latinský čtverec (všechny prvky v každém řádku), implikace ale neplatí obráceně
- Cayleyho graf
 - Prvky jsou uzly, hrana označuje složení s jiným prvkem a jeho výsledek

Podgrupy, cyklické grupy a jejich generátory

- Podgrupa $H = (N, \cdot)$ grupy $G = (M, \cdot)$
 - Definice
 - $N \subseteq M$
 - H je grupa
 - Triviální podgrupy
 - $(\{e\}, \cdot)$: pouze neutrální prvek
 - (M, \cdot) : grupa samotná
 - Průnik podgrup je podgrupa
 - Uzavřenost platí, asociativita je zachována díky stejné operaci, neutrální prvek zůstává stejný, uzavřenost vůči inverzi také platí
 - Alternativní kritérium: pro každé $a, b \in N$ platí $a \cdot b^{-1} \in N$
- Řád grupy: počet prvků nosiče
 - Lagrangeova věta: řád podgrupy dělí řád grupy
 - Sylowova věta: pro prvočíselný dělitel p konečného řádu grupy n , pokud p^k dělí n , pak grupa obsahuje podgrupu řádu p^k
- Grupa generovaná množinou: průnik všech podgrup dané grupy G , které množinu obsahují
 - Značíme $\langle N \rangle$
 - Podle věty o průniku grup je to skutečně grupa
 - (Pod)grupu generovanou množinou lze získat "grupovým obalem": složení všech mocnin (včetně záporných) všech prvků množiny
- Generátor grupy: prvek jednoprvkové generující množiny
- Cyklická grupa: pokud existuje a takové, že $\langle a \rangle = G$
 - a je pak její generátor
 - \mathbb{Z}_n^\times je cyklická, pokud n je 2, 4, p^k nebo $2p^k$ (kde $p \geq 3$ liché prvočíslo, k je kladné přirozené)
- Řád prvku: nejmenší kladné přirozené m takové, že $g^m = e$
 - Pokud neexistuje, řád prvku $\text{ord}(g)$ je nekonečno
 - Řád prvku g je roven řádu grupy $\langle g \rangle$
- Hledání generátorů
 - V cyklické grupě řádu n s generátorem a
 - a^k je také generátor, iff k a n jsou nesoudělná
 - Důkaz pomocí rozkladu $1 = uk - vn$
 - Důsledkem předchozí věty je následující
 - V cyklické grupě řádu n je $\phi(n)$ generátorů (Eulerova funkce)
 - \mathbb{Z}_n^\times je cyklická grupa řádu $n - 1$, má $\phi(n - 1)$ generátorů
- Libovolná podgrupa cyklické grupy je opět cyklická
- V grupě řádu n platí pro všechna a : $a^n = e$ (důsledek Lagrangeovy věty)
- MFV: pro libovolné prvočíslo p , $1 \leq a < p$ platí: $a^{p-1} \equiv 1 \pmod{p}$

Homomorfismus a izomorfismus

- Homomorfismus: zobrazení $h: M \rightarrow N$, pro nějž platí
 - Pro všechna $x, y \in M$: $h(x \cdot_G y) = h(x) \cdot_H h(y)$
 - Zachovává tedy strukturu danou operací (je jedno, co se aplikuje dříve)
 - Definujeme na grupoidu, definice se přenáší i na grupy
- Pokud je h
 - Injektivní: monomorfismus
 - Surjektivní: epimorfismus
 - Bijektivní: izomorfismus
- Grupy G, H jsou izomorfní, když existuje izomorfismus $G \rightarrow H$
 - Vlastnost vzájemného izomorfismu tvoří relaci ekvivalence na třídě grup
- Homomorfismus h grupy G do grupoidu H znamená, že $h(G) = (h(M), \cdot_H)$ je grupa
- Neutrální prvek grupy se homomorfismem zobrazí na neutrální prvek druhé, stejně tak inverze se zachovávají
- Libovolné dvě cyklické grupy stejného řádu jsou izomorfní
 - Izomorfismus se \mathbb{Z}_n^+ , $h(k) = a^k$
- Libovolná konečná grupa je izomorfní s nějakou grupou permutací

2 – Tělesa a okruhy: Základní definice a vlastnosti. Konečná tělesa. Okruhy polynomů, ireducibilní polynom.

Základní definice a vlastnosti

- Struktura s množinou a dvěma operacemi
- Okruh
 - Trojice $(M, +, \cdot)$
 - $(M, +)$ je abelovská grupa (aditivní grupa)
 - (M, \cdot) je monoid (multiplikativní monoid)
 - Platí levý a pravý distributivní zákon
 - Např. $(\mathbb{Z}, +, \cdot)$, čtvercové reálné matice $(\mathbb{R}^{n,n}, +, \cdot)$
 - Násobení nulovým prvkem dává nulový prvek: $a \cdot 0 = a \cdot (0+0) = a \cdot 0 + a \cdot 0$
- Obor integrity
 - Komutativní okruh, kde neexistují dělitelé nuly (nenulové a, b pro něž platí $a \cdot b = b \cdot a = 0$)
- Těleso
 - Okruh $(M, +, \cdot)$, kde $(M \setminus \{0\}, \cdot)$ je abelovská grupa (multiplikativní grupa)
 - Např. $(\mathbb{Q}, +, \cdot)$
- Homomorfismus a izomorfismus okruhů a těles
 - Zobrazení h z okruhu R do okruhu S :
 - h je homomorfismus z aditivní grupy R do S
 - h je homomorfismus z mult. monoidu R do S
 - platí $h(1_R) = 1_S$
 - Zobrazení h z tělesa R do tělesa S :
 - h je homomorfismus z aditivní grupy R do S
 - h je homomorfismus z mult. grupy R do S
 - Vlastnost izomorfismu je relací ekvivalence na třídě všech těles

Konečná tělesa

- Těleso s konečným počtem prvků (řádem)
- Např. \mathbb{Z}_p prvočíselného řádu s operacemi $+$ a \cdot modulo prvočíslo p
 - Aditivní grupa \mathbb{Z}_p^+
 - Má řád p
 - Každý nenulový prvek je generátorem a má tedy řád p
 - $(\mathbb{Z}_p, +)$ je grupou i pro neprvočíselná p
 - Multiplikativní grupa \mathbb{Z}_p^\times
 - Má řád $p - 1$
 - Je cyklická (má generátor)
 - Počet generátorů je roven $\varphi(p - 1)$
 - $(\mathbb{Z}_p \setminus \{0\}, \cdot_p)$ je grupa pouze pro prvočíselná p (jinak má dělitele 0)
 - Pro $k < p$, které dělí $p - 1$ existuje v \mathbb{Z}_p^\times podgrupa řádu k , která obsahuje ty prvky $a \in \mathbb{Z}_p$, pro které $a^k = 1$

- Galois field $GF(p^n)$
 - má p^n prvků
 - p : charakteristika tělesa
 - Aditivní grupa
 - Má řád p^n
 - Neutrální prvek je 0^n
 - Pro $n > 1$ není cyklická, pro každý prvek v platí $p \cdot v = 0$
 - Multiplikativní grupa
 - Má řád $p^n - 1$
 - Neutrální prvek je $0^{n-1}1$
 - Inverzi lze nalézt pomocí EEA v polynomiálním čase
 - Je vždy cyklická

Okruhy polynomů, ireducibilní polynom

- Polynom nad okruhem s formální proměnnou x
Mějme okruh R a $a_i \in R$, $i = 0, 1, \dots, n$. Formální výraz tvaru

$$P(x) = \sum_{i=0}^n a_i x^i$$

- Stupeň polynomu $\deg(P(x))$: $a_k \neq 0$ takové, že $k \leq n$ je nejvyšší možné
- Nulový polynom: $P(x) = 0$, má nedefinovaný stupeň
- Okruh polynomů $R[x]$ nad okruhem R : (M množina všech polynomů, $+$, \cdot)
Buď R okruh. Potom množina všech polynomů nad okruhem R spolu s operacemi sčítání a násobení definovanými předpisy

$$\sum_{i=0}^n a_i x^i + \sum_{i=0}^n b_i x^i := \sum_{i=0}^n (a_i + b_i) x^i$$

$$\left(\sum_{j=0}^n a_j x^j \right) \cdot \left(\sum_{k=0}^m b_k x^k \right) := \sum_{i=0}^{n+m} \left(\sum_{j+k=i} a_j b_k \right) x^i,$$

- Základní vlastnosti polynomů
 - Stupeň násobku polynomů je součtem stupňů činitelů
 - Dělení se zbytkem
 - Mějme těleso T a nenulové polynomy $f(x), g(x) \in T[x]$
 - Pak existují jednoznačně určené polynomy $q(x), r(x)$ takové, že $f(x) = q(x)g(x) + r(x)$, kde $r(x)$ je nulový nebo je jeho stupeň ostře menší než stupeň $g(x)$
 - Největší společný dělitel $h(x)$ polynomů $g(x)$ a $f(x)$: $\gcd(f(x), g(x))$
 - $h(x)$ dělí $f(x)$: existuje $f(x) = q(x)h(x)$
 - $h(x)$ dělí $g(x)$
 - každý polynom, který dělí zároveň $f(x)$ i $g(x)$, dělí také $h(x)$
 - Bézoutova rovnost pro polynomy
 - Pro nenulové polynomy $f(x), g(x)$ nad tělesem T existují $u(x), v(x)$ t. ž.
 - $\gcd(f(x), g(x)) = u(x)f(x) + v(x)g(x)$
 - Důkaz indukcí na počet stupňů $f(x), g(x)$

- Prvek $a \in T$ je kořenem p , iff $p(x) = (x - a)g(x)$, kde $g(x)$ je stupně $\text{ord}(f(x)) - 1$
- Ireducibilní polynom
 - "Princip prvočísel v polynomech"
 - $P(x)$ je ireducibilní nad okruhem K , iff pro každé dva polynomy $A(x), B(x)$ platí

$$A(x) \cdot B(x) = P(x) \implies (\deg(A(x)) = 0 \text{ NEBO } \deg(B(x)) = 0).$$
 - Existuje horní odhad počtu monických polynomů stupně n ireducibilních nad \mathbb{Z}_p , kde p je prvočíslo a monický polynom má koef. nejvyšší mocniny 1
 - Rozhodnout o ireducibilitě i najít ired. polynom lze v polynomiálním čase narozdíl od analogie v prvočíslech

3 – Funkce více proměnných: gradient, Hessián, definitnost matic, extrémy funkcí více proměnných bez omezení a s rovnostními omezeními.

Gradient

- Základní pojmy (viz [BI-SPOL 2022](#))
 - Limita funkce v bodě
 - $c \in \mathbf{R}$ je limitou funkce f v bodě a , pokud
$$(\forall H_c)(\exists H_a)(\forall x \in D_f)(x \in H_a \setminus \{a\} \Rightarrow f(x) \in H_c).$$
 - Spojitost funkce f v bodě a
 - Funkce je v bodě definovaná
 - V bodě existuje limita zprava i zleva a nabývá stejné hodnoty
 - Derivace funkce f v bodě a
 - Funkce je definovaná na okolí bodu, derivace je (pokud existuje)
$$\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}$$
 - Funkce je diferencovatelná, pokud je derivace konečná
- Funkce více proměnných
 - Reálná funkce více reálných proměnných je zobrazení $D_f \rightarrow \mathbf{R}$
 - $D_f \subset \mathbf{R}^n$ pro n kladné celé
 - Graf funkce: množina
$$\Gamma_f = \{(b_1, b_2, \dots, b_n, f(b_1, b_2, \dots, b_n)) : (b_1, b_2, \dots, b_n) \in D_f\} \subset \mathbf{R}^{n+1}$$
- Limita funkce více proměnných
$$\forall H(L) \quad \exists H(\mathbf{b}) \quad \mathbf{x} \in (D_f \cap H(\mathbf{b})) \setminus \{\mathbf{b}\} \implies f(\mathbf{x}) \in H(L)$$
- Parciální derivace funkce f ve směru x_i v bodě $\mathbf{b} = (b_1, \dots, b_n)$, t. ž. $\exists H(\mathbf{b}) \subset D_f$
$$\frac{\partial f}{\partial x_i}(\mathbf{b}) = \lim_{h \rightarrow 0} \frac{f(b_1, b_2, \dots, b_i + h, \dots, b_n) - f(b_1, b_2, \dots, b_i, \dots, b_n)}{h}$$
 - Směrnice tečny ke grafu funkce f ve směru osy x_i
- Gradient funkce f v bodě $\mathbf{b} \in D_f$: řádkový vektor
$$\nabla f(\mathbf{b}) = \left(\frac{\partial f}{\partial x_1}(\mathbf{b}), \frac{\partial f}{\partial x_2}(\mathbf{b}), \dots, \frac{\partial f}{\partial x_n}(\mathbf{b}) \right)$$
- Derivace funkce f v bodě $\mathbf{b} \in D_f$ ve směru $\mathbf{v} \in \mathbf{R}^{n,1}$, $\|\mathbf{v}\| = 1$
$$\nabla_{\mathbf{v}} f(\mathbf{b}) = \lim_{h \rightarrow 0} \frac{f(\mathbf{b} + h\mathbf{v}) - f(\mathbf{b})}{h}$$
 - Pro spojitě parciální derivace na okolí \mathbf{b} pak $\nabla_{\mathbf{v}} f(\mathbf{b}) = \nabla f(\mathbf{b}) \cdot \mathbf{v}$
- Tečná nadrovina

$$z = \frac{\partial f}{\partial x_1}(\mathbf{b})(x_1 - b_1) + \frac{\partial f}{\partial x_2}(\mathbf{b})(x_2 - b_2) + \cdots + \frac{\partial f}{\partial x_n}(\mathbf{b})(x_n - b_n) + f(\mathbf{b})$$

- Objekt v prostoru grafu funkce (\mathbb{R}^{n+1})
- Sjedení tečen (parc. derivací) ve všech směrech

Hessián

- Parciální derivace druhého řádu

$$\frac{\partial^2 f}{\partial x_j \partial x_i}(\mathbf{b}) = \frac{\partial}{\partial x_j} \left(\frac{\partial f}{\partial x_i} \right)(\mathbf{b})$$

- Zobrazení derivace lze skládat
- Smíšená druhá parc. derivace: pokud $i \neq j$

- Hessova matice (Hessián)

$$\nabla^2 f(\mathbf{b}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{b}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{b}) \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{b}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{b}) \end{pmatrix}$$

- Matice druhých parciálních derivací funkce v bodě
- Hessián je často symetrický: pokud existují druhé parc. derivace v obráceném pořadí a funkce je v bodě spojitá, jsou si rovny
- Druhá derivace v bodě \mathbf{b} ve směru, pokud má funkce na okolí \mathbf{b} spojitě všechny druhé parciální derivace

$$\nabla_{\mathbf{v}} (\nabla_{\mathbf{v}} f)(\mathbf{b}) = \mathbf{v}^T \cdot \nabla^2 f(\mathbf{b}) \cdot \mathbf{v}$$

Definitnost matic

- Pro matici $A \in \mathbb{R}^{n,n}$

1. pozitivně semidefinitní, pokud $\mathbf{x}^T A \mathbf{x} \geq 0$ pro $\forall \mathbf{x} \in \mathbb{R}^{n,1}$;

2. pozitivně definitní, pokud $\mathbf{x}^T A \mathbf{x} > 0$ pro $\forall \mathbf{x} \in \mathbb{R}^{n,1}, \mathbf{x} \neq 0$;

3. negativně semidefinitní, pokud $\mathbf{x}^T A \mathbf{x} \leq 0$ pro $\forall \mathbf{x} \in \mathbb{R}^{n,1}$;

4. negativně definitní, pokud $\mathbf{x}^T A \mathbf{x} < 0$ pro $\forall \mathbf{x} \in \mathbb{R}^{n,1}, \mathbf{x} \neq 0$;

5. indefinitní, pokud není pozitivně ani negativně semidefinitní.

- Kritéria pro *symetrickou* matici

■ Matice A je *pozitivně semidefinitní* právě tehdy, když všechna její vlastní čísla jsou *nezáporná*.

■ Matice A je *pozitivně definitní* právě tehdy, když všechna její vlastní čísla jsou *kladná*.

■ Matice A je *negativně semidefinitní* právě tehdy, když všechna její vlastní čísla jsou *nekladná*.

■ Matice A je *negativně definitní* právě tehdy, když všechna její vlastní čísla jsou *záporná*.

■ Matice A je *indefinitní* právě tehdy, když má alespoň jedno *kladné* a alespoň jedno *záporné* vlastní číslo.

- Sylvestrovo kritérium pro *symetrickou* matici a levé horní podmatice A_1, \dots, A_n

- Poz. def.: A_1, \dots, A_n mají kladný determinant
- Neg. def.: determinant A_k záporný pro k lichá a kladný pro k sudá
- Kritérium indefinitnosti: pokud má matice na diagonále prvky s různým znaménkem

Extrémy funkcí více proměnných bez omezení a s rovnostními omezeními

- Kritický bod (podezřelý z extrému)
 - Stacionární bod b : gradient roven 0
 - Bod ve kterém gradient f neexistuje
- Základ pro platnost následujících podmínek
 - Bod b je stacionární
 - Funkce má na jeho okolí spojitě druhé parciální derivace
- Nutná podmínka existence lokálního extrému (implikaci nelze obrátit!)
 - (Neostré) lokální minimum v b : Hessián je pozitivní semidefinitní
 - (Neostré) lokální maximum v b : Hessián je negativně semidefinitní
- Postačující podmínka existence lokálního extrému a sedlového bodu
 - Hessián v b pozitivně definitní: ostré lokální minimum
 - Hessián v b negativně definitní: ostré lokální maximum
 - Hessián v b indefinitní: sedlový bod
- Extrémy s rovnostními omezeními
 - Úloha minimalizace $f(x)$ za podmínek (rovnostních vazeb)

$$g_j(\mathbf{x}) = 0, \quad j \in \hat{m},$$
 - Množina přípustných řešení

$$\mathcal{M} = \{\mathbf{x} \in D: (\forall j \in \hat{m})(g_j(\mathbf{x}) = 0) \wedge (\forall k \in \hat{p})(h_k(\mathbf{x}) \leq 0)\}$$
 - Lagrangeova funkce $L: M \times \mathbf{R}^m \rightarrow \mathbf{R}$

$$L(\mathbf{x}; \lambda) = f(\mathbf{x}) + \sum_{j=1}^m \lambda_j g_j(\mathbf{x})$$
 - Lagrangeovy multiplikátory: koeficienty $\lambda = (\lambda_1, \dots, \lambda_m)$
 - Pokud má gradient vazby a optimalizované funkce stejný směr, vazba v daném bodě neprotíná vrstevnici funkce

Nechť $f, g_j, j \in \{1, \dots, m\}$ mají spojité všechny druhé parciální derivace na nějaké otevřené nadmnožině $\tilde{\mathcal{M}} \supset \mathcal{M}$. Pokud dvojice $(x^*; \lambda^*) \in \mathbb{R}^n \times \mathbb{R}^m$ splňuje podmínky:

- (0) (0. derivace) $x^* \in \mathcal{M}$;
- (1) (1. derivace) $\forall i, \frac{\partial L}{\partial x_i}(x^*; \lambda^*) = 0$;
- (2) (2. derivace) pro každý (sloupcový) vektor $0 \neq v \in \mathbb{R}^n$ splňující

$$\nabla g_j(x^*) \cdot v = 0, \quad \text{pro } \forall j \in \{1, \dots, m\},$$

platí

$$v^T \cdot \nabla_x^2 L(x^*; \lambda^*) \cdot v > 0;$$

kde $\nabla_x^2 L$ je Hessova matice funkce L vzhledem k proměnným $x = (x_1, x_2, \dots, x_n)$,

potom je x^* bodem ostrého lokálního minima.

4 – Integrál funkcí více proměnných (Darbouxova konstrukce).

Darbouxův integrál jedné proměnné

- Rozdělení intervalu $[a, b]$: konečná množina

$$\sigma = \{x_0, x_1, \dots, x_n\}$$

- Dělicí body: $a = x_0 < \dots < x_n = b$
- Norma rozdělení $v(\sigma)$: $\max\{\Delta_k: k = 1, 2, \dots, n\}$, kde $\Delta_k = x_k - x_{k-1}$
- Ekvidistantní rozdělení: Δ_k je stejné pro všechna $k \in 1, \dots, n$
- Darbouxův součet funkce f při rozdělení σ
 - Horní součet

$$M_i = \sup_{x \in [x_{i-1}, x_i]} f(x) \quad S_f(\sigma) = \sum_{i=1}^n M_i \Delta_i$$

- Dolní součet

$$m_i = \inf_{x \in [x_{i-1}, x_i]} f(x) \quad s_f(\sigma) = \sum_{i=1}^n m_i \Delta_i$$

- Darbouxův integrál funkce f na intervalu $[a, b]$
 - Horní integrál

$$D_f = \inf \left\{ S_f(\sigma) : \sigma \text{ je rozdělení } [a, b] \right\}$$

- Dolní integrál

$$d_f = \sup \left\{ s_f(\sigma) : \sigma \text{ je rozdělení } [a, b] \right\}$$

- Darbouxův integrál: pokud se horní a dolní rovnají

$$\int_a^b f(x) dx = D_f = d_f$$

- Pro funkci spojitou na intervalu existuje Darbouxův integrál a rovná se limitě Darbouxova horního i dolního součtu podle normální posloupnosti rozdělení (jejich normy jdou k 0)

- Vlastnosti Darbouxova integrálu (za podmínky spojitosti funkcí na intervalu)
 - Aditivita
 - Multiplikativita (vytýkání konstanty)
- Newtonova formule (za podmínky spojitosti a primitivní funkce F na (a, b))

$$\int_a^b f(x) dx = \lim_{x \rightarrow b-} F(x) - \lim_{x \rightarrow a+} F(x)$$

- Per partes pro určitý integrál

$$\int_a^b f(x)g(x)dx = [f(x)G(x)]_a^b - \int_a^b f'(x)G(x)dx$$

- Substitute v určitém integrálu

$$\int_{\alpha}^{\beta} f(\varphi(t)) \cdot \varphi'(t)dt = \int_{\varphi(\alpha)}^{\varphi(\beta)} f(x)dx$$

Darbouxův integrál více proměnných

- Funkce dvou proměnných, integrál je objem pod grafem funkce
- Dvourozměrné rozdělení $\sigma = \sigma_x \times \sigma_y$
- Darbouxova suma (horní a dolní) analogicky

$$S_f(\sigma) = \sum_{i=1}^n \sum_{j=1}^m M_{i,j}(x_i - x_{i-1})(y_j - y_{j-1})$$

- Darbouxův integrál (horní a dolní) analogicky; obecný pokud se h. a d. rovnají

$$D_f = \inf \left\{ S_f(\sigma) : \sigma \text{ je rozdělení } D \right\}$$

- Výpočet (Fubiniho věta)

- Pro funkci $f(x, y)$ integrabilní na $D = [a, b] \times [c, d]$ (musí být spojitá)
- Pokud existuje jeden z integrálů

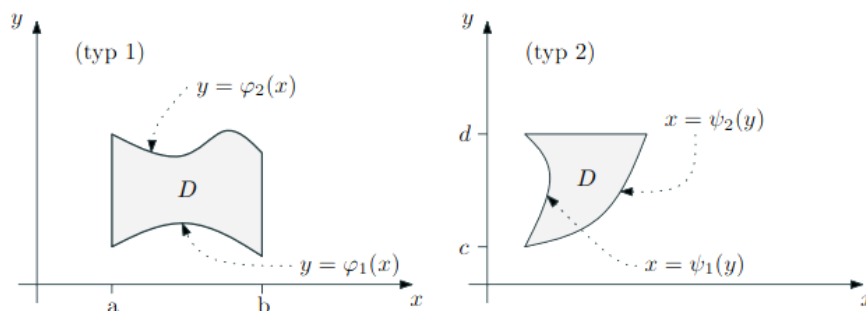
$$\int_a^b \left(\int_c^d f(x, y)dy \right) dx \quad \text{nebo} \quad \int_c^d \left(\int_a^b f(x, y)dx \right) dy$$

pak je roven dvojnému integrálu

- Výpočet nad obecnou oblastí

- Omezená podmnožina D čtvercové oblasti $[a, b] \times [c, d]$
- Dvojitý Darbouxův integrál se definuje pomocí alternativní funkce f , která má hodnotu 0 pro x patřící do čtvercové oblasti, ale ne do D
- Množina míry nula
 - Pro každé $\varepsilon > 0$ existují obdélníky takové, že jimi lze množinu pokrýt tak, aby měly v součtu obsah $< \varepsilon$
- Omezená funkce je integrabilní, pokud je spojitá *skoro všude*
 - *Není spojitá pouze na množině míry nula*

- Oblasti s hranicí danou spojitými funkcemi



■ je-li D typu 1, máme

$$\iint_D f(x, y) dx dy = \int_a^b \left(\int_{\varphi_1(x)}^{\varphi_2(x)} f(x, y) dy \right) dx.$$

■ je-li D typu 2, máme

$$\iint_D f(x, y) dx dy = \int_c^d \left(\int_{\psi_1(y)}^{\psi_2(y)} f(x, y) dx \right) dy.$$

- Substituce v dvojném integrálu

- Mějme zobrazení (funkční předpis v integrálu)

$$\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^n, \Psi(\mathbf{v}) = (\Psi_1(\mathbf{v}), \dots, \Psi_n(\mathbf{v}))$$

- Jacobiho matice zobrazení (pokud existují parciální derivace)

$$J_\Psi = \begin{pmatrix} \frac{\partial \Psi_1}{\partial v_1} & \dots & \frac{\partial \Psi_1}{\partial v_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_n}{\partial v_1} & \dots & \frac{\partial \Psi_n}{\partial v_n} \end{pmatrix}$$

- V řádku je gradient i -té složky zobrazení
- Zobrazení je bijekce a determinant jeho Jacobiho matice je nenulový
- Pak pro každou spojitou funkci platí

$$\int_{\psi(D)} f(\mathbf{x}) d\mathbf{x} = \int_D f(\Psi(\mathbf{v})) |\det J_\Psi(\mathbf{v})| d\mathbf{v}$$

5 – Numerická matematika: reprezentace čísel v počítači, chyby vznikající při výpočtech s pohyblivou řádovou čárkou, podmíněnost úlohy a stabilita numerických algoritmů.

Reprezentace čísel v počítači

- Numerická matematika: metody pro hledání přibližných řešení matematických úloh a jejich spolehlivosti (lineární rovnice, integrály, optimalizace, vlastní čísla)
- IEEE-754
 - Strojové číslo $x = (-1)^s \cdot 2^{e-b} \cdot m$
 - Znaménko s
 - Signifikant (mantisa) m
 - Exponent e
 - Parametr b
 - Různé přesnosti
 - Jednoduchá (single): délka $m = 23$, délka $e = 8$, $b = 127$
 - Speciální hodnoty
 - NaN: $e = \max$, $m \neq 0$
 - $\pm \text{Inf}$: $e = \max$, $m = 0$
 - ± 0
 - Normalizovaná čísla: $0 < e < 2^{\text{délka } e} - 1$; se skrytou jedničkou
 - Subnormální čísla: $e = 0$, $m \neq 0$; se skrytou nulou; $x = (-1)^s \cdot (0.m)_2 \cdot 2^{1-b}$
- Strojová čísla: konečná podmnožina racionálních čísel
 - Ne všechna čísla desítkové soustavy mají reprezentaci ve dvojkové
 - Množina strojových čísel F je charakterizovaná m , e , b
- Strojová přesnost ε_F : vzdálenost čísla 1 od nejbližšího většího v F
 - Pro single precision $\varepsilon_F = 2^{-23}$
 - Vzdálenost jakéhokoliv normalizovaného čísla $x \in F$
 - Minimálně $\varepsilon_F \cdot \frac{|x|}{2}$
 - Maximálně $\varepsilon_F \cdot |x|$

Chyby vznikající při výpočtech s pohyblivou řádovou čárkou

- Chyby algoritmů obecně
 - Chyba modelu (zanedbání fyziky apod.)
 - Chyba dat (nepřesné měření)
 - Chyba algoritmu (není úplný)
 - Zaokrouhlovací chyba - aritmetické operace s pohyblivou řádovou čárkou
- Číslo $a \in \mathbf{R}$ reprezentujeme jako strojové $\alpha \in F$
 - Zobrazení $fl: \mathbf{R} \rightarrow F$

- Chyba reprezentace
 - Absolutní chyba: $|\alpha - a|$
 - Relativní chyba (pro $a \neq 0$): $\frac{|\alpha - a|}{|a|}$
- Zaokrouhlovací chyba při reprezentaci
 - Zobrazení fl přiřazuje reálnému číslu nejbližší strojové
 - Zaokrouhlovací strategie
 - K nejbližšímu (round to nearest, ties to even/away from zero/infty)
 - K \pm nekonečnu
 - Náhodně
 - Useknutí směrem k nule
 - Pro reálné číslo a v rozsahu normalizovaných strojových čísel a usekávání k nule je $fl(a) = (1.m_1...m_{23})_2 \cdot 2^l$
 - Absolutní chyba: $|a - fl(a)| \leq 2^{-23+l}$
 - Relativní chyba: $\leq 2^{-23}$
 - Zaokrouhlovací jednotka u : horní mez pro relativní chybu
 - Závisí na konkrétním a , např. pro 0 je $u = 0$
- Krácení
 - Ztráta platných cifer při provádění aritmetických operací
 - Typicky odčítání podobně velkých čísel
 - Podle IEEE standardu se posune mantisa a nejnižší cifry se doplní 0
 - Prevence
 - Přeformulovat problém bez odčítání
 - Použít rozvoj funkce do Taylorovy řady
 - Použít intervalovou aritmetiku (dvě strojová čísla jako krajní body)
 - Zvýšení přesnosti nemusí dát přesnější výsledek
 - Krácení někdy může vyrušit zaokrouhlovací chyby a být tak výhodné

Podmíněnost úlohy a stabilita numerických algoritmů

- Stabilita
 - Přesný výstup $V^*(d)$, výstup ve strojové aritmetice $V(d)$
 - Dopředná chyba: $\Delta v = V^*(d) - V(d)$
 - Odchylka spočítaného řešení od přesného
 - Zpětná chyba: nejmenší (v normě) číslo Δd takové, že $V^*(d + \Delta d) = V(d)$
 - Promítnutí chyby strojové implementace do vstupu algoritmu
 - Tedy jaký problém byl vyřešen ve skutečnosti
 - Zpětně stabilní algoritmus: Δd je pro všechny vstupy d malé
- Podmíněnost
 - Závislost změny výstupu na malé změně vstupu o δd
 - Relativní číslo podmíněnosti (rel. změna výstupu ku rel. změně vstupu)

$$C_r = \lim_{\epsilon \rightarrow 0^+} \sup_{\substack{d + \delta d \in D \\ \|\delta d\| \leq \epsilon}} \frac{\frac{\|V^*(d + \delta d) - V^*(d)\|}{\|V^*(d)\|}}{\frac{\|\delta d\|}{\|d\|}}$$
 - Dobře podmíněná úloha: C_r je blízko 1
 - Tedy změna vstupu proporcionálně odpovídá změně výstupu

6 – Testování statistických hypotéz. T-testy, testy nezávislosti, testy dobré shody.

Testování statistických hypotéz

- Hypotézy
 - Nulová hypotéza H_0 : tvrzení, o němž cheme rozhodnout
 - Alternativní hypotéza H_A : opačné tvrzení
- Náhodný vektor \mathbf{X} s rozdělením P_θ (úvod viz BI-SPOL 2022)
 - Parametr rozdělení $\theta \in \Theta_0 \cup \Theta_A = \Theta$
- Nulová hypotéza platí, pokud má \mathbf{X} rozdělení P_θ , kde $\theta \in \Theta_0$
- Nezamítnutí H_0 není silný výsledek, nepotvrzuje platnost H_0
- Chyby
 - Chyba prvního druhu: zamítneme H_0 , přestože platí
 - Hypotézy volíme tak, aby tato chyba byla závažnější - chceme se vyhnout “falešnému obvinění”
 - Pravděpodobnost této chyby budeme kontrolovat
 - Hladina významnosti α : maximální pravděpodobnost chyby
 - Typicky 1 % či 5 %
 - Chyba druhého druhu: nezamítneme H_0 , ačkoliv neplatí
 - Pravděpodobnost této chyby nedokážeme kontrolovat
 - Testy konstruujeme tak, aby byla “nejnižší možná”
 - Takový test je *nejméně*
- Typy testů
 - Parametrické: tvrzení se týká hodnoty parametru rozdělení \mathbf{X}
 - Neparametrické: tvrzení se týká různých jiných vlastností rozdělení či jeho tvaru obecně (testy dobré shody)
- p -hodnota
 - Minimální hladina významnosti, na níž lze pro určitou realizaci \mathbf{X} zamítnout H_0
 - Při testu získáme p -hodnotu, kterou porovnáme s kýženou hladinou α
 - Pokud $p \leq \alpha$, pak na hladině α zamítáme H_0
- Testová statistika T
 - Funkce náhodného vektoru \mathbf{X} , u které při platnosti H_0 známe její rozdělení
 - Kritický obor: podmnožina S_α oboru hodnot T
 - Při platnosti H_0 má T hodnotu v S_α s pravděpodobností nejvýše α
 - Pokud $T \in S_\alpha$, zamítáme H_0

T-testy

- Parametrické testy založené na statistice se studentovým rozdělením (za platnosti H_0)
- Jednovýběrový t-test

H_0	H_A	testová statistika T	kritický obor
$\mu = \mu_0$	$\mu \neq \mu_0$	$T = \frac{\bar{X}_n - \mu_0}{s_n} \sqrt{n}$	$ T \geq t_{\alpha/2, n-1}$
$\mu \leq \mu_0$	$\mu > \mu_0$		$T \geq t_{\alpha, n-1}$
$\mu \geq \mu_0$	$\mu < \mu_0$		$T \leq -t_{\alpha, n-1}$

- Test o střední hodnotě náhodného výběru $X_1, \dots, X_n \sim N(\mu, \sigma^2)$
- Při neznámém rozptylu používáme výběrový rozptyl
 - Testová statistika má pak rozdělení t_{n-1} (proto jednovýběrový) při platnosti H_0
- (Při známém rozptylu používáme statistiku s $N(0, 1)$ - to ale není jednovýběrový test)
- Např. ověřujeme vyváženost mince, pokud padla pětkrát z dvaceti hodů hlava; $H_0: \mu = 0.5$

- Párový t-test

H_0	H_A	testová statistika T	kritický obor
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$T = \frac{\bar{Z}_n}{s_Z} \sqrt{n}$	$ T \geq t_{\alpha/2, n-1}$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$		$T \geq t_{\alpha, n-1}$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$		$T \leq -t_{\alpha, n-1}$

- Test o rovnosti středních hodnot náhodného výběru párů $(X_1, Y_1), \dots, (X_n, Y_n)$ z dvojrozměrného rozdělení se středními hodnotami (μ_1, μ_2)
- Definujeme $Z_i = X_i - Y_i$ a $\mu_\Delta = \mu_1 - \mu_2$
- Test se pak provede jako jednovýběrový t-test hypotézy $H_0: \mu_\Delta = 0$
- Např. ověření, jestli se hodnota ukazatele po provedení nějaké procedury změnila, tj. jestli procedura měla významný vliv
- Dvouvýběrový t-test
 - Test o rovnosti středních hodnot dvou nezávislých náhodných výběrů $X_1, \dots, X_n \sim N(\mu_1, \sigma_1^2)$ a $Y_1, \dots, Y_m \sim N(\mu_2, \sigma_2^2)$, n se nemusí rovnat m
 - Při stejných rozptylech $\sigma_1^2 = \sigma_2^2$

H_0	H_A	testová statistika T	kritický obor
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$T = \frac{\bar{X}_n - \bar{Y}_m}{s_{12}} \sqrt{\frac{n \cdot m}{n + m}}$	$ T \geq t_{\alpha/2, n+m-2}$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$		$T \geq t_{\alpha, n+m-2}$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$		$T \leq -t_{\alpha, n+m-2}$

, kde

$$s_{12} = \sqrt{\frac{(n-1)s_X^2 + (m-1)s_Y^2}{n+m-2}}$$

- Při různých rozptylech $\sigma_1^2 \neq \sigma_2^2$

H_0	H_A	testová statistika T	kritický obor
$\mu_1 = \mu_2$	$\mu_1 \neq \mu_2$	$T = \frac{\bar{X}_n - \bar{Y}_m}{s_d}$	$ T \geq t_{\alpha/2, n_d}$
$\mu_1 \leq \mu_2$	$\mu_1 > \mu_2$		$T \geq t_{\alpha, n_d}$
$\mu_1 \geq \mu_2$	$\mu_1 < \mu_2$		$T \leq -t_{\alpha, n_d}$

, kde

$$s_d = \sqrt{\frac{s_X^2}{n} + \frac{s_Y^2}{m}} \quad \text{a} \quad n_d = \frac{s_d^4}{\frac{1}{n-1} \left(\frac{s_X^2}{n} \right)^2 + \frac{1}{m-1} \left(\frac{s_Y^2}{m} \right)^2}$$

- Např. ověření, jestli se hodnoty ukazatelů při dvou různých procedurách o různém počtu vzorků významně liší

Testy dobré shody

- Multinomické rozdělení $M(n, p)$
 - Diskrétní náhodná veličina \mathbf{X} nabývající k hodnot $1, \dots, k$
 - Každá hodnota má pravděpodobnost p_1, \dots, p_k
 - Rozdělení \mathbf{X} označíme jako $\mathbf{p} = (p_1, \dots, p_k)$
 - Provedeme náhodný výběr X_1, \dots, X_n o n prvcích z rozdělení \mathbf{p} (kategorické)
 - Jeho výsledek (se zanedbáním pořadí) zaznamenáme pomocí četností výskytů hodnot $1, \dots, k$
 - Získáme náhodný vektor $\mathbf{N} = (N_1, \dots, N_k)$ s *multinomickým rozdělením*

$$P(N_1 = n_1, \dots, N_k = n_k) = \frac{n!}{n_1! \dots n_k!} p_1^{n_1} \dots p_k^{n_k}, \text{ kde}$$

$$n_i = 0..n \text{ a } n_1 + \dots + n_k = n$$

- Speciální případy pro $k = 2$ (binomické), pro $n = 1$ (zpět kategorické \mathbf{p})
- Pearsonův test dobré shody
 - Testová statistika: Pearsonova statistika
 - V čitateli je rozdíl naměřených a teoretických četností v kvadrátu
 - Pokud naměříme mnoho výskytů u teoreticky málo pravděpodobné hodnoty, statistika bude velmi velká
 - Test je asymptotický, proto potřebujeme alespoň 5 výskytů pro každou hodnotu (případně podle Yarnoldova kritéria)
 - Při známých parametrech

H_0	H_A	testová statistika χ^2	kritický obor
$\mathbf{p}' = \mathbf{p}$	$\mathbf{p}' \neq \mathbf{p}$	$\chi^2 = \sum_{i=1}^k \frac{(N_i - np_i)^2}{np_i}$	$\chi^2 \geq \chi_{\alpha, k-1}^2$

- Testujeme shodnost diskretních rozdělení
- Máme náhodný výběr \mathbf{X} o velikosti n z diskretního rozdělení \mathbf{p}'
- Četnosti $\mathbf{N} = (N_1, \dots, N_k)$ mají multinomické rozdělení $M(n, \mathbf{p}')$
- H_0 : skutečné hodnoty pravděpodobností jsou p_1, \dots, p_k

- Při neznámých parametrech

H_0	H_A	testová statistika χ^2	kritický obor
$\mathbf{p}' = \mathbf{p}$	$\mathbf{p}' \neq \mathbf{p}$	$\chi^2 = \sum_{i=1}^k \frac{(N_i - np_i)^2}{np_i}$	$\chi^2 \geq \chi_{\alpha, k-m-1}^2$

- Testujeme, že náhodný výběr \mathbf{X} pochází z rozdělení F_θ s neznámým parametrem θ
 - Např. jestli jde o výběr z normálního rozdělení $N(\theta_1, \theta_2)$
- Situaci převedeme na test hypotézy pro multinomické rozdělení
 - Rozdělíme hodnoty z \mathbf{X} na intervaly, jejich četnosti zaznamenáme jako $\mathbf{N} = (N_1, \dots, N_k)$
 - Intervaly volíme opět tak, aby každý měl aspoň 5 výskytů
- H_0 : rozdělení \mathbf{p} se rovná teoretickému \mathbf{p}' , které závisí na neznámém θ
- θ odhadneme metodou minimálního χ^2
 - Pro tento bodový odhad má testová statistika asymptoticky rozdělení χ_{k-m-1}^2
 - Odečítáme stupně volnosti za odhadnuté parametry (celkem počet binů - počet parametrů - 1)

Testy nezávislosti

- Náhodný vektor $\mathbf{X} = (Y, Z)$ s diskretním rozdělením, Y nabývá hodnot $1..r$ a Z $1..c$
 - Sdružená pravděpodobnost $p_{ij} = P(Y = i, Z = j)$
 - Marginální pravděpodobnosti $p_{i\bullet}$ a $p_{\bullet j}$
- Náhodný výběr z rozdělení \mathbf{X} o velikosti $n \rightarrow$ náhodná multinomická veličina N_{ij}
- Kontingenční tabulka

Kontingenční tabulka

Y	Z			Σ
	1	...	c	
1	N_{11}	...	N_{1c}	$N_{1\bullet}$
...
r	N_{r1}	...	N_{rc}	$N_{r\bullet}$
Σ	$N_{\bullet 1}$...	$N_{\bullet c}$	n

Matice pravděpodobností

Y	Z			Σ
	1	...	c	
1	p_{11}	...	p_{1c}	$p_{1\bullet}$
...
r	p_{r1}	...	p_{rc}	$p_{r\bullet}$
Σ	$p_{\bullet 1}$...	$p_{\bullet c}$	1

- Náhodná matice \mathbf{N} rozměru $r \times c$ se složkami N_{ij}
- Marginální četnosti $N_{i\bullet}$ a $N_{\bullet j}$
- Testujeme nezávislost dvou náhodných veličin, tj. že $p_{ij} = p_{i\bullet} \times p_{\bullet j}$
- Počet nezávislých parametrů $p_{i\bullet}$ a $p_{\bullet j}$ je $m = (r - 1) + (c - 1)$
 - Poslední parametr je již pevně určen předchozími, aby byl součet = 1
- Počet stupňů volnosti je $(r - 1) * (c - 1)$
- Použijeme test dobré shody při neznámých parametrech, jichž je m

H_0	H_A	testová statistika χ^2	kritický obor
$p_{ij} = p_{i\bullet} p_{\bullet j}$	$p_{ij} \neq p_{i\bullet} p_{\bullet j}$	$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{\left(N_{ij} - \frac{N_{i\bullet} N_{\bullet j}}{n}\right)^2}{\frac{N_{i\bullet} N_{\bullet j}}{n}}$	$\chi^2 \geq \chi_{\alpha, (r-1)(c-1)}^2$

7 – Základy teorie informace a kódování, entropie.

Entropie

- Entropie je “mírou neuspořádanosti či neurčitosti”
- Základní pojmy, s nimiž entropie pracuje
 - Diskrétní náhodná veličina: X
 - Pravděpodobnostní funkce veličiny: $p(x)$, odpovídá $P(X = x)$
- Míra neurčitosti (vlastní informace) hodnoty x : $I(x) = -\log p(x)$
- Definice entropie

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

- Entropie je tedy střední hodnotou, očekávanou mírou neurčitosti X
 - Logaritmus je o základu 2 (odtud je jednotka entropie *bit*)
 - $0 \log 0 = 0$
 - Nezáporná, pro jisté jevy nulová
 - Maximální je pro rovnoměrné rozdělení
 - Při pozorování nezávislých jevů se sčítá
- Sdružená entropie

$$H(X, Y) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y)$$

- Podmíněná entropie

$$H(Y|X) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x) \quad , \text{ kde } p(y|x) = p(x, y)/p(x)$$

- Řetězové pravidlo
$$H(X, Y) = H(X) + H(Y|X)$$
 - $H(Y|X)$ tedy určuje, jaká informace je v Y navíc proti X
- Relativní entropie (Kullback-Leiblerova vzdálenost)

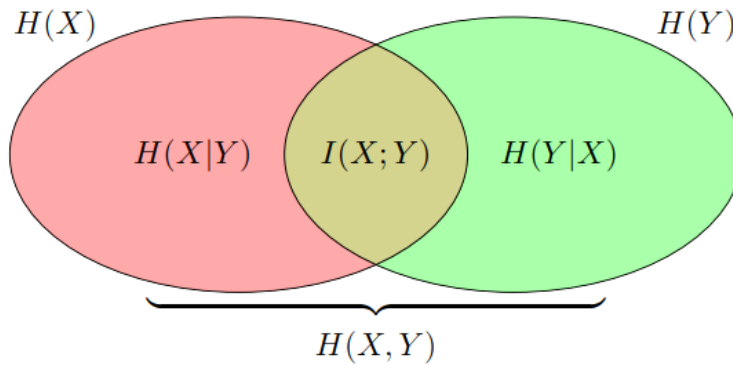
$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$

- Je nezáporná a rovná se 0, pokud jsou p a q stejné
 - Není to ale metrika - není symetrická a neplatí trojúhelníková nerovnost
- Vzájemná informace

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

- Relativní entropie skutečného sdruženého rozdělení a nezávislých rozdělení se stejnými marginálami
- Určuje tak “míru nezávislosti” sdruženého rozdělení
- Je nezáporná a symetrická

- Vztah entropie a vzájemné informace



- Jensenova nerovnost
 - Pro konvexní funkci f

$$E f(X) \geq f(E X)$$
 - Pro ryze konvexní f platí rovnost, pokud $X = EX$ s pravděpodobností 1
- Informační nerovnost
 - $D(p||q)$ je nezáporná,
 $D(p||q) = 0$ iff $p(x) = q(x)$ pro všechna x
- Důsledky informační nerovnosti
 - Nezápornost vzájemné informace
 - $I(X, Y) = 0$ iff X a Y jsou nezávislé
 - Narozdíl od kovariance lze pomocí $I(X, Y)$ tedy poznat i nezávislost
 - Maximalizace entropie
 - $H(X) \leq \log|X|$: entropie je nejvýše log. počtu prvků, kterých může X nabývat
 - Rovnost (maximalizace) nastává, když má X rovnoměrné rozdělení
 - Podmiňování redukuje entropii
 - $H(X|Y) \leq H(X)$
 - $H(X|Y) = H(X)$ iff X a Y jsou nezávislé
 - Znalost další náhodné veličiny Y může pouze zredukovat neurčitost v X (informace neublíží)
 - Platí ale pouze v průměru - pro konkrétní y může být $H(X|Y = y)$ větší než $H(X)$

Kódování

- Teorie kódování řeší problém zápisu zdrojové zprávy do nějaké přenositelné posloupnosti symbolů co nejefektivnějším způsobem
 - Zahrnuje problematiku komprese a odolnosti vůči chybám přenosu (chyby se v této otázce neřeší)
- Základní pojmy
 - Zdrojová zpráva $x_1 \dots x_n$: konečná posloupnost složená ze *znaků* z množiny \mathbf{X} , jejichž četnosti jsou určeny náhodnou veličinou X
 - \mathbf{D} -ární abeceda přenositelných *symbolů* z množiny \mathbf{D}
 - \mathbf{A}^* : množina všech konečných řetězců složených z prvků množiny \mathbf{A}
- Kód: zobrazení $C: \mathbf{X} \rightarrow \mathbf{D}^*$
- Kódové slovo: obraz $C(x)$ zprávy $x \in \mathbf{X}$, má délku $l(x)$

- Střední délka kódu

$$L(C) = \sum_{x \in \mathcal{X}} \ell(x)p(x)$$

- Platí $L(C) = E\ell(X)$
- Hierarchie kódů
 - Všechny kódy
 - Nesingulární kódy: jednotlivá kódová slova jsou zpětně dekodovatelná
 - C je prosté zobrazení

$$x \neq x' \Rightarrow C(x) \neq C(x')$$
 - Zprávy (posloupnosti kódových slov) ale nemusí jít dekodovat
 - Jednoznačně dekodovatelné kódy: celé zprávy jsou zpětně dekodovatelné
 - C^* je nesingulární
 - Rozšíření kódu C na posloupnosti zobrazením $C^*: \mathbf{X}^* \rightarrow \mathbf{D}^*$
 - Dekódování ale může vyžadovat přijetí celé zprávy
 - Instantní kódy: žádné kódové slovo není prefixem jiného
 - Jakmile v dosud přijatých symbolech najdeme kódové slovo, můžeme ho převést na původní znak x

- Kraftova nerovnost

- Pro délky kódových slov l_1, \dots, l_n nad D -ární abecedou platí

$$\sum_i D^{-\ell_i} \leq 1$$

- Ke každé n -tici délek splňující tuto nerovnost existuje instantní kód s kódovými slovy těchto délek
- McMillan: toto platí i pro jednoznačně dekodovatelné kódy, takže ubráním požadavku instantní dekodovatelnosti si nepřidáme další možnosti délek kódových slov - řešení optimality instantních kódů řeší i optimalitu jednoznačně dekodovatelných

- Dolní mez délky instantního kódu

$$L(C) \geq H_D(X)$$

- Pro D -ární kód a entropii s logaritmem o základu D
- Rovnost (optimalita) nastává, pokud

$$D^{-\ell_i} = p_i \text{ pro všechna } i = 1, \dots, |\mathcal{X}|$$

- Střední délka optimálního kódu

$$H_D(X) \leq L(C^*) < H_D(X) + 1$$

- Pro optimální instantní D -ární kód C^* diskrétní náhodné veličiny X
- Optimálním kódem se tak od dolní meze dané entropií vzdálíme max o 1
- Důkaz pomocí Shannonova kódu, kde $\ell_i = \lceil \log_D \frac{1}{p_i} \rceil$
- Huffmanovo kódování (tvorba binárního kódu)
 - Spojujeme nejméně pravděpodobné hodnoty do dvojic rekurzivně
 - Výsledné jediné hodnotě přiřadíme prázdný řetězec jako kódové slovo
 - Zpětným chodem konstruuje kódová slova původních hodnot
 - Huffmanův D -ární kód je optimální
 - Hladový algoritmus generuje globální optimum, což je pozoruhodné 😊

8 – Markovské řetězce s diskretním časem. Jejich limitní vlastnosti.

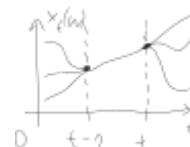
Náhodný proces

- Reálný náhodný proces
 - Systém náhodných veličin $\mathbf{X} = \{X_t \mid t \in T\}$
 - Náhodná veličina X_t
 - Indexová množina $T \subseteq \mathbf{R}$
 - Spočetná: diskretní čas
 - Nespočetná: spojitý čas
 - Množina stavů \mathbf{S} : minimální podmnožina \mathbf{R} t.ž. $P(X_t \in \mathbf{S}) = 1$
 - Spočetná
 - Nespočetná
- Trajektorie náhodného procesu (realizace)

$$f(t) := X_t(\omega), \quad \forall t \in T$$
 - Funkce $f: T \rightarrow \mathbf{R}$, přiřazuje časové značce hodnotu
 - $X_t(\omega)$ je hodnotou náhodné veličiny v čase t při realizaci procesu $\omega \in \Omega$
- Rozdělení v čase

$$\mathbf{p}_i(n) := P(X_n = i), \quad \mathbf{p}(n) := (\mathbf{p}_1(n), \mathbf{p}_2(n), \dots)$$
- Matice pravděpodobností přechodu z $i \in \mathbf{S}$ v čase n do $j \in \mathbf{S}$ v čase m

$$\mathbf{P}_{ij}(n, m) := P(X_m = j \mid X_n = i)$$



Markovský řetězec

- Markovský řetězec s diskretním časem
 - Náhodný proces $\{X_n \mid n \in \mathbf{N}_0\}$ splňující markovskou podmínku

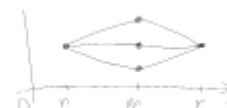
$$P(X_n = s \mid X_{n-1} = s_{n-1}, \dots, X_1 = s_1, X_0 = s_0) = P(X_n = s \mid X_{n-1} = s_{n-1})$$
 - Markovská podmínka znamená zapominání historie, aktuální krok je ovlivněn pouze krokem přímo předešlým
 - Odpovídá přechodům mezi stavy

$$P(X_{n_0} = s_0, \dots, X_{n_k} = s_k) = \mathbf{p}_{s_0}(n_0) \cdot \mathbf{P}_{s_0 s_1}(n_0, n_1) \cdot \mathbf{P}_{s_1 s_2}(n_1, n_2) \cdot \dots \cdot \mathbf{P}_{s_{k-1} s_k}(n_{k-1}, n_k)$$
- Chapman-Kolmogorova rovnice

$$\mathbf{P}(n, r) = \mathbf{P}(n, m) \cdot \mathbf{P}(m, r)$$
 - Pro matici přechodu markovského řetězce a $n \leq m \leq r$
- Homogenita

$$P(X_{n+1} = j \mid X_n = i) = P(X_1 = j \mid X_0 = i)$$
 - Pravděpodobnost přechodu z i do j je nezávislá na pořadí kroku
 - Proto pravděpodobnost přechodu z i do j za čas n je mocninou matice \mathbf{P}

$$\mathbf{P}(m, m+n) = \mathbf{P}(0, n) = \mathbf{P}^n, \text{ značíme pak } \mathbf{P}(n) := \mathbf{P}(0, n)$$



- Chapman-Kolmogorova rovnici pro homogenní řetězec pak je

$$\mathbf{P}(n+m) = \mathbf{P}(n) \cdot \mathbf{P}(m), \text{ což odpovídá } \mathbf{P}^{n+m} = \mathbf{P}^n \cdot \mathbf{P}^m$$
- A pravděpodobnosti stavů v čase n získáme pouze z pravděpodobností počátečních stavů a matice přechodů jako

$$\mathbf{p}(n) = \mathbf{p}(m) \cdot \mathbf{P}^{n-m} = \mathbf{p}(0) \cdot \mathbf{P}^n$$
- Stochastická matice (přechodu)
 - Definice
 - Má nezáporné prvky
 - Součet řádku je roven 1
 - Součin stochastických matic je stochastická matice
 - K libovolné čtvercové stochastické matici existuje homogenní markovský řetězec s diskrétním časem s touto maticí přechodu
- Deterministické počáteční rozdělení: právě jeden stav má pravděpodobnost 1
- Stacionární rozdělení π

$$\pi \cdot \mathbf{P} = \pi$$
 - π je rozdělení: má nezáporné prvky a součet prvků je roven 1

Limitní vlastnosti

- Klasifikace stavů
 - Trvalý: když v něm začnu, existuje doba, za kterou se do něj vrátím
 - Střední doba návratu: průměrný čas první návštěvy

$$\mu_i := \sum_{n=1}^{\infty} n f_{ii}(n)$$
 pro $f_{ii}(n)$ = pst. první návštěvy v kroce n ,
 - Nenulový: střední doba návratu je konečná
 - Nulový: střední doba návratu je $+\infty$
 - Přechodný: pravděpodobnost návratu je menší než 1
 - Střední doba návratu: definována jako $+\infty$
- Periodicita stavů
 - Perioda $d(i)$ stavu i : gcd časů, kdy se řetězec vrátí do stavu i
 - Klasifikace stavů
 - Periodický: $d(i) > 1$
 - Aperiodický: $d(i) = 1$
- Vzájemně dosažitelné stavy
 - V konečném čase se lze dostat tam a zpět (nenulová pst. přechodu $\mathbf{P}_{ij}(n)$)
 - Mají shodnou periodu a jsou shodného typu
- Jednoznačný rozklad množiny stavů
 - Uzavřená množina stavů: pravděpodobnost přechodu jinde je rovna 0
 - Absorbční stav: uzavřená množina tvořená jediným stavem
 - Nerozložitelná (ireducibilní) množina: všechny stavy vzájemně dosažitelné
 - Stavy lze rozložit ve tvaru

$$S = T \cup C_1 \cup C_2 \cup \dots$$
 - T : všechny přechodné stavy
 - C_i : vzájemně disjunktní nerozložitelné uzavřené mn. trvalých stavů

$$\mathbf{P} = \begin{matrix} & \begin{matrix} T & C_1 & C_2 & \dots \end{matrix} \\ \begin{matrix} T \\ C_1 \\ C_2 \\ \vdots \end{matrix} & \begin{pmatrix} \mathbf{T} & \mathbf{R}_1 & \mathbf{R}_2 & \dots \\ \mathbf{0} & \mathbf{C}_1 & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \end{matrix}$$

- Existence stacionárního rozdělení
 - Řetězec lze rozložit na uzavřené nerozložitelné podřetězce s maticí přechodu C_i
 - Pro nerozložitelný markovský řetězec
 - Všechny stavy přechodné nebo trvalé nulové: stacionární rozdělení neexistuje
 - Všechny stavy trvalé nenulové: stacionární rozdělení existuje a je jednoznačně určené
 - Napříč C_i jsou lineárně nezávislá
 - Jejich libovolná konvexní kombinace je stac. rozdělením původního řetězce
 - Konvexní kombinace: lineární kombinace se součtem koeficientů rovným 1
 - Pro konečnou množinu stavů existuje trvalý nenulový stav, tudíž existuje stacionární rozdělení

- Pohlcení

- Matici přechodu zapíšeme ve tvaru

$$\mathbf{P} = \begin{matrix} & \begin{matrix} T & C \end{matrix} \\ \begin{matrix} T \\ C \end{matrix} & \begin{pmatrix} \mathbf{T} & \mathbf{R} \\ \mathbf{0} & \mathbf{C} \end{pmatrix} \end{matrix}$$

- Čas absorpce: minimální čas přechodu z T do C
- U_{ij} : pst. pohlcení přes stav j při startu z i
- N_{ik} : střední počet průchodů stavem k před pohlcením
- $N_{i\bullet}$: střední doba do pohlcení při startu z i
- Tyto matice získáme jako

$$\mathbf{N} = (\mathbf{I} - \mathbf{T})^{-1}, \quad \mathbf{U} = \mathbf{N} \cdot \mathbf{R}, \quad \mathbf{N}_{\bullet} = \mathbf{N} \cdot \mathbf{1}$$

- Fundamentální matice řetězce $(\mathbf{I} - \mathbf{T})$: je regulární, má inverzi

- Konvergence

- Pokud jsou trvalé stavy aperiodické
- Matice \mathbf{P}^n konverguje k

$$\lim_{n \rightarrow +\infty} \mathbf{P}^n = \lim_{n \rightarrow +\infty} \begin{matrix} & \begin{matrix} T & C \end{matrix} \\ \begin{matrix} T \\ C \end{matrix} & \begin{pmatrix} \mathbf{T}^n & \mathbf{R}_n \\ \mathbf{0} & \mathbf{C}^n \end{pmatrix} \end{matrix} = \begin{matrix} & \begin{matrix} T & C \end{matrix} \\ \begin{matrix} T \\ C \end{matrix} & \begin{pmatrix} \mathbf{0} & \mathbf{U} \\ \mathbf{0} & \tilde{\mathbf{C}} \end{pmatrix} \end{matrix}$$

- Matice \mathbf{C}^n konverguje ke stacionárním rozdělením podřetězců v řádcích

9 – Markovské řetězce se spojitým časem.

Souvislost s Markovskými řetězci s diskrétním časem a s Poissonovým procesem.

Markovské řetězce se spojitým časem

- Markovský řetězec se spojitým časem
 - Náhodný proces

$$\{X_t \mid t \geq 0\}$$
 - $t \in \mathbf{R}$, množina stavů \mathbf{S} je spočetná
 - Splňuje markovskou podmínku

$$P(X_{t_k} = s_k \mid X_{t_{k-1}} = s_{k-1}, \dots, X_{t_0} = s_0) = P(X_{t_k} = s_k \mid X_{t_{k-1}} = s_{k-1})$$
 - Matice přechodu za čas s mezi časy s a t

$$\mathbf{P}_{ij}(t, s) := P(X_s = j \mid X_t = i)$$
- Chapman-Kolmogorova věta analogicky k diskrétnímu případu

$$\mathbf{P}(t, r) = \mathbf{P}(t, s) \cdot \mathbf{P}(s, r), \text{ pak } \mathbf{P}(t, t+s) = \mathbf{P}(0, s) := \mathbf{P}(s)$$
- Homogenita analogicky: pst. přechodu za čas s se v čase nemění

$$\mathbf{P}(t, t+s) = \mathbf{P}(0, s) := \mathbf{P}(s)$$
 - Rozdělení v čase t lze získat z matice přechodu

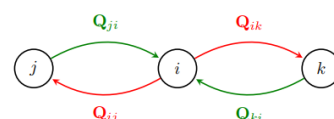
$$\mathbf{p}(t) = \mathbf{p}(0)\mathbf{P}(t)$$
 - Matici přechodu $\mathbf{P}(t)$ ale nezískáme snadno, proto se definuje následující
- Matice skokových intenzit
 - Neexistuje "atomická" matice jako v diskrétním čase, kde $\mathbf{P} = \mathbf{P}(1)$
 - Chování řetězce lze popsat pro $\mathbf{P}(h)$, kde $h \rightarrow 0_+$
 - Matice skokových intenzit je "derivací \mathbf{P} v čase 0"

$$\mathbf{Q} := \lim_{h \rightarrow 0_+} \frac{1}{h}(\mathbf{P}(h) - \mathbf{I}) = \lim_{h \rightarrow 0_+} \frac{1}{h}(\mathbf{P}(h) - \mathbf{P}(0))$$

tedy pro její prvky platí

$$Q_{ii} = \lim_{h \rightarrow 0_+} \frac{P_{ii}(h) - 1}{h}, \quad Q_{ij} := \lim_{h \rightarrow 0_+} \frac{P_{ij}(h)}{h}$$
 - V čase 0 se nikam nepřesuneme, proto $\mathbf{P}(0) = \mathbf{I}$
 - Prvek na diagonále je ≤ 0 : intenzita, s jakou chceme odejít
 - Mimo diagonálu naopak ≥ 0 : intenzita, s jakou tam chceme přejít
 - Součet prvků v řádku je roven 0
- Získání matice přechodu z matice skokových intenzit
 - Matice přechodu je řešením soustavy diferenciálních rovnic

$$\mathbf{p}'(t) = \mathbf{p}(t)\mathbf{Q}, \quad \mathbf{p}(0) = \mathbf{p}_{\text{initial}}$$
 - Řešení závisí na počátečním rozdělení
 - Změna pravděpodobnosti $\mathbf{p}'(t)$ pro stav i sestává ze zisku (intenzity z ostatních stavů sem) a ztráty (intenzity ven)



- Kolmogorovy rovnice
 - Dopředná

$$\mathbf{P}'(t) = \mathbf{P}(t) \cdot \mathbf{Q}$$
 - Zpětná

$$\mathbf{P}'(t) = \mathbf{Q} \cdot \mathbf{P}(t)$$
- Stacionární rozdělení
 - Chceme rozdělení π splňující

$$\pi \mathbf{P}(t) = \pi$$
 - Nalezneme ho jako řešení rovnice

$$\pi \mathbf{Q} = 0$$
 - Stacionární rozdělení existuje pro konečnou množinu stavů \mathbf{S}
 - Rozdělení je stacionární, pokud splňuje detailní rovnováhu

$$\pi_j Q_{ji} = \pi_i Q_{ij}$$

Poissonův proces

- Čítací proces: $\{N_t \mid t \geq 0\}$ s nezápornými celočíselnými neklesajícími trajektoriemi
- Poissonův proces:
 - Pro nezávislé náhodné veličiny

$$\{X_j \mid j \in \mathbb{N}\}$$
 s exponenciálním rozdělením $\text{Exp}(\lambda)$
 - Definujeme časy událostí vzdálené podle X_j jako

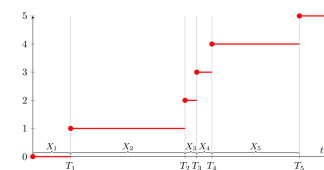
$$\{T_n \mid n \in \mathbb{N}\}, \text{ kde } T_0 = 0, \quad T_n = T_{n-1} + X_n = \sum_{j=1}^n X_j$$

- Poissonův proces je pak

$$\{N_t \mid t \in [0, +\infty)\}$$

se schodovitou zprava spojitou trajektorií

$$N_t(\omega) := \max\{n \in \mathbb{N}_0 \mid T_n(\omega) \leq t\}$$



- Bezpečnost exponenciálního rozdělení souvisí s markovským řetězcem
 - Pokud jsme čekali deset minut, pravděpodobnost příjezdu autobusu za dalších pět minut je stejná, jako kdybychom dosud vůbec nečekali
- Simulace pomocí procesu skokových intenzit
 - Sestavíme matici skokových intenzit
 - Čas do výskoku z i je exponenciální s $\lambda_i = -Q_{ii}$
 - Pravděpodobnost skoku do jednotlivých stavů je dána poměrem intenzit Q_{ij}

$$Q = \begin{pmatrix} 0 & \lambda & 0 & 0 & \dots \\ -\lambda & 0 & \lambda & 0 & \dots \\ 0 & -\lambda & 0 & \lambda & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Souvislost s Markovskými řetězci s diskrétním časem

- Diskrétní řetězec časovaný Poissonovým procesem
 - Pro Poissonův proces
$$\{N_t \mid t \geq 0\}$$
s intenzitou λ
 - Máme homogenní markovský řetězec s diskrétním časem
$$\{Y_n \mid n \in \mathbb{N}_0\}$$
a maticí přechodu \mathbf{D}
 - Definujeme homogenní markovský proces se spojitým časem jako
$$\{X_t \mid t \geq 0\}, \text{ kde } X_t := Y_{N_t},$$
tedy index kroku diskrétního procesu určuje Poissonův proces pro spojitý čas t a pravděpodobnost přeskočení z i do j je určena \mathbf{D}_{ij}
- Matice skokových intenzit řetězce X_t je
$$\mathbf{Q} = \lambda(\mathbf{D} - \mathbf{I})$$
a proto získáme matici přechodu diskrétního procesu jako
$$\mathbf{D} = \mathbf{I} + \frac{1}{\lambda} \mathbf{Q}$$
 - Matice \mathbf{D} musí být stochastická, což je splněno, pokud jsou diagonální prvky matice skokových intenzit konečné
 - To platí pro konečnou množinu stavů \mathbf{S}
- Postup simulace spojitého řetězce diskrétním
 - Získáme intenzitu Poissonova procesu λ
$$\lambda := \sup_{i \in S} (-Q_{ii})$$
 - Supremum musí být konečné, aby postup fungoval
 - Nekonečné bude např. v systému hromadné obsluhy $M|M|^\infty$
 - Zkonstruujeme matici přechodu \mathbf{D}
$$\mathbf{D} := \mathbf{I} + \frac{1}{\lambda} \mathbf{Q}$$
 - Vygenerujeme trajektorii Poissonova procesu s intenzitou λ
 - Vygenerujeme trajektorii diskrétního řetězce Y_n z matice \mathbf{D}
 - Trajektorii spojitého řetězce získáme jako
$$X_t := Y_{N_t}$$

10 – Systémy hromadné obsluhy a jejich limitní vlastnosti. Souvislost s Markovskými řetězci se spojitým časem.

Systémy hromadné obsluhy a jejich limitní vlastnosti

- Poissonův proces - viz otázku 9
- Model hromadné obsluhy
 - Příchody požadavků podle Poissonova procesu - exponenciální rozestupy
 - Doba vyřizování požadavků podle exponenciálního rozdělení
 - Všechny časy jsou vzájemně nezávislé
- Exponenciální závody - mezi nezávislými časy S a T
 - V systému je v čase t právě n požadavků
$$X_t = n$$
 - Jeden je zpracováván
 - $n - 1$ je ve frontě
 - Zpracovávání probíhá podle exponenciálních hodin $S \sim \text{Exp}(\mu)$
 - Fronta přibývá podle exponenciálních hodin $T \sim \text{Exp}(\lambda)$
 - Závod skončí v náhodném čase $\tau = \min\{S, T\}$
 - Pokud vyhraje server S , počet požavků se sníží o 1
$$X_{t+\tau} = n - 1$$
a naopak
 - Dynamika procesu: za náhodný čas $\tau = \text{Exp}(\mu + \lambda)$ volím nový stav $n - 1$ nebo $n + 1$ v poměru $\mu : \lambda$
- Proces hromadné obsluhy
$$\mathbf{X} = \{X_t \mid t \geq 0\}$$
 - Zaznamenává počet požadavků v systému hromadné obsluhy (ve frontě + na serveru) v čase t
- Kendallova notace: $A|S|c|K|N|D$
 - A : rozdělení časů příchodu F_A
 - S : rozdělení časů obsluhy F_S
 - c : počet obslužných míst
 - K : kapacita systémů (obslužná místa + fronta), výchozí $+\infty$
 - N : velikost populace, výchozí $+\infty$
 - D : typ obsluhy, výchozí FIFO
- Typy rozdělení A a S
 - $M(\lambda)$: exponenciální (markovské)
 - $D(d)$: degenerované - soustředěné v hodnotě d
 - G : jiné
- Hustota provozu ρ pro počet obslužných míst c
$$\rho = \frac{\lambda}{c\mu}$$
 - $\rho < 1$: systém se ustálí v rovnovážném stavu; $\rho > 1$: zahlcení

- Systém M|M|1

- Proces zrodu a zániku s parametry

$$\lambda_n = \lambda, \quad n \in \mathbb{N}_0, \quad \mu_m = \mu, \quad m \in \mathbb{N}$$

- Homogenní markovský řetězec se stavy $\mathbf{S} = \{0, 1, 2, \dots\}$

- Doba čekání ve frontě: náhodná veličina W

- Pokud je server prázdný

$$Q = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & \dots \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \end{matrix} & \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ \mu & -(\lambda + \mu) & \lambda & 0 & \dots \\ 0 & \mu & -(\lambda + \mu) & \lambda & \dots \\ 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix} \end{matrix}$$

$$P(W = 0) = P(X_t = 0) = \pi_0 = 1 - \rho = 1 - \frac{\lambda}{\mu}$$

- Pokud je server obsazen

$$(W | W > 0) \sim \text{Exp}(\mu - \lambda)$$

- Limitní vlastnosti

- Pokud $\lambda < \mu$, pak existuje stacionární rozdělení

$$\pi_n = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^n$$

- Střední počet požadavků v systému ve stacionárním stavu

$$E N = E N_s + E N_f = \frac{\rho}{1 - \rho}$$

- $E N_s$: střední počet obsluhovaných požadavků

- $E N_f$: střední počet požadavků ve frontě

- Systém M|M| ∞

- Proces zrodu a zániku s parametry

$$Q_{n,n+1} = \lambda_n \equiv \lambda, \quad Q_{n,n-1} = \mu_n = n \cdot \mu$$

- Nekonečně obslužných míst

- Doba čekání je tedy nulová

- Limitní vlastnosti: stacionární rozdělení je Poissonovo s parametrem λ/μ , existuje vždy a je ve tvaru

$$Q = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & \dots \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \end{matrix} & \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots \\ \mu & -(\lambda + \mu) & \lambda & 0 & \dots \\ 0 & 2\mu & -(\lambda + 2\mu) & \lambda & \dots \\ 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix} \end{matrix}$$

$$P_{\pi}(X_t = n) = \pi_n = \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n e^{-\frac{\lambda}{\mu}}$$

- Systém M|M|c

- Proces zrodu a zániku s intenzitami závislými na počtu obslužných míst c

$$Q_{n,n+1} = \lambda_n \equiv \lambda, \quad Q_{n,n-1} = \mu_n = \min\{c, n\} \cdot \mu = \begin{cases} n \cdot \mu & n \leq c, \\ c \cdot \mu & n > c. \end{cases}$$

- V matici intenzit se tedy intenzita skoku zpět (vyhraje obslužení) postupně škáluje, až se koeficient zastaví na max. kapacitě c

$$Q = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & \dots & c+1 & c+2 & \dots \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ c \\ c+1 \\ \vdots \end{matrix} & \begin{pmatrix} -\lambda & \lambda & 0 & 0 & \dots & 0 & 0 & \dots \\ \mu & -(\lambda + \mu) & \lambda & 0 & \dots & 0 & 0 & \dots \\ 0 & 2\mu & -(\lambda + 2\mu) & \lambda & \dots & 0 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & c\mu & -(\lambda + c\mu) & \lambda & 0 & \dots \\ 0 & 0 & 0 & 0 & c\mu & -(\lambda + c\mu) & \lambda & \dots \\ 0 & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix} \end{matrix}$$

- Limitní vlastnosti: stacionární rozdělení existuje pro

$$\rho = \frac{\lambda}{c\mu}$$

splňující $\rho < 1$, je pak ve tvaru

$$\pi_n = \begin{cases} \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n \pi_0 & n \leq c \\ \frac{c^c}{c!} \left(\frac{\lambda}{c\mu} \right)^n \pi_0 & n > c \end{cases}$$

- Z tabulky distribuční funkce Poissonova rozdělení lze získat nutný počet obslužných míst takový, aby pravděpodobnost, že požadavek bude čekat, byla maximálně nějaká

$$P_{\pi}(X_t \geq c) \leq 0.1$$

- Littleho věta

- Pro striktně stacionární (= ve stac. stavu) proces hromadné obsluhy

$$\mathbf{X} = \{X_t \mid t \geq 0\}$$

- A parametry

- EN : střední počet požadavků v systému
- ET : střední doba strávená požadavkem v systému
- λ : intenzita příchoďů

- Platí, jsou-li střední hodnoty konečné,

$$EN = \lambda \cdot ET$$

- Tedy požadavků je v systému tím více tím,

- čím více času požadavek průměrně stráví v systému
- čím větší je intenzita příchoďů

- Systém G|G|1

- λ : intenzita příchoďů
- μ : intenzita obsluhy, střední doba obsluhy pak $ES = 1/\mu$
- T_k : doba strávená k -tým požadavkem v systému rozdělená na
 - W_k : doba čekání ve frontě
 - S_k : doba obsluhy
- Z Littleho věty

- Střední počet požadavků v systému

$$EN = \lambda ET_k = \lambda EW_k + \lambda ES_k$$

- Střední počet požadavků ve frontě

$$EN_f = \lambda EW_k$$

- Lze také získat počáteční rozdělení

Souvislost s Markovskými řetězci se spojitým časem

- Hromadná obsluha využívá Poissonův proces pro příjem a odbavování požadavků, zachovává se bezpaměťovost
- Systémy lze vyjádřit pomocí matic intenzity spojitého markovského procesu

11 – Význam tříd NP a NPH pro praktické výpočty.

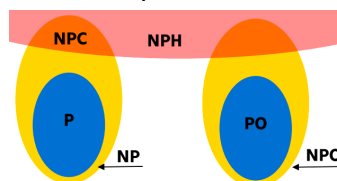
Kombinatorický problém a Turingův stroj

- Kombinatorická matematika
 - Řeší konečné diskrétní problémy - s konečným počtem proměnných a hodnot
 - Lze proto hrubou silou vyzkoušet všechny možnosti, ale prakticky to většinou není použitelné
- Kombinatorický problém
 - Vstupní a výstupní proměnné, vstupní a výstupní abeceda (jazyk)
 - Konfigurační proměnné: popisují konkrétní stav, lze z něj získat výstup
 - Omezení
 - Optimalizační kritérium (není vždy nutné)
- Konfigurace a řešení
 - Konfigurace: ohodnocení konfiguračních proměnných
 - Řešení: konfigurace vyhovující omezením
 - Optimální řešení: má nejlepší hodnotu optimalizačního kritéria
 - Suboptimální řešení: má "přijatelnou" hodnotu optimalizačního kritéria
- Typy řešení (výstupů): všechny mají stejné konfigurační proměnné, výstup se liší
 - Rozhodovací problém: existuje řešení?
 - Početní problém: kolik je řešení?
 - Konstruktivní problém: sestavit nějaké řešení
 - Enumerační problém: sestavit všechna řešení
- Turingův stroj (DTS = deterministický, NTS = nedeterministický)
 - Výpočetní model původně pro rozhodnutelnost problémů
 - Další modely jako RAM stroj či Booleův obvod
 - Postupné rozšiřování automatů (se zásobníkem, s páskou)
 - Konstrukce: neomezená páska, RW hlava, konečný stavový automat
 - Definice
 - Γ : množina symbolů pásky b
 - $\Sigma \subset \Gamma$: množina vstupních symbolů
 - Q : množina stavů automatu
 - q_0 : počáteční stav
 - q_{ANO}, q_{NE} : koncové stavy (pro rozhodovací problém)
 - δ : přechodová funkce $(Q \setminus \{q_{ANO}, q_{NE}\}) \times \Gamma \rightarrow \Gamma \times \{-1, 0, +1\}$
 - Podle stavu a symbolu na pásce zapíšeme nový symbol a případně posuneme hlavu
 - Program M pro DTS řeší rozhodovací problém Π , pokud se po konečném počtu kroků výpočet zastaví pro každou instanci problému Π
 - Lze omezit také na čas (počet kroků) a paměť
 - V poly. čase lze na TS emulovat RAM stroj s přímým adresováním
 - Způsob kódování problému ovlivní čas výpočtu jen polynomiálně

Třídy NP a NPH

- Třída P: DTS řeší problém v čase $O(n^k)$, kde n je velikost problému a k konečné
- Třída NP (nedeterministicky polynomiální)

- Problém nemusí mít polynomiální algoritmus, ale cesta prostorem konfigurací je polynomiální vůči velikosti instance
- NTS
 - Přejchodová funkce zobrazuje do potenční množiny
 - V každém kroku se tedy TS naklonuje, cesta k výsledku je ale polynomiální
 - Pokud NTS řeší problém v čase $T(n)$, DTS jej řeší v $2^{O(T(n))}$
- Pro problém Π definujeme množinu jeho instancí Π_{ANO}
- Problém patří do NP (ekvivalentní definice)
 - Podle NTS: NTS každou instanci v Π_{ANO} řeší v čase $O(n^k)$
 - Podle certifikátu: ověření správnosti řešení každé instance v Π_{ANO} patří do P
- Třída co-NP:
 - Doplněk k NP problémům
 - Certifikát odpovědi pro Π_{NE} je polynomiálně ověřitelný
- Třída NPH (NP-hard)
 - Problém patří do NPH, pokud efektivní řešení všech problémů z NP jdou efektivně zredukovat na efektivní řešení problému v NPH
 - Proto NPH problém je alespoň tak těžký, jako všechny NP problémy
 - SAT, 3SAT, knapsack, obchodní cestující, Hamiltonova kružnice, diskretní logaritmus
- Karpova redukce
 - Problém je Karp-redukovatelný, pokud lze na DTS v polynomiálním čase převést každou jeho instanci na instanci jiného tak, že výstup bude stejný
 - Platí tranzitivita
- Turingova redukce
 - Problém Π_1 je Turing-redukovatelný na Π_2 , pokud používá pro řešení instance Π_1 jako podprogram řešení Π_2
- Třída NPC (NP-complete)
 - Problém je NPH a zároveň NP - vše na něj zredukujeme a stále je to NP
 - Redukce Karpovská anebo Turingovská polynomiální
 - SAT je NPC (Cookova věta převádí všechny NP problémy na SAT)
- Třída NPO (NP-optimization)
 - Výstup je vůči vstupu polynomiální
 - Omezující podmínky lze vyhodnotit v poly. čase
 - Optimalizační kritérium lze vyhodnotit v poly. čase
- Třída PO (P-optimization)
 - Patří do NPO
 - Existuje polynomiální řešení pro DTS



Význam pro praktické výpočty

- NP a NPH problémy nelze efektivně řešit
- Neznámé problémy můžeme převést na jiné - např. SAT
- Někdy stačí přibližné/suboptimální řešení
- Metody zrychlování výpočtů
 - Redukce stavového prostoru: prořezáváme nepoužitelné výsledky
 - Pseudopolynomiální algoritmy: závisí polynomiálně na velikosti instance a na dalším nesouvisejícím parametru - $O(n * M)$
 - Aproximace: jednodušší heuristiky (simulované ochlazování, genetika) pro přibližné řešení
 - Randomizace
- Aproximativní algoritmy
 - Parametry
 - $C(S)$: hodnota optimalizačního kritéria řešení S
 - $APR(I)$: aproximované řešení instance I
 - $OPT(I)$: optimální řešení instance I
 - Relativní kvalita algoritmu: čím blíží 1, tím lepší
$$R \geq \max_{\forall I} \left\{ \frac{C(APR(I))}{C(OPT(I))}, \frac{C(OPT(I))}{C(APR(I))} \right\}$$
 - Relativní chyba algoritmu: čím blíží 0, tím lepší
$$\varepsilon \geq \max_{\forall I} \left\{ \frac{|C(APR(I)) - C(OPT(I))|}{\max\{C(OPT(I)), C(APR(I))\}} \right\}$$
 - R-aproximativní (ε -aproximativní) algoritmus každou instanci problému řeší v polynomiálním čase s relativní kvalitou R (chybou ε)
 - Třída APR: problémy R--aproximativní pro konečné R
 - R : aproximační práh
- PTAS (Polynomial Time Approximation Scheme)
 - Polynomiální aproximační schéma problému: APR algoritmus, který řeší každou instanci s relativní chybou $0 < \varepsilon < 1$ v polynomiálním čase
 - Tedy algoritmus parametrizovaný akceptovatelnou relativní chybou
- FPTAS (Full PTAS)
 - PTAS, jehož čas výpočtu závisí polynomiálně pouze na $1/\varepsilon$
 - Tedy algoritmus závisející pouze na požadované relativní chybě
- Typické NPH úlohy

12 – Experimentální vyhodnocení algoritmů, zejména randomizovaných.

Randomizované algoritmy

- Založené na náhodné volbě
- Dva základní přístupy
 - Monte Carlo: z kasina vyjdu vždy ráno, ale nevím, kolik peněz mi zůstane
 - Miller-Rabin test prvočísel, genetika, simulované ochlazování
 - Nemáme jistotu, že algoritmus rozhodne/zkonstruuje řešení
 - Las Vegas: nevím, kdy z kasina vylezu, ale určitě mě oberou o všechno
 - Rychlost QuickSortu závisí na volbě pivota
- Výhody
 - Strukturní jednoduchost
 - Očekávaná kvalita výsledku může být lepší než u aproximativních algo.
 - Nezávislým opakováním se dá kvalita zlepšit (Monte Carlo)
 - Využití stejné pravděpodobnosti pro jednotlivé možnosti při rozhodování dalšího kroku poskytuje nestrannost, když vzorkujeme
- Typy randomizovaných algoritmů
 - Náhodná procházka: v každém stavu náhodný posun do sousedního
 - Zaujatá náhodná procházka: pst. posunu není uniformně rozdělená
 - Heuristické algoritmy s randomizací: genetika, simulované ochlazování

Vyhodnocení algoritmů

- Cíle
 - Složitost: z hlediska teorie i nasazení
 - Kvalita řešení
 - Porozumění: závislost výpočetního času a kvality na parametrech
 - Eventuálně porovnání různých algoritmů navzájem
- Požadované vlastnosti
 - Zobecnění: obecná platnost vyhodnocení
 - Nezávislost na nezahrnutých veličinách: redukováný šum
 - Výpověď o významné závislosti: měření správných veličin
- Způsoby vyhodnocení
 - Jednoduché otázky lze zkoumat analyticky, složité experimentálně
 - Pro známé problémy existují benchmarky (SAT)
- IMRaD: způsob popisu experimentu
 - Introduction: kontext, proč, otázka výzkumu,
 - Methods: jak, proč zrovna tak, příp. co se zkoušelo dříve
 - Results: výsledky, odpověď na otázku výzkumu
 - Discussion: co odpověď znamená, perspektivy pro další výzkum

Experiment

- Struktura experimentu
 - Otázka - plán - provedení - sběr dat - interpretace - odpověď - (znovu)
- Provedení
 - Generátor instancí
 - Tvoří nezávislé instance podobného charakteru
 - Je tak zdrojem variance
 - Nemusíme mít k dispozici generátor, existují standardní sady pro známé problémy, zaručují ale dobré statistické vlastnosti?
 - Algoritmus
 - Pokud chceme hledat optimální parametry (vstupní metriku) algoritmu, je třeba izolovat je od ostatních, neznámých
 - Statistika
 - Zkoumá výstupní metriku: kvalitu (vůči optimu, je-li známé), dobu trvání, paměťovou náročnost
 - Potlačuje varianci vzniklou dříve a interpretuje (zobecňuje) metriku na celý algoritmus pomocí průměru, mediánu apod.
- Měření robustnosti
 - Zjišťujeme, jestli si algoritmus poradí s různě reprezentovanými problémy stejně dobře
 - Popis jedné instance se perturbuje se stejnou pravděpodobností (např. pořadí definice hradel optimalizovaného obvodu)
 - Statistika měří varianci vyvolanou perturbací
- Vlastnosti pro randomizované algoritmy
 - Algoritmus přináší také varianci do experimentu
 - Analýza poskytuje průměrné hodnoty při předpokládaném (většinou chceme uniformní) rozdělení charakteristik vstupních instancí
 - Ladění heuristiky na konkrétní problém
 - White box fáze: ladíme na známých instancích
 - Black box fáze: evaluujeme nastavení na nových neznámých instancích

13 – Princip lokálních heuristik, pojem globálního a lokálního minima, obrana před uváznutím v lokálním minimu.

Další počtení zde [BI-ZI 2022](#) , fitness funkci se tam říká heuristika pro odhad ceny cesty. Otázka 26 tamtéž řeší v podstatě to samé, co tahle.

Princip lokálních heuristik

- Stavový prostor: graf
 - Uzly: stavy tvořené ohodnocením konfiguračních proměnných problému
 - Hrany: akce (aplikace operátoru na stav)
 - Okolí stavu: množina stavů dosažitelných jednou akcí
 - k -okolí stavu: množina stavů dosažitelných jednou až k akcemi
 - Může být cyklický, záleží na operátorech
 - Acyklický: snadné řízení, omezená délka cesty, typicky hladové algoritmy
 - Cyklický: komplikovanější řízení, typicky pokročilé heuristiky
 - Užitečné vlastnosti
 - Silná souvislost grafu
 - Symetrie vzdálenosti vrcholů (aspoň částečná)
- Pohyb stavovým prostorem
 - Úplná strategie: navštíví všechny stavy kromě těch, které jistě neobsahují řešení
 - Systematická strategie: navštíví každý stav nejvýše jednou
 - Nejhorší případ odpovídá hrubé síle
 - Naleznou (optimální) řešení, pokud existuje
- Lokální heuristika
 - Metoda prohledávání stavového prostoru
 - Prochází hrany a hledá optimální řešení pomocí fitness funkce
 - Fitness funkce: ideálně *metrika* kombinující optimalizační kritérium a měřítko kvality stavu
 - Výsledek heuristického prohledávání by neměl záviset na volbě počátečního stavu (toho se těžko dosahuje)
- Metoda best-first (hill-climbing, hladové prohledávání)
 - Vždy přechází do nejlepšího sousedního stavu
- Prořezávání stavového prostoru
 - Zahodíme podstrom určitých ohodnocení, pokud je jisté, že nevede k řešení lepšímu než dosud nejlepšímu navštívenému

Globální a lokální minimum

- Snažíme se najít stav s nejlepší hodnotou fitness funkce
 - Globální minimum: žádný jiný stav nemá lepší fitness
 - Lokální minimum: žádný sousední stav nemá lepší fitness
- Při prohledávání se algoritmus může dostat do lokálního minima, odkud nemůže pokračovat, protože v okolí je vše horší
 - Typický problém best-first metod

Obrana před uváznutím v lokálním minimu

- Prohledávání stavového prostoru lze provádět dvěma přístupy
 - Diverzifikace: velká ochota k akci zhoršující fitness, můžeme se ale dostat do lepší oblasti
 - Intenzifikace: malá ochota k akci zhoršující fitness, konvergujeme k finálnímu řešení (nemusí být globálně optimální)
- Řešení problému lokálního optima
 - Vyvážení intenzifikace a diverzifikace
 - Zahazování nalezených konfigurací (tabu search)
 - Generování zcela jiných konfigurací (mutace v genetice)
 - Náhodné přeskakování mezi stavy
- Kontrola kvality řešení: běhy s různými počátečními stavy končí ve stejném (globálním) minimu

14 – Princip genetických algoritmů, význam selekčního tlaku pro jejich funkci.

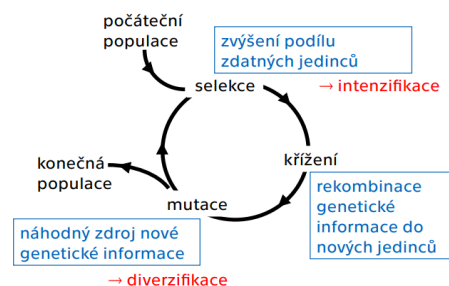
Další počtení zde [BI-ZI 2022](#), jsou tam fajn obrázky. Je to opět prakticky stejná otázka.

Princip genetických algoritmů

- Genetické algoritmy patří mezi lokální heuristické metody
- Řeší kombinatorické a optimalizační úlohy
- Inspirace evoluční biologií
- Princip šlechtění populace kandidátů na řešení, kvalitní jedinci se vybírají k další reprodukci s křížením a mutací
- Kompromis mezi diverzifikací a intenzifikací
- Terminologie
 - Genotyp: reprezentace řešení, jaké má parametry (dále příklady na bitovém vektoru)
 - Gen: parametr řešení
 - Fenotyp: konkrétní ohodnocení parametrů
 - Fitness: metrika kvality jedince
 - Jedinec: kandidující řešení složené z fenotypu a fitness
 - Populace: množina šlechtěných jedinců
 - Generace: čítač hlavního cyklu genetického algoritmu
- Fáze algoritmu
 - Inicializace: tvorba počáteční populace
 - Selektce: výběr rodičů další generace na základě jejich fitness
 - Reprodukce: tvorba potomků z rodičů pomocí křížení a mutace
 - Mutace
 - Drobná změna genotypu jedince
 - Brání uváznutí v lokálním optimu
 - Křížení: výměna informací mezi jedinci, konstrukce potomků
 - Náhrada: náhrada některých nebo všech jedinců novými potomky

Varianty genetických algoritmů

- Genetický algoritmus
 - Genotyp: bitový vektor
 - Inicializace: náhodná/podle heuristiky
 - Selektce
 - Ruletová
 - Pravděpodobnost výběru přímo úměrná fitness
 - Lze nejprve přeskálovat pro úpravu selekčního tlaku
 - Turnajová
 - Náhodný los několika jedinců a výběr nejlepšího
 - Nezávisí tak na konkrétních hodnotách fitness
 - Mutace: flip každého bitu s nějakou pravděpodobností
 - Křížení



- n -bodové: genomy se rozdělí na stejných n místech a části se křížově prohodí
- Uniformní: náhodné promíchání bitů, extrémní případ n -bodového, kde $n = \text{počet bitů} - 1$ a prohazujeme náhodně
- Náhrada: obvykle novou generaci tvoří potomci
- Genetické programování
 - Genotyp
 - Graf operací výpočtu
 - V listech vstupy, ve vnitřních uzlech operace
 - Flexibilní - struktura a velikost řešení je předmětem optimalizace
 - Inicializace: generování stromu požadované hloubky
 - Mutace: změny podstromů, odřezávání, permutace potomků uzlů
 - Křížení: výměna podstromů
- Evoluční strategie
 - Genotyp: vektory reálných čísel a směr mutací
 - Selektce: μ rodičů, λ potomků
 - $(\mu + \lambda)$: výběr μ nejlepších z rodičů i potomků
 - (μ, λ) : výběr μ nejlepších pouze z potomků
 - Mutace: malá změna složek vektoru
 - Gaussovská: přičtení vzorku z normálního rozdělení
 - Metaevoluce: mutuje se i rozdělení, jedinci si drží std. odchylky, které se mění podle úspěšnosti mutace (fit = menší odchylka)
 - Křížení
 - Uniformní: po složkách
 - Diskrétní: složka vektoru se převezme od jednoho z rodičů
 - Aritmetické: složka je průměrem hodnot rodičů
 - Parametr ρ : určuje počet rodičů potomka
 - $\rho = 1$: standardní selektce podle evoluční strategie
 - $\rho = 2$: křížení podobné genetickému algoritmu
 - $\rho = \mu$: na tvorbě potomka se podílí celá populace
- Evoluční programování
 - Genotyp: automat, neuronová síť či program s fixní strukturou
 - Selektce: turnajová
 - Mutace: přidání a odebrání stavu, symbolu, změna přechodu
 - Křížení: není

Význam selekčního tlaku pro jejich funkci

- Selektční tlak: jak důležitá je fitness jedince pro jeho přežití
- Mění poměr intenzifikace/diverzifikace v algoritmu
- Velký selekční tlak: nebezpečí degenerace, uváznutí v lokálním minimu
- Malý selekční tlak: pomalá konvergence
- Pokud šum vzniklý mutací převáží nad konvergencí, nastává divergence, zhoršování kvality populace

15 – Princip simulovaného ochlazování, význam parametrů a způsoby jejich řízení.

Princip simulovaného ochlazování

- Lokální heuristická metoda prohledávání stavového prostoru
- Inspirace pozvolným ochlazováním taveniny (simulované žíhání)
- Rozšiřuje hill-climbing o parametr teploty t
- Upravený algoritmus hill-climbingu
 - Kontrola teploty, je-li příliš nízká, konec
 - Postup do nového stavu
 - Je-li lepší, přijme se vždy
 - Je-li horší, přijme se s pravděpodobností závislou na
 - Míře zhoršení (rozdíl staré a nové fitness)
 - Aktuální teplotě t : čím vyšší t , tím vyšší pravděpodobnost
 - Ochlazení

Význam parametrů a způsoby jejich řízení

- Počáteční teplota: ovlivňuje celkovou dobu běhu
- Aktuální teplota
 - Vysoká: diverzifikace
 - Nízká: intenzifikace
- Koncová teplota
 - Pevná hranice
 - Detekce stagnace - změn (k lepšímu) je méně než pevná mez
- Délka ekvilibria N : počet evaluovaných sousedů k postupu
 - Pevný počet ve vztahu k velikosti instance
 - Počet závislý na teplotě (souvisí s chlazením)
- Chlazení
 - Pevný koeficient $\alpha < 1$
 - Závislý na počtu přijatelných sousedů: pokud mnoho, chladí rychleji
- Fitness: může být vhodné normalizovat na stejný rozsah pro všechny instance, aby pravděpodobnost přijetí horšího stavu byla stejně proporcionální k teplotě
- Pokud spadáme do lokálního minima, je třeba zvýšit diverzifikaci
 - Vyšší počáteční teplota
 - Pomalejší chlazení
 - Delší ekvilibrium
 - Změna fitness funkce (relaxace - přirážka k fitness za omezující kritéria)
- Pokud je výpočet příliš pomalý, naopak zvyšujeme intenzifikaci

16 – Výkonnostní měřítka paralelních algoritmů, PRAM model, APRAM model, škálovatelnost.

Detailní výpisky  NI-PDP 2024

Výkonnostní měřítka paralelních algoritmů

- Tak jako u sekvenčních algoritmů se snažíme zjistit, jak rychlost závisí na vstupu
- V problému je nějaká míra paralelismu, kterou lze zrychlit přidáním procesorů
- Od určitého počtu procesorů je paralelismus vyčerpán a přidávání dalších může výpočet naopak zpomalit kvůli nákladům na režii
- Základní definice (pro problém K)
 - n : velikost instance (dat)
 - p : počet procesorů
 - $SL(n)$: spodní mez sekvenční složitosti (nelze sekvenčně řešit rychleji)
 - $SU(n)$: horní mez sek. slož. (nejhorší čas nejlepšího známého algoritmu)

- Paralelní čas $T(n, p)$
 - Celkový čas strávený výpočtem
 - Součet náročnosti výpočtu a režie

- Paralelní cena $C(n, p)$

$$C(n, p) = p \times T(n, p)$$

- Součet časů strávených ve výpočtu všemi procesory
- Obvykle jsou totiž procesory alokovány po celou dobu výpočtu
- Cenová optimalita: procesory jsou plně využity, optimum je sekvenční alg.

$$C(n, p) = \Theta(SU(n))$$

- Paralelní zrychlení $S(n, p)$

$$S(n, p) = \frac{SU(n)}{T(n, p)}$$

- Poměr času sekvenčního a paralelního běhu, chceme následovně
- Lineární zrychlení: paralelní je tolikrát rychlejší, kolik máme procesorů

$$S(n, p) = \Theta(p)$$

- Superlineární zrychlení: např. pokud ušetříme swapování paměti, tedy obejdeme znevýhodnění HW podmínek sekvenčního algoritmu

- Paralelní efektivnost $E(n, p)$

$$E(n, p) = \frac{SU(n)}{C(n, p)} \leq 1$$

- Poměr ceny sekvenčního a paralelního řešení, relativní vytížení procesorů
- Konstantní efektivnost: vytížení procesoru nikdy neklesne pod $x\%$

$$E(n, p) = \Omega(1)$$

- Paralelní optimalita

- Lineární zrychlení \Leftrightarrow cenová optimalita \Leftrightarrow konstantní efektivnost

PRAM model, APRAM model

- RAM (Random Access Machine)
 - Základní abstrakce Von Neumannova počítače
 - Instrukce, indexovatelná paměť s $O(1)$ přístupem, společná s daty
- PRAM (Parallel RAM)
 - Více procesorů RAM, navíc sdílená paměť
 - Synchronní režim operace: všichni najednou provádějí jednu ze tří operací, operace trvají 1 jednotku času
 - Čtení
 - Lokální výpočet
 - Zápis
 - Ošetření konfliktů při zápisu do sdílené paměti
 - EREW (Exclusive Read Exclusive Write)
 - CREW (Concurrent Read Exclusive Write)
 - CRCW (Concurrent Read Concurrent Write): zápisy stejné buňky
 - Priority: procesory mají pevnou prioritu zápisu
 - Arbitrary: zapisuje náhodný, algoritmus s tím musí počítat
 - Common: všichni musí zapisovat stejnou hodnotu / chyba
- APRAM (Asynchronous PRAM)
 - Procesory operují asynchronně
 - Synchronizace explicitně pomocí bariér, kde je to třeba
 - Výpočet je posloupnost globálních fází oddělených bariérami
 - Centrální čítač: $O(dp)$, inkrementace každým procesem
 - Binární redukční strom: $O(d \log p)$, čekání uzlu na potomky
 - Doba přístupu k paměti není konstantní
 - Pokud v jedné globální fázi jeden zapisuje, ostatní tam nesmí přistupovat

Škálovatelnost

- Schopnost paralelního algoritmu udržet paralelní optimalitu při změně množství dat nebo procesorů
 - Silná škálovatelnost: kolik procesorů dává smysl pro náš problém
 - Měří rychlost poklesu efektivnosti při rostoucím p a konstantním n
 - Amdahlův zákon dává mez
 - Slabá škálovatelnost: jak velký chceme problém, abychom využili procesory
 - Jak se musí měnit n , aby při rostoucím p zůstala efektivnost konstantní
- Amdahlův zákon saturace paralelizace
 - Každý sekvenční algoritmus se skládá z
 - Inherentně sekvenčního podílu $0 < f_s < 1$
 - Paralelizovatelného podílu $1 - f_s$
 - Horní mez zrychlení při (libovolně) p vláknech je

$$S(n, p) = \frac{T_A(n)}{f_s \cdot T_A(n) + \frac{1-f_s}{p} \cdot T_A(n)} = \frac{1}{f_s + \frac{1-f_s}{p}} \leq \frac{1}{f_s}$$

- Gustafsonův zákon
 - S rostoucím počtem procesorů p máme navyšovat i velikost problému n
 - Sekvenční část pak trvá konstantně nezávisle na p , paralelizovatelná lineárně škáluje s p

$$S(n, p) = \frac{t_{\text{seq}} + t_{\text{par}}(n, 1)}{t_{\text{seq}} + t_{\text{par}}(n, p)}$$

- S rostoucí velikostí problému n se zrychlení limitně blíží p
- Izoefektivní funkce
 - Chceme zajistit konstantní efektivnost $0 < E_0 < 1$
 - Asymptoticky minimální

$\psi_1(p)$ je **asymptoticky minimální** funkce taková, že

$$\forall n_p = \Omega(\psi_1(p)) : E(n_p, p) \geq E_0.$$
 - Pro počet procesorů dává dolní mez na velikost problému
 - Asymptoticky maximální

$\psi_2(n)$ je **asymptoticky maximální** funkce taková, že

$$\forall p_n = O(\psi_2(n)) : E(n, p_n) \geq E_0.$$
 - Pro velikost problému dává dolní mez na počet procesorů
 - Min/max souvisí se slabou/silnou škálovatelností

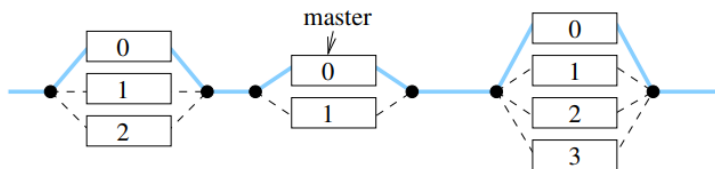
17 – Programování nad sdílenou pamětí, programový model OpenMP, datový a funkční paralelismus, synchronizace vláken, vícevláknové algoritmy (násobení polynomů, násobení matic a vektorů, řazení).

Programování nad sdílenou pamětí

- APRAM model, procesory mají jak každý svoji, tak sdílenou paměť
- Problém zejména při paralelním zápisu
- Falešné sdílení: překrývání zapisovaného prostoru v cache

Programový model OpenMP

- Paralelismus na úrovni vláken (POSIX threads) nad virtuálně sdílenou pamětí
- OpenMP: vysokoúrovňové API pro vláknový paralelismus
 - Direktivy překladače `#pragma omp direktiva klauzule...`
 - Globální proměnné (počet vláken apod.)
 - Knihovna runtime operací
- Fork-join model



- Master: řídící vlákno
- Thread pool: množina recyklovatelných vláken pro zlehčení režie
- Paralelní region: pouze v něm se využívají vlákna z thread poolu
- OpenMP neřeší distribuovanou paměť, deadlocky, race conditions, ...
- Chování proměnných v paralelním regionu
 - Shared: sdílený skalár, bez zámku
 - Private: každé vlákno má svou neinicizovanou instanci
 - Firstprivate: každé vlákno dostane inicializovanou instanci
 - Threadprivate: přežívá všechny paralelní regiony pro každé vlákno
 - Reduction: redukce po skončení paralelního regionu
- Nastavení paralelizace
 - Paralelizace pouze při splnění klauzule `if(cond)`
 - Zpracování pomocí `num_threads(expr)` vláken

Datový a funkční paralelismus

- Datový paralelismus `#pragma omp parallel for klauzule...`
 - Pro datově nezávislé for cykly

- Scheduling, přiřazování iterací vláknům
 - Static
 - Po sobě jdoucí bloky podle chunk-size, jinak rovnoměrně
 - Malá režie, ale nerovnoměrná zátěž při různé složitosti úloh
 - Dynamic
 - Přiřazování až čekajícímu vláknu, chunk-size default 1
 - Vyšší režie s menším chunk-size, zajišťuje ale rovnoměrnost
 - Guided
 - Dynamicky přiřazuje p -tinu dosud nehotové části, minimálně však chunk-size s default hodnotou 1
 - Vyšší režie s menším chunk-size, ale rovnoměrné zatížení pokud složitost úloh postupně roste
 - Ordered: iterace v sekvenčním pořadí
 - Collapse: zkolabování víceúrovňového cyklu do jediného
- Funkční paralelismus (task paralelismus)


```
#pragma omp parallel {
  #pragma omp single {
    #pragma omp task klauzule... {}
  }
}
```

 - Pro algoritmy *rozděl a panuj*
 - V paralelním regionu se pomocí single spustí fronta producenta
 - Pomocí task se vytváří konzumenti
 - Lze zadat threshold jako podmínku, aby se malá úloha dál neštěpila - režijní náklady by zhoršily efektivitu

Synchronizace vláken

- Direktivy pro synchronizace
 - Barrier: čekání na ostatní vlákna v paralelní oblasti
 - Master: provádění bloku pouze hlavním vláknem
 - Single: provádění bloku jediným libovolným vláknem
 - Critical: kritická sekce pro přístup ke sdíleným prostředkům
 - Atomic: atomická read-modify-write operace
 - Taskwait: synchronizace po spuštění několika task (pro rekurzi, fční par.)

Vícevláknové algoritmy

- Násobení polynomů
 - Sekvenční: dva vnořené cykly, na místo dle součtu stupňů přičítáme součin koeficientů

```
int A[m+1], B[n+1], C[n+m+1];
for (int k = 0; k <= m + n, k++)
    C[k] = 0;
for (int i = 0; i <= m; i++)
    for (int j = 0; j <= n; j++)
        C[i+j] += A[i]*B[j];
```

- Paralelizace vnějšího cyklu
 - Ve vnitřním cyklu atomický update k přičtení koeficientů
 - Vlákna se tvoří jednou, režie jen kvůli atomickému updatu
- Paralelizace vnitřního cyklu
 - Opět atomický update, přičítání koeficientů se překrývá
 - Režie navíc kvůli opakovanému tvoření vláken a synchronizaci
- Paralelizace vnějšího cyklu s disjunktními oblastmi: nejlepší
 - Je nutný jiný než statický scheduler, jinak prostřední vlákna budou více zatížena
 - Dynamický má zbytečnou režii, lze to předpočítat

k		B _j
0		0.0
1		0.1 + 1.0
2		0.2 + 1.1 + 2.0
3		0.3 + 1.2 + 2.1 + 3.0
4		0.4 + 1.3 + 2.2 + 3.1 + 4.0
5		0.5 + 1.4 + 2.3 + 3.2 + 4.1 + 5.0
6		1.5 + 2.4 + 3.3 + 4.2 + 5.1
7		2.5 + 3.4 + 4.3 + 5.2
8		3.5 + 4.4 + 5.3
9		4.5 + 5.4
10		5.5

- Násobení hustých matic
 - Sekvenční: tři vnořené cykly, v nejvnitřnějším akumulujeme součiny

```
float A[n][n], B[n][n], C[n][n];
for (int i = 0; i < n; i++)
    for (int j = 0; j < n; j++) { //n^2 iterations
        float s = 0.0;
        for (int k = 0; k < n; k++) //scalar product
            s += A[i][k]*B[k][j];
        C[i][j] = s; }
```

- Paralelizace vnějšího cyklu: nejlepší
 - Rovnoměrné a efektivní se statickým schedule
- Paralelizace prostředního cyklu
 - Stále disjunktní, ale $n \times$ víc synchronizace
- Paralelizace prostředního cyklu s chunk-size=1
 - Moc režie na synchronizaci
 - Falešné sdílení

- Násobení řídké matice vektorem

- Souřadnicový formát COO: paralelní pole indexů řádků, sloupců a hodnot
 - Sekvenční: nepřímá indexovace vektoru při procházení polí COO

```
struct A; //sparse matrix in the COO format
float x[n]; //array representing the input vector x
float y[n]; //array representing the output vector y=Ax
for (i = 0; i < n; i++)
    y[i] = 0.0;
for (k = 0; k < N; k++)
    y[A.RowInd[k]] += A.Elems[k] * x[A.ColInd[k]];
```


18 – Programování nad distribuovanou pamětí, programový model MPI (vícevláknové procesy, komunikátory, 2-bodové blokující a neblokující komunikační operace, kolektivní operace), paralelní násobení hustých matic, paralelní mocninná metoda.

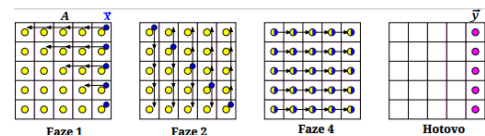
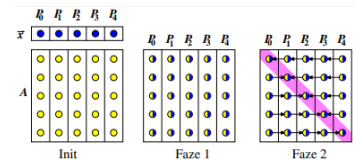
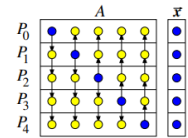
- Výpočet na více procesorech přináší problém rozdělení úloh a synchronizace

Programový model MPI

- MPI (Message Passing Interface)
 - Komunikace procesů, OpenMP pak rozděluje práci mezi jednotlivá vlákna
 - Knihovní funkce pro C, C++ a Fortran podobně jako OpenMP
 - Mnoho implementací standardu pro různé systémy
- Inicializace a volba spolupráce s vlákny (různé implementace umí různé varianty)
 - Single: pouze MPI procesy bez vláken
 - Funneled: vícevláknové procesy, MPI volá jen master vlákno
 - Serialized: MPI volá v jednu chvíli jedno vlákno
 - Multiple: jakékoliv vlákno volá MPI (všeportový model)
- Komunikátory
 - Skupiny procesů pro komunikaci, World obsahuje všechny
 - Interkomunikátor dovoluje posílat zprávy napříč skupinami
- Dvoubodová komunikace
 - MPI_Send: pointer na hodnotu, cílový proces, status a komunikátor
 - MPI_Recv: pointer na paměť, zdrojový proces, zpracování statutu, kom.
 - Je třeba uvést datový typ konstantou z MPI nebo vytvořit vlastní
 - Blokující operace
 - Standardní: nelokální, čeká se na kopii zprávy do bufferu či na odeslání podle implementace MPI (zaručuje přenositelnost kódu)
 - Buffered: čeká se jen na kopii, příjemce nemusí existovat (lok. op.)
 - Synchronní: čeká se na zahájení přijímání druhou stranou a přijetí
 - Ready: čeká se na odeslání, příjemce musí být již připraven
 - Neblokující operace
 - Po zavolání Send se buffer nekopíruje, musí být zachován
 - Na odeslání se lze dotazovat pomocí Wait
- Kolektivní komunikace
 - Broadcast: one-to-all od mastera všem
 - Gather: all-to-one od všech masterovi
 - Allgather: přijímají všichni (všichni pak mají všechno)
 - Scatter: rozdělení dat od mastera všem včetně mastera samotného
 - Alltoall: symetrická distribuce všichni všem (princip maticové transpozice)

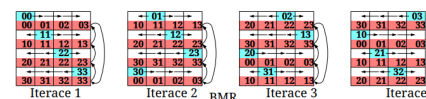
Paralelní násobení hustých matic vektorem

- Matice se mapuje mezi procesy po pruzích (řádky/sloupce) nebo jako šachovnice
 - Sekvenčně a/nebo blokově po více pruzích či obdélnících
- Řádkové mapování
 - Všichni pošlou svůj prvek vektoru ostatním
 - Každý pak počítá jeden prvek výsledného vektoru
 - Konstantní škálovatelnost, rychlost závisí na broadcastu
- Sloupčové mapování
 - Každý spočítá příspěvek svého sloupce do všech prvků vektoru
 - Poté redukce se sčítáním na diagonálu
 - Stejně parametry jako řádkové mapování
- Šachovnicové mapování
 - Pravý krajní blok pošle svůj subvektor diagonálnímu bloku
 - Diagonály rozešlou subvektor do celého sloupce
 - Každý lokálně pronásobí submatici a subvektor
 - Nakonec řádková redukce



Paralelní násobení hustých matic maticí

- Cannonův algoritmus
 - Jednu matici cyklicky posouváme v řádcích, druhou ve sloupcích
 - Optimální komunikace ve 2D toroidu
 - Paměťově optimální, data se neduplikují
- Foxův algoritmus (viz [příklad matic 2x2](#))
 - První matici rozepíráme do celého řádku
 - Přičteme součin do cílové submatice
 - Submatice se cyklicky rotují o řádek nahoru
 - Podobně složité jako Cannon

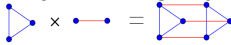


Paralelní mocninná metoda

- Mocninná metoda: výpočet největšího vlastního čísla a příslušného vl. vektoru
 - Násobíme matici nějak inicializovaným vektorem
 - Vypočteme normu výsledného vektoru
 - Výsledek znormalizujeme a opakujeme násobení
 - Norma konverguje k vl. číslu a vektor k příslušnému vl. vektoru
- Pro řídkou matici a vektor, tedy podobně jako v otázce 17
- Řádkové mapování: po pronásobení Gather, výpočet normy a distribuce normalizovaného vektoru zpět
- Libovolné mapování: po pronásobení některé části Gather se součtovou redukcí
- Šachovnicové mapování: sloupčová distribuce, poté řádková redukce do diagonály, normalizace a opět distribuce

19 – Přímé ortogonální a hyperkubické propojovací sítě paralelních počítačů (definice, vlastnosti, vnořování). Základní pojmy

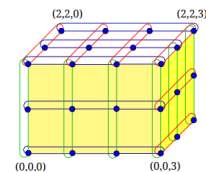
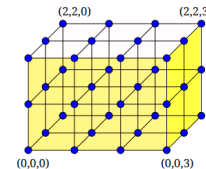
Základní pojmy - "základní", lol

- Topologie G_n
 - Množina grafů se strukturou a velikostí definovanou parametrem
 - n : velikost dimenze
- Škálovatelnost topologie
 - Inkrementální: definovaná pro všechna $n \in \mathbf{N}$
 - Částečná: definovaná pro nekonečnou podmnožinu \mathbf{N}
- Hustota topologie
 - Řídká: stupně uzlů omezeny konstantou
 - Hustá: stupně uzlů jsou rostoucí funkcí n (např. hyperkrychle)
- Vzdálenosti v grafu
 - Excentricita uzlu: maximální vzdálenost do všech ostatních uzlů
 - Průměr grafu: maximální excentricita
 - Poloměr grafu: minimální excentricita
 - Uzlově a hranově disjunktní cesty: zajímají nás pro toleranci výpadků
- Struktura
 - Regularita: všechny uzly mají stejný stupeň
 - Hierarchická rekurzivita: instance menších dimenzí jsou podgrafy větších
 - Kartézský součin: vytvoří ortogonální graf 
 - Uzlová symetrie
 - Automorfismus mezi každými dvěma uzly
 - Uzlově symetrický graf je regulární a průměr se rovná poloměru
 - Hamiltonovský graf: lze navštívit každý uzel právě jednou a vrátit se
- Souvislost
 - Uzlový/hranový řez: množina, jejíž odstranění naruší souvislost grafu
 - Uzlová/hranová souvislost: minimální počet uzlů/hran potřebný pro narušení souvislosti grafu
 - Optimální souvislost: uzlová = hranová
- Bipartitnost a bisekce
 - Bipartitní graf: existuje obarvení vrcholů dvěma barvami tak, že vrcholy spojené hranou mají rozdílnou barvu
 - Vyvážený bipartitní graf: počty vrcholů obou barev se rovnají
 - Bisekční šířka: velikost nejmenšího hranového řezu na dvě poloviny uzlů
- Praktické požadavky
 - Regularita: levné univerzální směrovače
 - Malý průměr a průměrná vzdálenost: rychlá komunikace
 - Spodní mez průměru n -uzlové řídké sítě je $\Omega(\log n)$
 - Uzlová symetrie a hierarchická rekurzivita: jednoduchý návrh algoritmů
 - Vysoká souvislost: redundance pro toleranci výpadků a přetížení, přenos velkých zpráv paralelně po více cestách

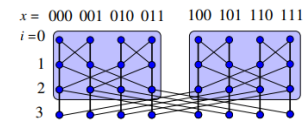
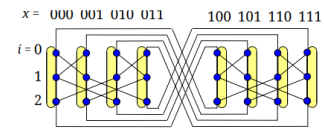
- Bisekční šířka
 - Velká pro rychlost přenosu pro algoritmy rozděl a panuj
 - Malá pro méně nutných spojů a tudíž nižší cenu HW
- Vnořitelnost (do) jiných topologií: efektivní provedení algoritmů, které implementují jinou topologii
- Podpora pro směrování a kolektivní komunikační operace: broadcast...

Přímé ortogonální a hyperkubické propojovací sítě

- n -rozměrná mřížka
 - Základní vlastnosti
 - Hierarchicky rekurzivní (konstrukce kartézským součinem jednorozměrných řetězků)
 - Není regulární, proto ani uzlově symetrická
 - Bipartitní, ne vždy vyvážená
 - Obsahuje hamiltonovskou kružnici při aspoň jedné sudé dimenzi
 - Manhattanská vzdálenost uzlů (metrika): pohyb mezi uzly po ortogonálních hranách
 - V praxi nejčastěji 2D (čipy) a 3D (server rack)
 - Směrování posunem seřazeným po dimenzích
- n -rozměrný toroid
 - Zabalená mřížka, n -rozměrná kružnice
 - Základní vlastnosti
 - Není hierarchicky rekurzivní
 - Uzlově symetrický
 - Bipartitní pokud jsou všechny rozměry sudé
 - Průměrná vzdálenost poloviční oproti mřížce
 - Bisekční šířka dvojnásobná oproti mřížce (řez kružnice vs. řetězu)
 - Obsahuje hamiltonovskou kružnici vždy
 - Manhattanská vzdálenost uzlů
 - Jedna z nejúspěšnějších komerčních topologií
- Binární hyperkrychle
 - Základní vlastnosti
 - Hustá: stupeň uzlů je logaritmický - náročná na HW
 - Hierarchicky rekurzivní
 - Uzlově symetrická
 - Bipartitní
 - Optimálně souvislá (uzlová = hranová)
 - Hrany mezi uzly o hammingově vzdálenosti 1 (např. 1000-1001)
 - Základní testovací topologie díky své optimální souvislosti
 - Má ale logaritmický stupeň uzlů a je škálovatelná jen po mocninách 2
 - Směrování seřazením cest lexikograficky (e-cube)
- Úvod k motýlkům
 - Řídké grafy
 - Hypekrychle, kde je každý uzel rozvinutý do více uzlů
 - Konstantní stupeň uzlů a logaritmický průměr
 - Horší škálovatelnost než hyperkrychle (pro ještě méně n než mocniny 2)
 - Přirozená topologie pro mnoho paralelních algoritmů

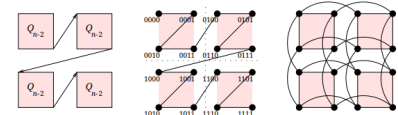


- Zabalený motýlek
 - Základní vlastnosti
 - Není hierarchicky rekurzivní - kružnici nerozložíš
 - Uzlově symetrický
 - Vyvážený bipartitní pro sudá n
 - Hamiltonovský
 - Konstantní stupeň uzlu 4
 - Každý uzel má dvě hrany ve vlastní kružnici a dvě hrany do vedlejších kružnic
 - Optimální průměr a průměrné vzdálenosti: nejhůř projdeme n hyperkubických hran a $\lfloor n/2 \rfloor$ na druhou stranu kružnice
- Obyčejný motýlek
 - Základní vlastnosti
 - Hierarchicky rekurzivní
 - Není regulární, proto ani uzlově symetrický
 - Bipartitní
 - Není hamiltonovský
 - Vznikne ze zabaleného motýlka rozdělením 0. vrcholu kružnic na dva uzly
 - Směrování e-cube (existuje jediná nejkratší cesta mezi dvěma uzly)
 - Vhodný jako levná minimální permutační síť



Vnořování

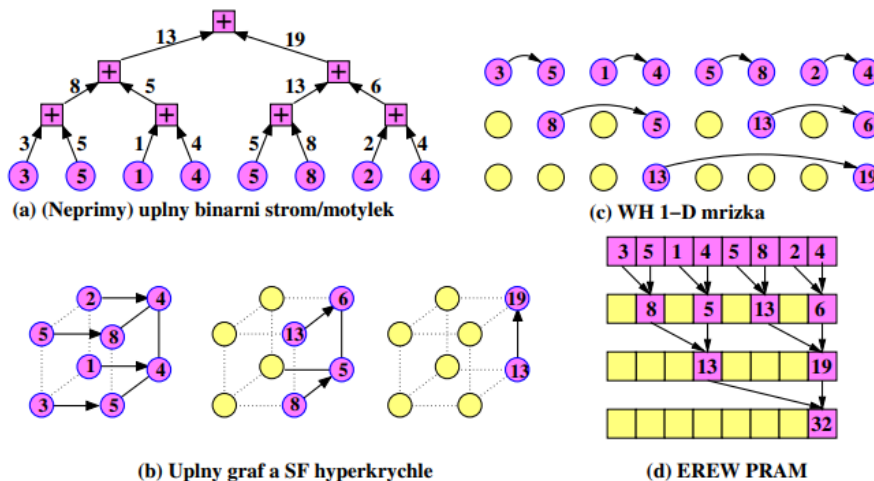
- Statické vnořování
 - Uspořádaná dvojice zobrazení procesů a hran na procesory a fyz. spoje
 - Graf procesů chceme namapovat na fyzickou topologii pro maximalizaci efektivity komunikace (např. často komunikující procesy jsou blízko)
 - NP-hard problém (kombinatorika)
- Měřítka kvality
 - Maximální zatížení cílového uzlu: kolik max. běží procesů na procesoru
 - Expanze vnoření: poměr počtu procesorů a procesů
 - Větší než 1: plýtvání procesory, zbytečně drahé
 - Menší než 1: přetěžování procesorů, pomalé
 - Maximální dilatace: max. délka fyz. cesty, na kterou se namapuje hrana
 - Maximální zahlcení cílové hrany: kolik hran využívá jeden fyzický spoj
- Kvaziizometrická topologie: statické vnoření tam i zpět takové, že jsou měřítka kvality omezená konstantou
 - Vnořený algoritmus tak má nejvýše konstantní zpoždění
 - Asymptoticky je pak výpočetně ekvivalentní
- Vnoření hyperkrychle do nízkorozměrných mřížek
 - Obvykle se programuje na hyperkrychli
 - Vnořujeme do 2D mřížky
 - Mortonova křivka: využíváme hierarchickou rekurzivitu obou topologií
 - Hyperkrychli dělíme na 4 podkrychle, mřížku na 4 kvadranty
 - Konec při dosažení mřížky 2×2
 - Poté tvorba lexikografické cesty ve tvaru Z (fraktální křivka)
 - Lépe zachová blízkost než Karnaugova/Svobodova mapa



20 – Paralelní algoritmy pro redukci, prefixový součet a segmentový prefixový součet na PRAM, v ortogonálních, hyperkubických a obecných topologiích, aplikace.

Redukce

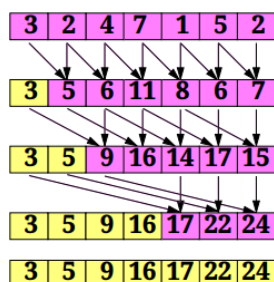
- All-to-one redukce: pole n prvků chceme redukovat pomocí asociativní (případně i komutativní) operace na skalár
- Logaritmický čas, dobrá škálovatelnost (rychleji to nejde)
- Optimální implementace na hyperkubické síti (*normální hyperkub. algoritmus*)
- Implementace



- Hyperkrychle: předávání po směrech postupně (binom. kostra)
- Strom/motýlek: redukce z listů ke kořeni
- Pokud se vstupní pole nemapuje na procesory v pořadí jejich indexů, operace musí být komutativní

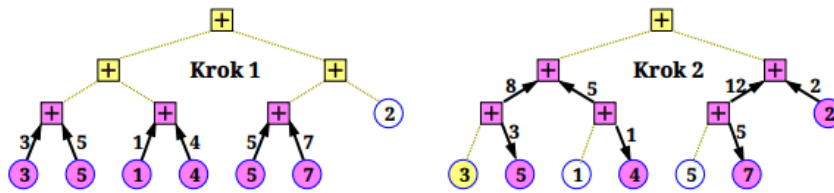
Prefixový součet

- Pole chceme sečíst tak, aby každá buňka obsahovala součet předchozích buněk a sebe samotné
- PRAM



- Logaritmický čas, v k -tém kroku seřazeno 2^k prvků
- In-place

- Nepřímý strom či motýlek



- Nepřímý strom: výpočet jen v listech, vnitřní uzly pouze redukuje
 - Logaritmický čas
- Přímý strom
 - Přímý strom: všechny uzly jsou výpočetní
 - Vyžaduje linearizaci stromu - postorder je pak podobné nepřímému
- Hyperkrychle
 - Postupně podle os v lexikografickém pořadí
 - Kumulujeme pak pouze čísla pocházející z nižších uzlů
- Mřížky
 - Store-and-forward switching
 - Zpráva po paketech, switche mají buffery, libovolná cesta
 - Dle linearizace, např. pro 2D po řádcích: doprava, pak poslední sloupec dolů, pak zpět propagace z posledního sloupce do řádků doleva
 - Wormhole switching
 - Zpráva po flitech, okamžité přeposlání, všechny flity za sebou stejnou cestou, ta je blokována
 - Simulace binárního stromu, propagace vpřed a pak zpětné doplňování
- Libovolný souvislý řádkový graf
 - Nalezení souvislé kostry pomocí BFS, poté postorder stromový přístup
 - $O(\text{diam}(G))$ kroků

Aplikace prefixového součtu

- Packing problem
 - Problém výpočtu pořadí procesu v podmnožině (označené 1)
 - Nad těmito procesy se provede prefixový součet, pak zná každý své pořadí
- RadixSort
 - Řazení v lexikografickém pořadí od nejnižšího řádu po nejvyšší
- Sčítáčka s predikcí přenosu
 - Paralelně se sečtou sčítance X a Y , výsledkem je vektor predikce B
 - $0 + 0 = \text{stop}$: určitě bez přenosu
 - $0 + 1$ nebo $1 + 0 = \text{propagate}$: možná s přenosem
 - $1 + 1 = \text{generate}$: určitě s přenosem
 - Vektor B doplníme zprava znakem s
 - Prefixový součet nad B
 - Převedeme vektor B na číslíčko C podle pravidla $g = 1$, jinak 0
 - Paralelně sečteme C s původními sčítanci X a Y
 - Následně prefixový součet nad řetězcem s, p, g

Segmentový prefixový součet

- Pole rozdělené do segmentů sčítáme kumulativně izolovaně v rámci segmentu
- Hranice segmentu $|$ je součástí binární operace, která jej zachovává ($a + |b = |b$)
 - Taková operace zachovává asociativitu původní operace
- Na nepřímém stromu vypadá stejně jako prefixový součet

Aplikace segmentového prefixového součtu

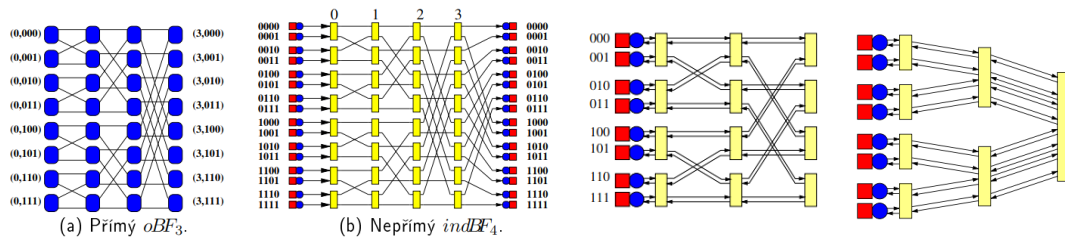
- Segmentový QuickSort
 - Pro manipulaci s daty stačí pouze segmentový prefixový součet
 - Proto rovnoměrná škálovatelnost, implementace nad distribuovanou i sdílenou pamětí
 - Logaritmická složitost pro náhodné vstupní pole
 - Vstup se rozdělí rovnoměrně podle procesů
 - Rekurze začíná globálním segmentem
 - Proces náhodně volí rozdělovač segmentu ve své části vstupu
 - Pivot je první prvek segmentu
 - Multicast pivota sousedům, se kterým proces sdílí segment
 - Třikrát zhušťování (postupně pro části $<$, $>$, $=$ pivot): segmentovým prefixovým součtem získáme nové indexy prvků
 - Podle získaných indexů pole permutujeme

Lád'a za odměnu pro ty, kdo to dočetli až sem



Zjistil jsem, že to není potřeba, ale bylo mi to líto smazat, takže tady je zapomněnka:

- Přímý (obyčejný), nepřímý, obousměrný motýlek a tlustý strom



- Nativní topologie pro normální hyperkubické algoritmy: v každém paralelním kroku se použijí pouze hrany jedné dimenze hyperkrychle, postupně se po krocích používají následující hrany
- Přímý motýlek
 - Uzel je zároveň výpočetní i komunikační
 - Směrování podle toho, zda chceme daný bit invertovat
- Nepřímý motýlek
 - Komunikační uzly jsou pouze přepínače (identita, inverze, bcast.)
 - Směrování podle hodnoty daného bitu - poslat nahoru či dolů
- Obousměrný motýlek
 - Obousměrné přepínače
 - Směrování nejprve jakkoliv ke kořeni nejvyššího společného podstromu, pak zpět jednoznačně
- Tlustý strom
 - Topologicky ekvivalentní s obousměrným motýlkem, počet linek k rodiči je součtem linek k potomkům
 - Redundantní spolehlivé směrování
 - Využití v datacentrech s různými parametry (arita, výška stromu) umožňuje pak inkrementální škálování

