STUDENTSKÝ ČASOPIS A KORESPONDENČNÍ SEMINÁŘ

Ročník XXXII

Číslo 6

M&M

MATEMATIKA FYZIKA INFORMATIKA

Uvnitř najdete několik témat a s nimi souvisejících úloh. Zamyslete se nad nimi a pošlete nám svá řešení. My vám je opravíme a ta nejzajímavější z nich otiskneme. Nejlepší řešitele zveme na podzim a na jaře na soustředění.

Milá čtenářko, milý čtenáři,

končí školní rok a s ním i 32. ročník korespondenčního semináře M&M. V tomto čísle už nenajdeš úlohy či problémy, za jejichž řešení lze získat body, ale stále obsahuje spoustu zajímavého čtení. Můžeš se podívat na vzorová řešení svého oblíbeného tématka a zjistit, jak k úlohám přistupovali Tví kolegové. Pokud bys radši nějaké odpočinkové čtení vhodné na prázdninové polehávání, doporučujeme článek z konfery z jarního soustředění ve Zlaté Koruně o výrobě papíru od Dr.$^{\mathrm{MM}}$ Filipa Dvořáka, Bc.$^{\mathrm{MM}}$ Filipa Gašparína a Bc.$^{\mathrm{MM}}$ Mikuláše Hořenka. Společně s článkem o zkoumání propojenosti smyslů, který byl vydán ve třetím čísle, byl vyhlášen nejlepším článkem ročníku, autoři se tak na příštím soustředění dočkají dortu. Gratulujeme! Nezapomeň, že i Ty můžeš sepsat článek o problému týkajícím se některého z tématek, o svém bádání na soustředění nebo o něčem úplně jiném, co Tě zrovna zaujme.

Na konci čísla najdete finální výsledkovou listinu tohoto ročníku. Celkovou vítězkou se stala Prof.$^{\mathrm{MM}}$ Julie Klementová se ziskem 149 bodů. Druhé místo obsadil Dr.$^{\mathrm{MM}}$ Jakub Thomitzek a třetí pozici těsně uhájil Dr.$^{\mathrm{MM}}$ Lukáš Koma. Nejlepší řešitelé získají věcné ceny a brzy obdrží pozvánky na podzimní soustředění, které se bude konat v termínu 17.–25. října. Pokud jsi pozvánku neobdržel/a, nezoufej a rychle se pusť do řešení 1. čísla nového ročníku. Na podzimní soustředění totiž již tradičně zveme i nejlepší řešitele z prvního deadlinu nového ročníku. Pokud máš tedy kamarády, kteří by soustředění rádi zažili, i oni ještě stále mají příležitost. A pokud při registraci do semináře uvedou, že se o M&M dozvěděli právě od Tebe, bude na Tebe čekat pěkná odměna ;)

Věříme, že Tě řešení semináře bavilo a hodně Ti toho dalo. Budeme moc rádi, pokud nám dáš prostřednictvím ankety[1] vědět, jak můžeme příští ročník udělat ještě lepší. Za všechny odpovědi děkujeme. A až Tě bude nudit poflakování na pláži nebo budeš mít nohy bolavé od celodenních túr, pohodlně se usaď a pusť se do čtení prvního čísla[2] nového, již 33. ročníku, našeho semináře :-)

Krásné léto plné zaslouženého odpočinku přejí

Tví organizátoři

Anketa

1. číslo 33. ročníku

[1] https://forms.office.com/e/gm7Bpw3tEX

[2] https://mam.mff.cuni.cz/media/cislo/pdf/33/33-1.pdf

Obsah

Řešení témat

Téma 1 – Grupy aneb Kterak matika k souměrnosti přišla

Díl 6: I grupy nekonečné musejí jednou skončit

Drazí čtenáři a řešitelé grupového tématka, dobrali jsme se k závěru. Nejen symetriemi mnohoúhelníků, elementárními částicemi a konečnými automaty proplouvá loď struktur zvaných grupy. Budeme rádi, když při plavbě nepřehlédnete ostrovů Lieových grup[3], kryptografie na eliptických křivkách[4] nebo hudební teorie množin[5]. Jistě vás oslní stejně jako nás: třeba i natolik, že nám o nich něco napíšete. Sbohem a šťastnou plavbu.

Řešení 3. dílu

Úloha 3.1

Zadání:
Dokažte, že

$$\mathrm{SU}(n) = (\{X \in \mathbb{C}^{n\times n} \mid X^*X = I \ \ a \ \ \det X = 1\}, \cdot, ^{-1}, I),$$

je opravdu grupa. Smíte použít kterýchkoliv vlastností matic a jejich determinantů z tématka Vektory a matice *z předchozího ročníku.*

Řešení od Bc.$^{\mathtt{MM}}$ Sabiny Bažantové:

Mějme libovolné prvky $A, B, C \in \mathrm{SU}(n)$. Čtyři grupové axiomy:

1. *Uzavřenost.* Musí platit $A^*A = I, B^*B = I, \det A = 1, \det B = 1$.

   Ověření unitárnosti součinu:

   $$(AB)^*(AB) = (B^*A^*)(AB) = B^*(A^*A)B = B^*IB = I,$$

   kde jsem využila vlastnosti $(AB)^* = B^*A^*$ a asociativity násobení matic. Determinant:

   $$\det(AB) = \det A \cdot \det B = 1 \cdot 1 = 1.$$

   Množina $\mathrm{SU}(n)$ je uzavřená na součin matic.

2. *Asociativita.* Plyne z asociativity násobení matic:

   $$A(BC) = (AB)C.$$

[3] Více v článku `https://en.wikipedia.org/wiki/Lie_theory`
[4] Více zde: `https://en.wikipedia.org/wiki/Elliptic-curve_cryptography`
[5] Vizte `https://en.wikipedia.org/wiki/Set_theory_(music)`

3. *Neutrální prvek.* Jednotková matice splňuje

$$I^* I = I, \det I = 1,$$

tedy $I \in \mathrm{SU}(n)$ a platí

$$AI = IA = A,$$

čili I je neutrální prvek.

4. *Inverzní prvek.* Z podmínky $A^*A = I$ plyne $A^* = A^{-1}$. Jedná se o charakteristickou vlastnost unitárních matic – inverz je roven hermitovsky sdružené matici.

   Unitárnost a determinant:

$$\begin{aligned}(A^{-1})^* A^{-1} &= (A^*)^* A^* = AA^* = I,\\ \det(A^{-1}) &= (\det A)^{-1} = 1^{-1} = 1,\end{aligned}$$

   takže $A^{-1} \in \mathrm{SU}(n)$.

Díky ověření všech čtyř grupových axiomů bylo dokázáno, že se opravdu jedná o grupu.

Úloha 3.2

Zadání:

V sekci o slabé nukleární síle jsme tvrdili, že $\mathrm{SU}(2)$ *lze generovat třemi rodinami komplexních* 2×2 *matic. Postupně vás provedeme důkazem tohoto tvrzení. Rozdělili jsme jej na části, abyste si mohli zvolit, kterým se hodláte věnovat. Za jednotlivé části lze získat body zvlášť.*

1. *Dokažte, že každá matice* $X \in \mathrm{SU}(2)$ *lze napsat ve tvaru*

$$X = \begin{pmatrix} a & b \\ -\overline{b} & \overline{a} \end{pmatrix}$$

   pro komplexní čísla $a, b \in \mathbb{C}$.

   (a) *[1b] Předpokládejte, že* X *je obecná* 2×2 *komplexní matice, čili*

$$X = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

   a rozepište, které všechny rovnosti pro čísla $a, b, c, d \in \mathbb{C}$ *plynou z podmínek* $X^*X = I$ *a* $\det X = 1$*. Nezapomeňte, že* $z\overline{z} = \overline{z}z = |z|^2$ *pro jakékoli* $z \in \mathbb{C}$.

   (b) *[1b] Upravte rovnost* $|a|^2 + |b|^2 = |c|^2 + |d|^2$ *plynoucí z* $X^*X = I$ *tak, aby obsahovala pouze čísla* b *a* c*. Doporučujeme využít rovností* $\det X = 1$ *a* $|c|^2 + |d|^2 = 1$.

(c) [2b] *Odvoďte, že* $c = -\overline{b}$. *Nabízíme tři pomůcky. Platí* $z + \overline{z} = 2\mathrm{Re}(z)$ *pro jakékoli* $z \in \mathbb{C}$, *kde* $\mathrm{Re}(z)$ *je reálná část čísla* z. *Když si už nebudete vědět rady, jak výraz dále upravit, doporučujeme čísla* b *a* c *přepsat do „algebraického" tvaru* $x + iy$. *Konečně,* $c = -\overline{b}$ *právě tehdy, když* $\mathrm{Re}(c) = -\mathrm{Re}(b)$ *a* $\mathrm{Im}(c) = \mathrm{Im}(b)$, *kde* Im *pro změnu značí část imaginární.*

(d) [1b] *Znovu využijte rovnosti* $X^*X = I$ *a výsledku z (c), abyste dokázali, že* $d = \overline{a}$.

2. [1b] *Vezměme úhly* $\alpha, \beta \in [0, 2\pi)$ *a uvažme dvě rodiny matic:*

$$Z(\alpha) = \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}, \quad Y(\beta) = \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix}.$$

Ukažte, že $Z(\alpha)$ *i* $Y(\beta)$ *jsou matice z* $\mathrm{SU}(2)$ *pro každé* α, β. *Argumentujte, že jejich součin je též matice z* $\mathrm{SU}(2)$.

3. [2b] *Ukažte, že každá matice* $X \in \mathrm{SU}(2)$ *lze zapsat jako součin tří matic z těchto rodin jako*

$$X = Z(\alpha)Y(\beta)Z(\gamma)$$

pro vhodné úhly $\alpha, \beta, \gamma \in [0, 2\pi)$. *Napovíme, že je užitečné použít tvar matice* X *z bodu 1. a též si uvědomit, že rovnost* $|a|^2 + |b|^2 = 1$ *klade důrazné omezení na možné hodnoty čísel* $a, b \in \mathbb{C}$.

4. [4b] *Dokázali byste geometricky interpretovat (s dostatečným komentářem) matice* $Z(\alpha)$ *a* $Y(\beta)$. *Co vlastně „dělají" s dvoudimensionálními komplexními vektory? Můžeme si posléze nějak představit i jejich součin?*

Řešení od Bc.$^{\mathrm{MM}}$ Patrika Fialy:

1. (a) Z determinantu matice X nám vyplyne následující vztah:

$$ad - bc = 1.$$

Nyní provedeme násobení X s X^*. Vzhledem k tomu, že víme, že je X^* inverzní maticí k X, taktéž víme, že je násobení v tomto případě komutativní, jelikož $A^{-1} \cdot A = A \cdot A^{-1}$. Učiňme tedy součin v obou pořadích:

$$X^*X = \begin{pmatrix} \overline{a} & \overline{c} \\ \overline{b} & \overline{d} \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} |a|^2 + |c|^2 & \overline{a}b + \overline{c}d \\ a\overline{b} + c\overline{d} & |b|^2 + |d|^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$XX^* = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \overline{a} & \overline{c} \\ \overline{b} & \overline{d} \end{pmatrix} = \begin{pmatrix} |a|^2 + |b|^2 & a\overline{c} + b\overline{d} \\ \overline{a}c + \overline{b}d & |c|^2 + |d|^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Z toho nám vyplyne hned několik vztahů:

$$|a|^2 + |c|^2 = |b|^2 + |d|^2 = |a|^2 + |b|^2 = |c|^2 + |d|^2 = 1,$$
$$\overline{a}b + \overline{c}d = a\overline{b} + c\overline{d} = a\overline{c} + b\overline{d} = \overline{a}c = \overline{b}d = 0.$$

(b) Jedná se o velice jednoduchou úpravu. Využijeme rovností z části (a) a postupně dosazujeme:

$$\begin{aligned} |a|^2 + |b|^2 &= |c|^2 + |d|^2 \\ |a|^2 + |b|^2 &= |c|^2 + 1 - |b|^2 \\ 1 - |c|^2 + |b|^2 &= |c|^2 + 1 - |b|^2 \\ 2|b|^2 &= 2|c|^2 \\ |b| &= |c|. \end{aligned}$$

(c) Vezměme si obecný tvar matice X a její determinant $\det X = ad - bc = 1$ a třeba vztah $\overline{a}b + \overline{c}d = 0$. Z první rovnice si jednoduše vyjádříme $d = \frac{1+bc}{a}$ a dosadíme do rovnice druhé:

$$\begin{aligned} \overline{a}b + \overline{c}d &= 0 \\ \overline{a}b + \overline{c}\frac{1+bc}{a} &= 0 \\ \overline{a}b + \frac{\overline{c}}{a} + \frac{b|c|^2}{a} &= 0 \\ |a|^2b + \overline{c} + b|c|^2 &= 0 \\ b(|a|^2 + |c|^2) + \overline{c} &= 0 \\ b &= -\overline{c} \\ c &= -\overline{b}. \end{aligned}$$

(d) Nuže, jdeme na to:

$$\begin{aligned} \overline{a}b + \overline{c}d &= 0 \\ \overline{a}b &= -\overline{c}d \\ \overline{a}b &= bd \\ \overline{a} &= d. \end{aligned}$$

2. Nejdříve se podíváme na matici $Z(\alpha)$. Vidíme z jejího zápisu, že je čtvercová. Ověřme tedy nejdřív, že její součin s hermitovsky transponovanou maticí vyjde jako jednotková matice:

$$\begin{aligned} Z(\alpha)^* Z(\alpha) &= \begin{pmatrix} e^{-i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix} = \begin{pmatrix} e^{-i\alpha} \cdot e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \cdot e^{-i\alpha} \end{pmatrix} \\ &= \begin{pmatrix} e^0 & 0 \\ 0 & e^0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I. \end{aligned}$$

Determinant nyní jednoduše vypočítáme:

$$\det Z(\alpha) = e^{i\alpha} \cdot e^{-i\alpha} - 0 \cdot 0 = 1.$$

Teď se vrhneme na druhou matici, $Y(\beta)$. Jelikož ta má pouze reálná čísla, nemusíme komplexně sdružená čísla uvažovat, provedeme tedy pouze běžnou transpozici:

$$\begin{aligned} Y(\beta)^* Y(\beta) &= \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix} \\ &= \begin{pmatrix} \cos^2\beta + \sin^2\beta & 0 \\ 0 & \sin^2\beta + \cos^2\beta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I. \end{aligned}$$

Teď opět jednoduše vypočítáme determinant:

$$\det Y(\beta) = \cos\beta \cdot \cos\beta - (-\sin\beta \cdot \sin\beta) = \cos^2\beta + \sin^2\beta = 1.$$

Teď, když víme, že obě matice patří do grupy SU(2), můžeme ověřit, zda tam patří i jejich součin. To jsme však již dokázali v úloze 3.1 obecně pro SU(n). Determinanty obou matic jsou rovny 1, tudíž i determinant jejich součinu bude roven 1, jelikož opět: $\det AB = \det A \cdot \det B$. Stejně tak jsme dokázali i druhé pravidlo s hermitovskou transpozicí pro součin dvou různých matic z SU(n), čili je jasné, že bude i součin matic $Z(\alpha)$ a $Y(\beta)$ v dané grupě.

3. Nejdříve provedeme násobení matic:

$$\begin{aligned} X &= Z(\alpha)Y(\beta)Z(\gamma) \\ &= \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix} \begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} e^{i\gamma} & 0 \\ 0 & e^{-i\gamma} \end{pmatrix} \\ &= \begin{pmatrix} e^{i\alpha}\cos\beta & e^{i\alpha}\sin\beta \\ -e^{-i\alpha}\sin\beta & e^{-i\alpha}\cos\beta \end{pmatrix} \begin{pmatrix} e^{i\gamma} & 0 \\ 0 & e^{-i\gamma} \end{pmatrix} \\ &= \begin{pmatrix} e^{i\gamma}e^{i\alpha}\cos\beta & e^{-i\gamma}e^{i\alpha}\sin\beta \\ -e^{i\gamma}e^{-i\alpha}\sin\beta & e^{-i\gamma}e^{-i\alpha}\cos\beta \end{pmatrix} \\ &= \begin{pmatrix} e^{i(\alpha+\gamma)}\cos\beta & e^{i(\alpha-\gamma)}\sin\beta \\ -e^{-i(\alpha-\gamma)}\sin\beta & e^{-i(\alpha+\gamma)}\cos\beta \end{pmatrix}. \end{aligned}$$

Lze si povšimnout, že struktura odpovídá matici $\left(\begin{smallmatrix} a & b \\ -\bar{b} & \bar{a} \end{smallmatrix}\right)$. Víme, že $|a|^2 + |b|^2 = 1$. Určeme si tedy členy $|a|$ a $|b|$ z této rovnice pomocí determinantu druhé podoby matice:

$$\begin{aligned} \det X &= (e^{i(\alpha+\gamma)}\cos\beta \cdot e^{-i(\alpha+\gamma)}\cos\beta) - (-e^{-i(\alpha+\gamma)}\sin\beta \cdot e^{i(\alpha-\gamma)}\sin\beta) \\ &= \cos^2\beta + \sin^2\beta = 1. \end{aligned}$$

Vidíme, že $|a|^2 = \cos^2\beta$ a $|b|^2 = \sin^2\beta$. Z toho nám plyne, že $|a| = \cos\beta$ a $|b| = \sin\beta$. Jelikož víme, že kladných hodnot nabývají zároveň funkce cosinus a sinus pouze v rozmezí $\beta \in [0, \pi/2]$, bude β nabývat pouze těchto hodnot pro dodržení vztahu.

Teď nám zbývají úhly α a γ. Obecně lze zapsat komplexní číslo $z \in \mathbb{C}$ jako $z = |z|(\cos\theta + i\sin\theta)$. Můžeme tak zkusit zapsat komplexní čísla a a b. Již jsme zjistili, že $|a| = \cos\beta$. Zbývá nám tedy už jen člen $\cos\theta + i\sin\theta$, který lze však také zapsat jako $e^{i\theta}$. To už nás přibližuje našemu výsledku. Upravme tedy tvar a z matice, aby odpovídal našemu novému zápisu:

$$a = \cos\beta \cdot e^{i(\alpha+\gamma)} = |a|e^{i(\alpha+\gamma)}.$$

Z toho nám plyne rovnost $\theta = \alpha + \gamma$. Pro b to bude obdobné, jen se záměnou γ za $-\gamma$. Rozlišme je proto tak, že úhel pro a označíme jako θ_1 a pro b jako θ_2, čili máme $\theta_1 = \alpha + \gamma$ a $\theta_2 = \alpha - \gamma$. Z toho si vyjádříme výsledné úhly:

$$\begin{aligned}\theta_1 + \theta_2 &= 2\alpha\\ \alpha &= \frac{\theta_1 + \theta_2}{2}\\[1em] \theta_1 - \theta_2 &= 2\gamma\\ \gamma &= \frac{\theta_1 - \theta_2}{2}.\end{aligned}$$

To by měly být tedy všechny podmínky pro úhly.

4. Nejdříve se podívejme na matici $Z(\alpha)$ a zkusme jí vynásobit obecný dvoudimenzionální komplexní vektor $\left(\begin{smallmatrix} z_1 \\ z_2 \end{smallmatrix}\right)$. Hleďte:

$$\begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} e^{i\alpha} z_1 \\ e^{-i\alpha} z_2 \end{pmatrix}.$$

Dostali jsme velice zajímavý výsledek, který nám popisuje rotace pro každou komplexní složku v její vlastní komplexní rovině, jelikož se neovlivňují navzájem. Taktéž si můžeme všimnout, že složka z_1 rotuje v kladném směru, avšak složka z_2 v záporném směru, a to o stejný úhel α. Čili, celkový rozdíl úhlů, které svírají jednotlivé složky s jejich reálnou osou bude větší o 2α.

Stejný postup využijeme i při druhé matici:

$$\begin{pmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} z_1\cos\beta + z_2\sin\beta \\ z_2\cos\beta - z_1\sin\beta \end{pmatrix}.$$

Tady se již ovlivňují jednotlivé složky navzájem; předchozí matice zachovávala relativní velikost složek, to se tady již neděje. Z toho plyne, že se mění relativní poloha v celém dvoudimenzionálním komplexním prostoru $\mathbb{C}^2$. Kdyby byl vektor složen z reálných čísel, matice by popisovala rotaci o úhel β ve dvou dimenzích v záporném směru kolem počátku. Z toho je vhodné vyvodit, že i náš komplexní vektor bude rotovat kolem počátku prostoru $\mathbb{C}^2$.

Řešení 4. dílu

Úloha 4.1

Zadání:
V prvním díle tématka jsme viděli grupu „totožnou" s grupou $F(\{x\} \mid x^n = e)$ v tom smyslu, že má stejně prvků a jednu grupu z druhé získáme pouhým přeznačením prvků x^k, operace $*$, inversu $^-$ a prázdného slova e. Která to je a hlavně proč?

Řešení:
Struktura, ve které nás n-krát zopakovaná operace vrátí zpátky na počáteční prvek, je velmi důležitá a častá; není tak divu, že jsme ji na své cestě světem grup potkali i my. Konkrétně jde o grupu n-tých odmocnin z jedné, značenou jako $\mathfrak{U}_n$. Ještě je dobré podotknout (a vskutku bylo podotknuto Mgr.^MM Jakubem Thomitzekem), že jsme se dostali na dohled od grupy rotací vybraných ze všech symetrií pravidelného n-úhelníku, která je také „stejná" jako volná grupa ze zadání. Kvůli triplicitě, kterou by přineslo potřetí přetřásat tu „stejnou" grupu, necháme tento vztah jako prostor k procvičení níže popsaného postupu.

Z grupy $F(\{x\} \mid x^n = e)$ uděláme $\mathfrak{U}_n$ následovně.

- Prvek x^k přeznačíme na $e^{\frac{k}{n} i\pi}$.
- Operaci $*$ budeme vnímat jako násobení komplexních čísel.
- Inverz k z^k bude v grupě $\mathfrak{U}_n$ prvek z^{n-k}.
- Ekvivalentem prázdného slova je samozřejmě číslo 1.

Na to, že předem zmíněné grupy jsou si až podezřele podobné, se dá přijít mnoha způsoby. Například, mohou být generované jedním prvkem, mají (pro stejné n) stejně prvků a splňují onu cykličnost, o které byla řeč na začátku řešení. Tím narážíme na to, proč je dobré považovat tyto dvě grupy za stejné. Ne každého hned napadne, že v grupě $F(\{x\} \mid x^n = e)$ se násobením prvkem x jakoby točíme dokola; jakmile si uvědomíme podobnost s $\mathfrak{U}_n$, je tato skutečnost průzračná.

Úloha 4.2

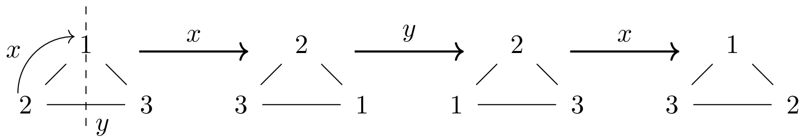
Zadání:
(1) Najděte zmíněnou relaci vyjadřující vztah rotací a reflexí v grupě D_{2n}. Opravdu stačí jen jedna. Nedoporučujeme řídit se slovy: „Složení rotace a reflexe je opět reflexe," neboť ve většině případů je výsledná reflexe rozdílná

a my značíme písmenem y jenom jednu *nějakou reflexi. Spíše, jakou reflexi dostaneme, když výslednou reflexi zase složíme s touž rotací jako prve?*

(2) Dokažte, že grupa $F(\{x,y\})$ s relacemi $x^n = e$, $y^2 = e$ a relací z části (1) je opravdu totožná grupě $(D_{2n}, \circ, ^{-1}, \mathrm{id})$. To znamená, že má stejně prvků, písmena x a y lze ztotožnit s generující rotací a reflexí, operace $\circ$ se „chová jako“ $*$, invers $^{-1}$ jako $^{-}$ a id *jako e.*

Řešení:
Relace, která se skrývala pod tajemným uvedením v sekci (1), je $xyx = y$ (nebo ekvivalentně $xy = y\bar{x}$, když celou rovnici zprava vynásobíme inverzem k x). Nahlédnout, že takováto relace platí pro rotační symetrii pravidelného n-úhelníku zvanou x a pro kteroukoli jeho symetrii osovou křtěnou y, je možné přes nakreslení pár příkladů nebo přes trošku složitější argument. Začneme obrázkem.

Abychom to měli jednoduché, zvolíme si $n = 3$, symetrii y si zvolíme reflexi podle svislé osy a rotace x bude o 60 stupňů. Poté rotujeme a překládáme podle obrázku 1, kde poslední trojúhelník je ten původní jen reflektovaný symetrií y (toto je tvoje dnešní znamení být jako trojúhelník a reflektovat se!).



Obrázek 1: Reflexe obložená v rotacích prodírající se skrz.

Argumentace vypadá následovně. Představíme si nějaký vrchol pravidelného n-úhelníku který je m vrcholů od bodu[6] p, kde osa symetrie protíná celý útvar (doporučujeme si symetrie představovat výhradně svisle). Pro takový vrchol platí, že ho rotace nejdříve posune o nějakých k vrcholů od bodu p, který nerotujeme. Reflexe tuto jeho vzdálenost změní na $-m-k$ (symbol $-$ značí vzdálenost od bodu p v opačném směru), a pokud k ní zas přičteme k (tedy celý útvar zrotujeme pomocí x) získáme $-m$, což je přesně to, co bychom čekali, že udělá reflexe.

Teď se od rozmluv o tom, proč relace $xyx = y$ platí i pro reflexe a rotace, přesuneme k tomu, proč společně s relacemi ze zadání svazuje volnou grupu $F(\{x,y\})$ do podoby D_{2n}.

Začneme tím, že každý prvek volné grupy zobrazíme na prvek D_{2n}. Prvky x, y pošleme na nějakou generující rotaci a libovolnou reflexi. Ty složitější prvky jako například slovo yx^kyx^ly (kde k, l jsou celá a $k > l$) se po aplikaci relací $xyx = y$, $y = \bar{y}$, $x^n = e$ zredukují na yx^{k-l} (přičemž můžeme díky třetí relaci předpokládat, že $k - l < n$) což je zaručeně prvek D_{2n}, protože x, y generují tuto

[6] Musíme zde rozlišovat mezi bodem a vrcholem, protože pro sudá n máme i osy symetrie, které neprotínají své n-úhelníky v jejich vrcholech.

grupu. Dále nám zbývá dokázat, že se obě zmíněné grupy podrobují těm stejným vztahům.

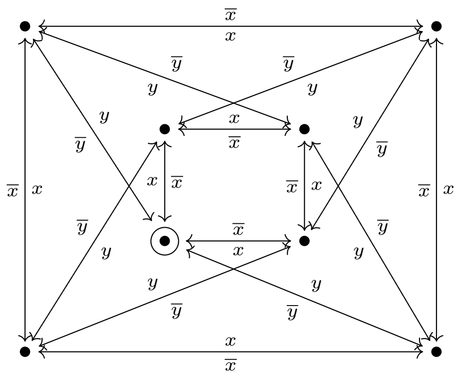
Protože všechny tři relace definující $F(\{x,y\} \mid x^n = e,\ xyx = y,\ y = \bar{y}) = F$, jak bylo ukázáno v tomto a úplně prvním díle tématka – platí i v D_{2n} a z definice v této grupě neplatí žádné jiné, můžeme si být jisti, že F nemá žádné relace navíc oproti D_{2n}.

Posledním úkolem je ověřit, že grupa D_{2n} nemá nějakou relaci navrch oproti F. Jak už víme, grupa D_{2n} má $2n$ prvků a všechny jsou ve tvaru yx^k nebo x^k pro $k < n$. Ty mezi sebou nesdílejí žádné další relace, protože jsou si navzájem různé a další relace by jim jejich odlišnost odpírala. Grupa D_{2n} tak nemůže mít více než ty tři relace uvedené v definici F, protože o nich zaručeně víme, že v ní platí.

Ve dvou předchozích odstavcích jsme definovali zobrazení, které z prvků F jednoznačně dělá prvky D_{2n} respektujíc všechny relace i operace v D_{2n}. To stačí k závěru, že F a D_{2n} jsou „stejné“ grupy až na názvy prvků.

Úloha 4.3

Zadání:
Nakreslete konečný automat pro grupu D_8 symetrií čtverce. Znalost relace z úlohy 2, části (1) vám může přijít vhod, ale není pro nakreslení automatu nezbytná. Inspirujte se obrázkem 2. Grupy $\mathbf{Q}_8$ a D_8 sice nejsou, ale jisté strukturální podobnosti mají.



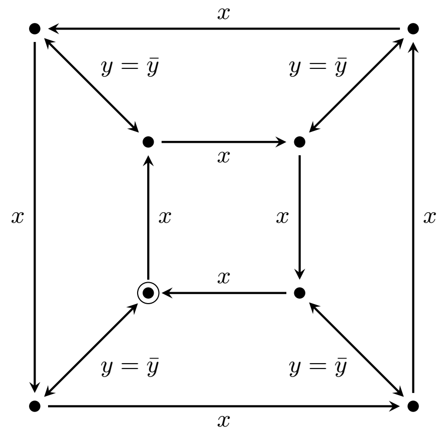
Obrázek 2: Konečný automat pro grupu kvaternionů $\mathbf{Q}_8$.

Řešení:
Jako řešení citujme zdařilý počin Mgr.[MM] Lukáše Komy.

> Všechny rotace x^n tvoří v diagramu čtverec s šipkou pro x (a $\bar{x}$ v opačném směru) mezi každými dvěma vrcholy. Reflexe yx^n tvoří také čtverec a mezi reflexemi také přecházím rotací. Z každého vrcholu jednoho čtverce se dostaneme reflexí

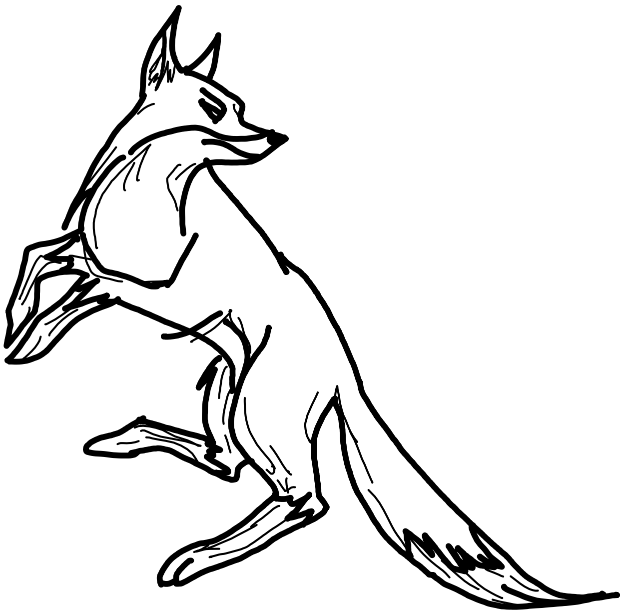
$y = \bar{y}$ do právě jednoho vrcholu druhého čtverce. Protože složením reflexe, rotace a téže reflexe získáme rotaci v opačném směru $((yx^m)(x^n)(yx^m) = x^{-n}$, vyjádřeno také relací z úlohy 5.2: $yxy = \bar{x}$), bude mít mezi vrcholy jednoho čtverce šipka x opačný směr než mezi odpovídajícími vrcholy druhého.

Výsledný konečný automat pro grupu D_8 symetrií čtverce:



Rotace $\bar{x}$ byly pro srozumitelnost vynechány.

Adam a Jáchym; `grupytematko@gmail.com`



Téma 3 – Elektrostatika

Rozloučení

Milý řešiteli, doufám, že sis užil řešení tématka o elektrostatice stejně, jako jsem si já užil jeho psaní. Doufám, že tě nadcházející léto kladně nabije do dalšího školního roku a snad i k řešení dalšího ročníku M&M. S přáním hezkého dne,

Radim N.; `radim05@post.cz`

Téma 4 – Výpočetní geometrie

Rozloučení

Tématko bylo a pro ty, co by si chtěli více zařešit, máme poslední zábavnou úložku na přemýšlení. Mějme mnohoúhelník a ještě druhý mnohoúhelník zvaný plot tak, že s prvním nemůžeme vůbec pohnout, aniž bychom protnuli druhý. Rozhodněte, zda je možné odstranit jednu hranu plotu tak, abychom dostali první mnohoúhelník ven. Mějte se hezky,

Dláža; `gadurekvojtech@outlook.com`

Téma 5 – Kombinatorika

Díl 6: Rozloučení

Tak se nám ročník a tedy i toto tématko nachýlilo ke konci. Doufáme, že vám naše tématko přišlo zajímavé a že jste se něco naučili. Ale pokud by vám naše tématko chybělo, tak se nebojte, už teď můžete řešit tématko na Grafy, které je do jisté míry jeho duchovní nástupce. Doufáme, že ho budete taky řešit :D

Řešení 4. dílu

Úloha 4.1

Zadání:
Máme tabulku 5×5 a dva lidé hrají hru. Pravidelně se střídají v tazích. Hráč na tahu vždy obarví jedno políčko tabulky svojí barvou. Pokud hráč svojí barvou zvládne obarvit 4 vrcholy nějakého obdélníku, tak vyhrál a hra končí. Dokažte, že hra nemůže skončit remízou, čili jeden hráč jistě vyhraje.

Řešení:
Stačí nám dokázat, že pro každé obarvení tabulky vznikne obdélník. Barvou hráče, který začal, bude na konci nabarveno 13 políček. Tedy z Dirichletova principu existuje alespoň 1 řádek obsahující alespoň 3 políčka této barvy (dále si je označíme PB).

Pokud řádek s největším počtem PB jich obsahuje 5, tak pak opět z Dirichletova principu existuje jiný řádek, kde jsou alespoň 2 PB a tedy nám vznikne obdélník.

Pokud obsahuje 4, tak poté nám zbývá 9 PB v ostatních řádcích, a pokud libovolná dvojice je ve stejných sloupcích, jako 4 PB v původním řádku, tak nám vzniknou vrcholy obdélníku. To ale znamená, že abychom nevytvořili vrcholy obdélníku, tak zvládneme obarvit maximálně 8 políček, ale my potřebujeme obarvit 9.

Pokud obsahuje 3, tak z Dirichleta existují alespoň 3 řádky, co obsahují 3 PB, ovšem ty nám musí tvořit vrcholy obdélníku, protože pokud by první dvě netvořily, tak by musely pokrývat všechny sloupce. Pak by ovšem třetí řádek měl alespoň 2 PB ve stejných sloupcích jako jeden z předchozích řádků. Tedy vždy vzniknou vrcholy obdélníku.

Úloha 4.2

Zadání:
Mějme 2 červené kuličky, 3 modré a 3 zelené. Kuličky stejné barvy jsou navzájem nerozlišitelné. Spočítejte, kolika způsoby je lze seřadit tak, aby vedle sebe nikdy nebyly všechny kuličky od jedné barvy (tudíž mezi první a poslední kuličkou jedné barvy v řadě musí být alespoň jedna kulička jiné barvy).

Řešení:
Použijeme princip inkluze a exkluze podle počtu barev, kde jsou všechny stejné barvy vedle sebe. Tedy nejprve vezmeme celkový počet uspořádání, od toho ode-

čteme všechna uspořádání, kde jsou všechny kuličky jedné barvy vedle sebe, poté přičteme ty, kdy jsou kuličky dvou barev vedle sebe, a nakonec přičteme všechny, kde jsou kuličky všech tří barev u sebe. To spočítáme standardním postupem z prvního dílu a to tak, že kuličky stejné barvy bereme za 1 kuličku ve výpočtu. Dostaneme tedy:

$$560 - 260 + 52 - 6 = 346.$$

Úloha 4.3

Zadání:
Z čísel $1, 2, \ldots, 1000$ *vyškrtáme všechny násobky 3, 5, 7 a 42. Kolik čísel nám zůstane?*

Řešení:
Nejprve si uvědomíme, že 42 je násobek 3 (a také 7), takže ho můžeme ignorovat. Poté pro 3, 5 a 7 postupujeme stejně jako v minulém příkladě (tedy pomocí PIE), pouze místo barev, které jsou vedle sebe, odečítáme nejprve čísla dělitelná jedním z čísel, poté přičteme ty dvěma a nakonec odečteme ty třemi. Protože čísla jsou vzájemně nesoudělná (tedy neexistuje přirozené číslo větší než 1, které by libovolná dvě z nich dělilo), tak je číslo dělitelné libovolnou k-ticí z nich, pokud je dělitelné součinem této k-tice (neboli např. číslo je dělitelné 3 a 7 právě tehdy, když je dělitelné 21). A počet čísel dělitelných k až do nějakého n je roven dolní celé části z $\frac{n}{k}$, protože pokud n je násobek k, tak tvrzení očividně platí a dolní celá část z $\frac{n}{k}$ je to samé jako $\frac{m}{k}$, kde m je největší násobek k do n. Poté už pouze spočítáme počet čísel − počet čísel dělitelný 3, 5, 7 + počet čísel dělitelný $3 \cdot 5$, $5 \cdot 7$, $3 \cdot 7$ − počet čísel dělitelný $3 \cdot 5 \cdot 7$:

$$1000 - 675 + 141 - 9 = 457.$$

Úloha 4.4

Zadání:
Určete počet cest délky $a + b$ *z levého dolního rohu do pravého horního v mřížce* $a \times b$.

Řešení:
Stačí nám pouze si uvědomit, že každopádně uděláme $a + b$ pohybů a jenom vybíráme, které budou nahoru a které doprava (pohyby dolů a doleva by nám přidaly jeden pohyb, který by nás ale oddálil od cíle, takže bychom se pak do cíle nedostali na a+b kroků). Pohybů nahoru bude a a těch doprava b, tedy vybíráme a (nebo b) prvků z $a + b$ bez závislosti na pořadí. Tedy řešení je $\binom{a+b}{a}$ (což je ekvivalentní s $\binom{a+b}{b}$).

Úloha 4.5

Zadání:
Rozkladem čísla n *délky* $k \geq 1$ *rozumíme konečnou nerostoucí posloupnost přirozených čísel* $a_1, \ldots, a_k$ *splňující* $a_1 + \cdots + a_k = n$. *Dokažte, že počet všech rozkladů čísla* n *je roven počtu rozkladů čísla* $2n$ *délky* n.

Řešení:
Vezmeme si nějaký rozklad n a doplníme ho nulami tak, abychom dostali posloupnost délky n a poté ke každému prvku této posloupnosti přičteme 1. Tím dostaneme rozklad $2n$ délky n a očividně ze dvou různých rozkladů n nemůžeme dostat stejný rozklad $2n$ délky n, a tedy platí, že počet rozkladů n je menší nebo roven počtu rozkladů $2n$ délky n. Naopak když v rozkladu $2n$ délky n od každého prvku odečteme 1 a smažeme 0, tak dostaneme rozklad n a opět ze dvou rozkladů $2n$ nemůžeme dostat stejný rozklad n, a tedy platí, že počet rozkladů počet n je větší nebo roven počtu rozkladů $2n$ délky n, a tedy je obou stejně.

Řešení 5. dílu

Úloha 5.1

Zadání:
Kolika způsoby můžeme obarvit tabulku 4×4 n barvami, jestliže považujeme za totožné tabulky, mezi nimiž můžeme přecházet otočením?

Řešení:
Jedná se o jednoduchou aplikaci Burnsidova lemmatu. Máme 4 otočení a to o 0°, 90°, 180° a 270°. Pevné body rotace o 0° jsou všechna obarvení (tedy n^{16}), pevné body rotace o 180° jsou určeny jednou polovinou čtverce, tedy jich je n^8. Pevné body rotací o 90° a 270° jsou určeny čtvrtinou čtverce, a tedy jich je n^4. Celkově je tedy počet obarvení $\frac{n^{16}+n^8+2n^4}{4}$.

Úloha 5.2

Zadání:
Máme tabulku tvaru $n \times 4$. Pepíček každé její políčko obarví jednou ze 3 barev. Určete nejmenší n takové, že nezávisle na Pepíčkově volbě barev můžeme vždy vytvořit jednobarevný čtverec 2×2 pouze prohazováním řádků a sloupců.

Řešení:
Úloha je principem velmi podobná úloze 4.1, protože nám pouze stačí najít rohy obdélníku a pak už jednoduše dostaneme čtverec. Jistě $n = 18$ nám nestačí, protože nám stačí pouze zvolit řádky tak, že v každém bude jedna dvojice jedné barvy a zbylé 2 budou mít zbylé 2 barvy. Tak máme v každém řádku pouze jednu dvojici, která může potenciálně tvořit rohy obdélníku, a protože platí, že $\binom{4}{2} = 6$, tak každou dvojici nějaké barvy můžeme umístit na 6 míst, a protože máme 3 barvy, tak můžeme vytvořit 18 řádků bez kolize. Ovšem zároveň vidíme, že už si nemůžeme dovolit 19 řádků, ve kterých každý obsahuje alespoň 1 dvojici, ovšem z Dirichleta každý řádek obsahuje alespoň 1 dvojici, a tedy v 19 řádcích vždy můžeme vytvořit čtverec.

Úloha 5.3

Zadání:
Jak se změní výsledek úlohy, kterou jsme počítali v tématku, pokud budeme za stejné považovat nejenom náramky, které na sebe otočíme, ale i náramky, které

na sebe převrátíme – tedy efektivně zobrazíme zrcadlem (neboli nějakou rovinovou souměrností) (tedy například posloupnosti $ZMOR$ a $ZROM$ považujeme za stejné náramky)? Co když budeme moct pouze převracet a ne otáčet?

Řešení:
Přidáním rotace jsme si přidali 4 další permutace do grupy permutací a to ty, které nám (1, 2, 3, 4) zobrazí na (4, 3, 2, 1), (3, 2, 1, 4), (2, 1, 4, 3), (1, 4, 3, 2). První má pevné body v náramcích, které mají stejnou barvu na pozicích 1 a 4 a na pozicích 2 a 3. Pro třetí platí to samé s dvojicemi 1, 2 a 3, 4. A druhá musí mít stejnou barvu na pozicích 1, 3 a čtvrtá na 2, 4. Tedy po dosazení to lemmatu dostaneme:

$$\frac{4^4 + 2 \cdot 4^3 + 3 \cdot 4^2 + 2 \cdot 4}{8} = 2^5 + 16 + 6 + 1 = 55.$$

To je očividně menší číslo, než nám vyšlo v původním příkladě, což dává smysl, protože jsme ztotožnili více náramků než v původním příkladu. Pokud bereme pouze převracení, tak si můžeme všimnout, že dvě různá převrácení dohromady nám dají rotaci, a tedy považujeme za stejné přesně stejné věci, jako když bereme i rotace (protože vztah „považujeme za stejné“ se zachová, i když zobrazujeme víc než jednou).

Lukáš, Terka; `troj.lukas@gmail.com`

Téma 6 – Lisp

Díl 6: Rozloučení

Srdečně děkuji všem, kteří Lisp řešili. Věřím, že jsme se společně přesvědčili o tom, že to není ezoterický jazyk, ale mocný nástroj, který je jen odlišný od většiny jiných jazyků. Rád bych uvedl dvě poslední myšlenky: První z nich je „kanonické řešení“[7], to jest takové řešení problému, pro něž lze očekávat, že by napadlo v podstatě kohokoli, klidně i mimozemšťany. Rád si představuji, že pokud se s nějakými někdy setkáme, budou taky mít nějaký programovací jazyk, jehož syntax se skládá pouze ze závorek. Druhá myšlenka je Greenspun's tenth rule[8]: Každý dostatečně složitý program obsahuje zabugovanou a pomalou implementaci půlky Common Lispu. S pomyšlením na toto „pravidlo“ vás žádám, abyste rovnou použili Lisp, a to ne nutně jako hlavní programovací jazyk, ale klidně (a možná i radši) i jako jazyk na konfiguraci nebo skriptování. Postupným rozšiřováním možností konfigurace a skriptování nevyhnutelně narazíte na Greenspun's tenth rule, a v takovém případě je lepší už Lisp rovnou používat. Přes léto si odpočiňte a zkuste přežít vedra. Přeji vám všem hezké prázdniny.

Řešení 4. a 5. dílu

V minulých dvou dílech jste si vyzkoušeli lambdy a vlastní makra (4. díl) a poté ruční implementaci `wget` přes sockety (5. díl). Níže vybíráme z vašich řešení.

Úloha 4.1 – range

Zadání:
Implementujte funkci `range` *podobně jako v Pythonu.*

Bc.[MM] Martin Vagner hezky sjednotil iteraci nahoru i dolů do jednoho cyklu: místo větvení podle znaménka kroku porovnává `(* start (signum stp))` s `(* end (signum stp))`, což v případě záporného kroku otočí nerovnost (vynásobením obou stran -1).

Řešení od Bc.[MM] Martina Vagnera:

```
(defun range (end &optional (start 0) (stp 1))
  (let ((res '()))
    (loop while (< (* start (signum stp)) (* end (signum stp))) do
          (setf res (append res (list start)))
          (setf start (+ start stp)))
    res))
```

[7] https://mathoverflow.net/questions/19644/what-is-the-definition-of-canonical
[8] https://en.wikipedia.org/wiki/Greenspun%27s_tenth_rule

Úloha 4.2 – třetí mocniny lichých čísel

Zadání:

Vygenerujte seznam třetích mocnin všech lichých čísel od 1 do 100. Použijte funkce `mapcar`, `remove-if-not` *a* `range`.

Prof.MM Julie Klementová postup rozepsala do pojmenovaných kroků přes `let*`, díky kterým je vidět každý mezivýsledek. Staví přitom na vlastní variantě `range` postavené na `loop ... collect`.

Řešení od Prof.MM Julie Klementové:

```
(defun range2 (end &optional (start 0) (step 1))
  (when (< end start) (rotatef start end))
  (loop for i from start below end by step collect i))

(defun odd-cubes-example (n)
  (let* ((seznam  (range2 n 1))
         (liche   (remove-if-not (lambda (x) (oddp x)) seznam))
         (mocniny (mapcar (lambda (x) (expt x 3)) liche)))
    mocniny))
```

Úloha 4.3 – vlastní funkce nad seznamy

Zadání:

Implementujte vlastní varianty funkcí `length`, `map`, `filter`, `left-fold` *a* `right-fold` *pro seznamy.*

Prof.MM Michael Jarvis zvolil tradiční tail-call styl, který je běžný v funkcionálních jazycích. Na foldech je hezky vidět rozdíl: levý fold akumuluje výsledek při sestupu (a používá tedy tail-call), kdežto pravý fold skládá výsledek až při návratu z rekurze.

Řešení od Prof.MM Michaela Jarvise:

```
(defun my-length-tail (sequence tail)
  (if (eq sequence NIL)
      tail
      (my-length-tail (cdr sequence) (+ tail 1))))
(defun my-length (sequence)
  (my-length-tail sequence 0))

(defun my-map-tail (fn sequence tail)
  (if (eq sequence NIL)
      tail
      (my-map-tail fn (cdr sequence)
                   (append tail (list (funcall fn (car sequence)))))))
```

```
(defun my-map (fn sequence)
  (my-map-tail fn sequence '()))

(defun my-filter-tail (pred sequence tail)
  (if (eq sequence NIL)
      tail
      (if (funcall pred (car sequence))
          (my-filter-tail pred (cdr sequence)
                          (append tail (list (car sequence))))
          (my-filter-tail pred (cdr sequence) tail))))
(defun my-filter (pred sequence)
  (my-filter-tail pred sequence '()))

(defun my-left-fold (fn initial sequence)
  (if (eq sequence NIL)
      initial
      (my-left-fold fn (funcall fn initial (car sequence)) (cdr sequence
    ))))

(defun my-right-fold (fn initial sequence)
  (if (eq sequence NIL)
      initial
      (funcall fn (car sequence) (my-right-fold fn initial (cdr sequence
    )))))
```

Úloha 4.4 – makro while

Zadání:
Implementujte makro `while`*.*

Nejpřímější možné řešení, ve kterém makro `while` jen přepíše svůj zápis na ekvivalentní `loop`. Přesně tohle makra běžně dělají: dají hezčí jméno (nebo příjemnější syntax) už existující konstrukci.

Řešení od Bc.$^{\text{MM}}$ Martina Vagnera:

```
(defmacro while (condition &body body)
  `(loop while ,condition do
        ,@body))
```

Když jsem úlohu zadával, nedošlo mi, jak snadná bude s použitím dostupných nástrojů. Níže uvedená autorská varianta loop nepoužívá a opakování smyčky zajišťuje rekurzí.

```
(defun recurse (fun)
  (funcall fun fun))
(defmacro while (condition &body body)
  `(recurse (lambda (self)
      (cond
        (,condition (progn ,@body (recurse self)))
        (T ()) ; konec smycky
    ))
  )
)
```

Úloha 4.5 – makro replace-symbol

Zadání:
Implementujte makro `replace-symbol`*, které ve svém těle nahradí všechny výskyty daného symbolu za jiný symbol.*

Prof.MM Michael Jarvis oddělil práci do čisté pomocné funkce, která rekurzivně projde syntaktický strom těla, a makro pak výsledek jen obalí do `progn`.

Řešení od Prof.MM Michaela Jarvise:

```
(defun replace-symbol-tail (old new body tail)
  (if (eq body NIL)
      tail
      (if (atom (car body))
          (if (eq old (car body))
              (replace-symbol-tail old new (cdr body) (append tail (list
     new)))
              (replace-symbol-tail old new (cdr body) (append tail (list
     (car body)))))
          (replace-symbol-tail old new (cdr body)
            (append tail (list (replace-symbol-tail old new (car body) '
    ())))))))

(defmacro replace-symbol (old new &body body)
  `(progn ,@(replace-symbol-tail old new body '())))
```

Problém 4.6 – vlastní makro

Zadání:

Vymyslete nějaké makro a vysvětlete, v jaké situaci byste ho použili.

Bc.[MM] Martin Vagner vymyslel hned dvě makra. `f` vytváří „f-stringy“ jako v Pythonu: `(f "Ahoj {jmeno}!")` se přeloží na volání `format` s vytaženými argumenty. `inline-loop` pak smyčku `(loop for x in <list> do <telo>)` rozbalí („unroll“) už při překladu na rozepsané kroky. Autor upozorňuje, že dosazení není scopované (to znamená, že nahradí **všechny** výskyty symbolu `x`, a to i v případě, kde je předefinován pomocí `let`), takže uvnitř těla nelze `x` bezpečně předefinovat.

Řešení od Bc.[MM] Martina Vagnera:

```
(defmacro f (str)
  (let ((fmt "") (getting-var-name nil) (varname "") (args '()))
    (loop for i in (range (length str)) do
      (cond
        ((and getting-var-name (char= (char str i) #\}))
         (setf args (append args (list (read-from-string varname)))
               varname "" getting-var-name nil))
        (getting-var-name
         (setf varname (concatenate 'string varname (subseq str i (+ i
    1)))))
        ((char= (char str i) #\{)
         (setf fmt (concatenate 'string fmt "~A") getting-var-name t))
        (t (setf fmt (concatenate 'string fmt (subseq str i (+ i 1))))))
    )
    `(format nil ,fmt ,@args)))

(defmacro inline-loop (&body body)
  (if (or (not (string= (car body) "FOR")) (not (string= (caddr body) "
    IN"))
          (not (string= (nth 4 body) "DO")))
      (error "inline-loop supports only for <x> in <list> do"))
  (setf body (cdr body))
  (let ((varname (car body)) (lst (caddr body)) (res '(progn)))
    (setf body (cddddr body))
    (cond
      ((eq (car lst) 'quote)
       (loop for i in (cadr lst) do
         (setf res (append res (list (macroexpand `(replace-symbol ,
    varname ,i ,@body)))))))
      ((loop for i in (eval lst) do
         (setf res (append res (list (macroexpand `(replace-symbol ,
    varname ,i ,@body))))))))
    res))
```

Úloha 5.1 – stavový kód a Content-Type

Zadání:

Upravte funkci `wget` *tak, aby na standardní výstup vypsala pouze stavový kód odpovědi (např. 200) a hodnotu hlavičky* `Content-Type`*.*

Dr.MM Lukáš Koma vyřešil celou HTTP ságu jediným klientem. Jeho kód uvedeme v zájmu čitelnosti postupně. Zde najdete pomocné funkce na čtení/zápis řádků, parsování stavového řádku a hlaviček a přístupové funkce k odpovědi, kterou si drží jako seznam `(stav-čhlaviky ětlo historie-řěpesmrování)`. Úloha 5.1 je pak už jen „pošli GET a vyzobni stav a Content-Type“. Proti originálu proběhla malá úprava: jména HTTP hlaviček jsou case-insensitive, proto v `get-header` porovnáváme přes `string-equal`.

Řešení od Dr.MM Lukáše Komy:

```
(ql:quickload :usocket)

(defun split-str-on-first-occur (string delimeter-char)
  ; Rozdeli retezec na prvnim vyskytu znaku
  (let ((string-rest (subseq string 1)))
    (if (char= delimeter-char (char string 0))
      (list "" string-rest)
      (let ((string-rest-parts (split-str-on-first-occur string-rest
    delimeter-char)))
        (list
          (concatenate 'string (subseq string 0 1) (car string-rest-
    parts))
          (nth 1 string-rest-parts))))))

(defun read-crlf-line (&rest args)
  (let ((line (apply #'read-line args)))
    (if line
      (string-right-trim '(#\Return) line)
      NIL)))

(defun http-line (stream line)
  (format stream "~a~c~c" line #\Return #\Linefeed))

(defun http-line-format (stream &rest rest-args)
  (http-line stream (apply #'format (cons NIL rest-args))))

(defun send-header (stream name value)
  (http-line-format stream "~a: ~a" name value))

(defun send-headers (stream headers)
  ; Vezme seznam (("Jmeno-Hlavicky" "hodnota") ("Dalsi-Jmeno" "dalsi/
    hodnota") ...)
```

```
  (loop for header in headers
    do (send-header stream (car header) (nth 1 header))))

(defun parse-response-status (response-first-line)
  ; Vrati seznam (stavový kod, popis stavu), napriklad (200 "OK")
  (let (
      (words
        (split-str-on-first-occur (nth 1 (split-str-on-first-occur
    response-first-line #\ )) #\ )))
    (list (parse-integer (car words)) (nth 1 words))))

(defun parse-header (line)
  ; Vrati seznam (Jméno hlavicky, hodnota), napriklad ("Content-Type" "
    applycation/json")
  (let ((parts (split-str-on-first-occur line #\:)))
    (list (car parts) (subseq (nth 1 parts) 1))))

(defun get-header-from-list (name headers)
  ; Vezme seznam (("Jmeno-Hlavicky" "hodnota") ("Dalsi-Jmeno" "dalsi/
    hodnota") ...) a jmeno,
  ; vrati hodnotu hlavicky (ěpripadn NIL).
  (when headers
    (let ((header (car headers)))
      (if (string-equal name (car header))
        (nth 1 header)
        (get-header-from-list name (cdr headers))))))

; Funkce pro prehledné ziskavani dat z odpovedi

(defun get-status-code (response)
  (car (car response)))

(defun get-status-comment (response)
  (nth 1 (car response)))

(defun get-content (response)
  (nth 2 response))

(defun get-headers (response)
  ; Vrati seznam (("Jmeno-Hlavicky" "hodnota") ("Dalsi-Jmeno" "dalsi/
    hodnota") ...)
  (nth 1 response))

(defun get-header (name response)
  (get-header-from-list name (get-headers response)))

(defun get-previous-responses (response)
```

```
  ; Vrati seznam predchozich odpovedi, jejichz presmerovani vedlo na tu
    finalni.
  (nth 3 response))
```

Úloha 5.2 – standardní URL adresy

Zadání:
Přidejte podporu pro používání standardních URL adres stylu *`http://httpforever.com/index.html`*.

Dr.[MM] Lukáš Koma z adresy vytáhne hosta, cestu a číslo portu. To sice nebylo v zadání, ale je součástí specifikace URL a občas se v praxi vyskytuje. Adresy bez zakončujícího lomítka navíc doplní, protože základní cesta (taková, která nijak dál nespecifikuje, jaký zdroj v rámci stránky chceme), je vždy alespoň `"/"`. Podobně jako u předchozí úlohy je snadné udělat chybu: je lákavé přečíst prvních 7 znaků adresy za účelem porovnání s řetězcem `https://`, ale existují validní adresy kratší než 7 znaků, jako je např. adresa `czc.cz`; proto si nejdřív ověříme délku.

Řešení od Dr.[MM] Lukáše Komy:

```
(defun parse-url (url)
  ; Vrati seznam (host cesta), napriklad ("httpbin.org" "/get")
  (let* (
      (url-without-protocol ; Zahodime protokol pokud je v adrese,
    stejne umime jen jeden
        (if (and (>= (length url) 7) (string= "http://" (subseq url 0 7)
    )) (subseq url 7) url))
      (slash-position (position #\/ url-without-protocol))
      (host-and-port (subseq url-without-protocol 0 slash-position))
      ; Pokud adresa obsahuje pouze hosta, nastavime cestu na /
      (path (if slash-position (subseq url-without-protocol slash-
    position) "/"))
      (colon-position (position #\: host-and-port))
      (host (subseq host-and-port 0 colon-position))
      (port (if colon-position (parse-integer (subseq host-and-port (+
    colon-position 1))) 80)))
    (list host port path)))
```

Úloha 5.3 – přesměrování

Zadání:

Přidejte podporu pro přesměrování. Pokud server vrátí stavový kód 301 nebo 302, najděte v hlavičkách hodnotu `Location` *Jádrem klienta je funkce* `http-request-unguarded-out-stream`*: otevře socket, pošle požadavek, přečte odpověď a v případě přesměrování se rekurzivně zavolá na novou adresu. Dr.*[MM] *Lukáš Koma tu ošetří i záludnost, na které spousta naivních klientů selže:* `Location` *může být* relativní *(začíná* `"/"`*, takže se zůstává na stejné adrese) i* absolutní*. Navíc si pamatuje celý řetěz předchozích odpovědí, takže je pak vidět, kudy se přesměrování vydalo. a stáhněte soubor z nové adresy. Omezte maximální počet přesměrování na 100, aby program necyklil.*

Jádrem klienta je funkce `http-request-unguarded-out-stream`: otevře socket, pošle požadavek, přečte odpověď a v případě přesměrování se rekurzivně zavolá na novou adresu. Dr.[MM] Lukáš Koma tu ošetří i záludnost, na které spousta naivních klientů selže: `Location` může být *relativní* (začíná `"/"`, takže se zůstává na stejném hostu) i *absolutní*. Navíc si pamatuje celý řetěz předchozích odpovědí, takže je pak vidět, kudy se přesměrování vydalo.

Řešení od Dr.[MM] Lukáše Komy:

```
(defun http-request-unguarded-out-stream (
    method
    host
    port
    path
    sent-headers
    sent-content
    headers-output-stream
    content-output-stream
    max-redirects)
  (let* (
      (all-sent-headers
        (append ; Pridame Host a pripadne Content-Length k zadanym
   hlavickam.
          (list (list "Host" host))
          (when sent-content (list (list "Content-Length" (length (
   string-to-octets sent-content)))))
          sent-headers))
      (socket (usocket:socket-connect host port))
      (stream (usocket:socket-stream socket)))
    (unwind-protect
      (progn
        (http-line-format stream "~a ~a HTTP/1.0"
          (cond
            ((eq method :get) "GET")
```

```
          ((eq method :head) "HEAD")
          ((eq method :post) "POST")
          ((eq method :put) "PUT")
          ((eq method :patch) "PATCH")
          ((eq method :delete) "DELETE")
          (t (error "Neznama HTTP metoda: ~a" method)))
        path)
      (send-headers stream all-sent-headers)
      (http-line-format stream "")
      (when sent-content (format stream "~a" sent-content))
      (force-output stream)
      (let* (
          (first-line (read-crlf-line stream))
          (status (parse-response-status first-line))
          (status-code (car status))
          (redirect (and (or (= status-code 301) (= status-code 302))
(> max-redirects 0)))
          (headers '())
          (content ""))
        (unless redirect (format headers-output-stream "~a~%" first-
line)) ; Chceme vypsat az finalni.
        (loop for line = (read-crlf-line stream "")
          until (string= line "") do
          (unless redirect (format headers-output-stream "~a~%" line))
          (setf headers (cons (parse-header line) headers)))
        (loop for line = (read-crlf-line stream NIL)
          while line do
          (unless redirect (format content-output-stream "~a~%" line))
          (setf content (format NIL "~a~a~%" content line)))
        (let ((result (list status (reverse headers) content NIL)))
          (if redirect
            (let* (
                (location (get-header "Location" result))
                (new-host-port-and-path
                  (if (char= #\/ (char location 0))
                    (list host port location)
                    (parse-url location)))
                (new-host (car new-host-port-and-path))
                (new-port (nth 1 new-host-port-and-path))
                (new-path (nth 2 new-host-port-and-path))
                (new-result
                  (http-request-unguarded-out-stream ; Rekurzivne
volame na novou lokaci.
                    method
                    new-host
                    new-port
                    new-path
```

```
                        sent-headers
                        sent-content
                        headers-output-stream
                        content-output-stream
                        (- max-redirects 1))))
                  ; Pridame odpoved do seznamu predchozich odpovedi.
                  (setf (nth 3 new-result) (cons result (nth 3 new-result)
    ))
                  new-result)
                result))))
  (usocket:socket-close socket))))
```

Problém 5.4 – další HTTP slovesa

Zadání:

Kromě GET existují i jiná HTTP „slovesa“. Přečtěte si o nich a některá z nich implementujte s vhodným rozhraním.

Volbu slovesa řeší `cond` na začátku jádra výše (doplnili jsme do něj větev pro neznámou metodu). Dr.MM Lukáš Koma nad jádrem postavil jedno jednotné rozhraní `http-request` s klíčovými argumenty, které pokrývá `GET`, `HEAD`, `POST`, `PUT`, `PATCH` i `DELETE`. Hlavičky `Host` a `Content-Length` se doplní samy, lze předat tělo, `Content-Type`, zapnout výpis hlaviček i uložení do souboru.

Řešení od Dr.MM Lukáše Komy:

```
(defun http-request (
    &key
    (method :get)
    (url)
    (host)
    (port 80)
    (path)
    (headers)
    (content)
    (content-type)
    (out-filename)
    (print-headers)
    (max-redirects 100))
  (unless (and host path) ; Musime je ziskat z URL.
    (let ((host-port-and-path (parse-url url)))
      (setf host (car host-port-and-path))
      (setf port (nth 1 host-port-and-path))
      (setf path (nth 2 host-port-and-path))))
  (let (
      ; Nepouziva with-open-file, protoze soubor otviram jen kdyz je
    zadan.
```

```
      (file (when out-filename (open out-filename :direction :output :if
    -exists :supersede))))
    (unwind-protect
      (http-request-unguarded-out-stream
        method
        host
        port
        path
        (append ; Pokud byl Content-Type zadan jako argument, pridam jej
     do hlavicek
          (when content-type (list (list "Content-Type" content-type)))
          headers)
        content
        print-headers
        file
        max-redirects)
      (when out-filename (close file)))))
```

A takhle vypadá klient v akci:

```
; === Co to umí: ===

(format T "Stejna chovana jako wget v tematku~%~%")

; Stejne choveni jako wget v tematku
(http-request :host "httpbin.org" :path "/get" :out-filename "response1.
    json" :print-headers T)

(format T "~%------~%Reseni ulohy 5.1~%~%")

; Reseni ulohy 5.1
(defparameter *response-5.1* (http-request :url "httpbin.org/get" :out-
    filename "response2.json"))
(format T "Stavový kod: ~a~%" (get-status-code *response-5.1*))
(format T "Hodnota hlavicky Content-Type: ~a~%" (get-header "Content-
    Type" *response-5.1*))

(format T "~%------~%Reseni ulohy 5.2~%~%")

; Reseni ulohy 5.2

; Je-li zadan protokol na zacatku adresy, je ignorovan, protoze stejne
    umim jen jeden.
(format T "~a~%" (get-content (http-request :url "http://httpbin.org/get
    ")))

; Neni-li zadana cesta, je nastavena na "/".
(format T "~a~%" (get-content (http-request :url "example.com")))
```

```
; Podporuje i ruzné porty.
; Tento radek vyzaduje spusteny http server na portu 8000.
(http-request :url "localhost:8000" :method :head :print-headers T)

(format T "Ř~%------~%šeení úlohy 5.3~%~%")

; Reseni ulohy 5.3

; Absolutni presmerovani
(defparameter *url* "httpbin.org/absolute-redirect/3")
(defparameter *response-abs-redirects* (http-request :url *url*))
(format T "~a" *url*)
(loop for response in (get-previous-responses *response-abs-redirects*)
  do (format T " -> ~a" (get-header "Location" response)))

(format T "~%~%")

; V pripade prekroceni maximalniho poctu presmerovani se zachazi s
    posledni odpovedi, jako by byla
; finalni.
; Relativni presmerovani
(http-request :url "httpbin.org/relative-redirect/5" :max-redirects 3 :
    print-headers T)

(format T "~%------~%Reseni problemu 5.4~%~%")

; Reseni problemu 5.4

; HEAD
(http-request :url "httpbin.org/get" :method :head :print-headers T)

; POST
(format T "~%~a~%"
  (get-content
    (http-request
      :url "httpbin.org/post"
      :method :post
      :content "Tesim se na vikendovku."
      :headers (list (list "Content-Type" "text/plain")))))

; PUT
(format T "~a~%"
  (get-content
    (http-request
      :url "httpbin.org/put"
      :method :put
```

```
      :content "A jeste vic na pristi sous.")))

; PATCH
(format T "~a~%"
  (get-content
    (http-request
      :url "httpbin.org/patch"
      :method :patch
      :content "{\"op\":\"replace\",\"path\":\"/points\",\"value
     \":\"300\"}"
      :content-type "application/json-patch+json"))) ; Prida Content-
    Type mezi hlavicky.

; DELETE
(format T "~a~%" (get-content (http-request :url "httpbin.org/delete" :
    method :delete)))
```

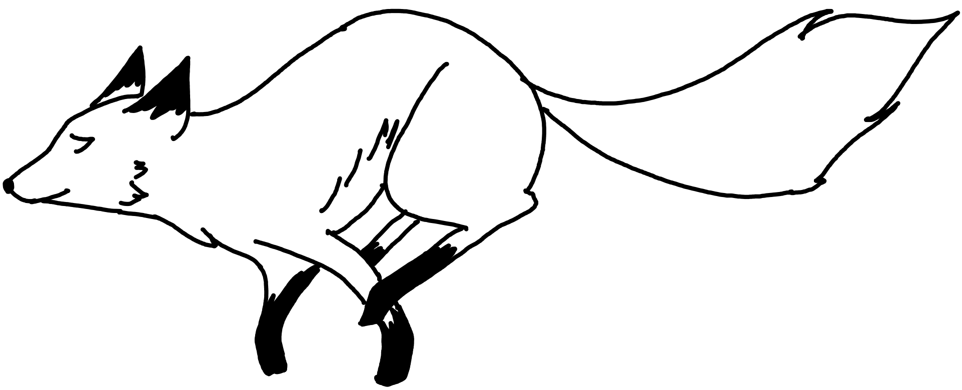
Problém 5.5 – HTTP server

Zadání:

Implementujte HTTP server, který bude schopný odpovědět naší `wget` *implementaci na dotazy. Obsah odpovědi je na vás.*

Dr.^MM^ Lukáš Koma zašel výrazně nad rámec zadání. Implementoval plnotučný HTTP/1 server, který umí zpracovávat UTF-8[9] a taky více požadavků najednou pomocí vláken. Server je napsaný velmi hezky a jeho zdrojový kód najdete na adrese `https://github.com/koskja/mam-lisp/blob/main/reseni5/http-server.lisp`. Doporučuji ho prozkoumat všem, kteří tématko řešili.

Jan Koška; `jan.koska@email.cz`



[9] https://en.wikipedia.org/wiki/UTF-8

Téma 7 – Vajíčko parašutista

Rozloučení

Milý řešiteli, za všechna vajíčka, která díky Tobě teď mohou seskakovat s padákem, bych ti chtěl poděkovat za řešení tohoto tématka. Šestým číslem sice 32. ročník končí, ale to určitě neznamená, že musí končit i tvá chuť experimentovat! Ať už půjde o úlohy v novém ročníku, či o tvou vlastní iniciativu, moc rádi budeme číst zprávy a články z tvých budoucích pokusů. Pokud bys chtěl podpořit svou chuť zkoumat, nebo by tě jen zajímalo, jak ostatní řešili toto tématko, můžeš si přečíst vzorová řešení níže.

Řešení 3. dílu

Úloha 3.1

Zadání:
Odvoď Newtonův odporový vzorec pomocí nastíněné rovnosti.

Následující řešení postupuje stejně jako řešení od Dr.$^{\mathsf{MM}}$ Petra Bartáka:
Řešení:
Klíčem k vyřešení této úlohy bylo uvažovat rovnost práce vykonané deskou a její kinetickou energií.

$$W = E_k$$

Dalším krokem bylo vyjádřit kinetickou energii pomocí hodnot spojených s letící deskou a s prostředím v kterém letí. Tedy $v = \frac{s}{t}$ a $m = \varrho V = \varrho SVt$, kde S je průmět plochy desky do směru jejího pohybu a je hustota prostředí. Nyní můžeme vyjádřit práci jako

$$W = \frac{1}{2}\varrho stv^3.$$

Po uvážení rovnosti $F = \frac{W}{s}$ dostaneme kýžený vztah

$$F = \frac{1}{2}\varrho Sv^2.$$

Úloha 3.2

Zadání:
Vyjádřete mezní rychlost v_m, pro kterou nastane rovnost mezi gravitační a odporovou silou.

Řešení:
Úloha 2 byla triviální a za sílu F stačilo dosadit tíhovou sílu $F = mg$. Mezní rychlost tedy vypočteme jako

$$v_m = \sqrt{\frac{2mg}{\varrho S}}.$$

Problém 3.3

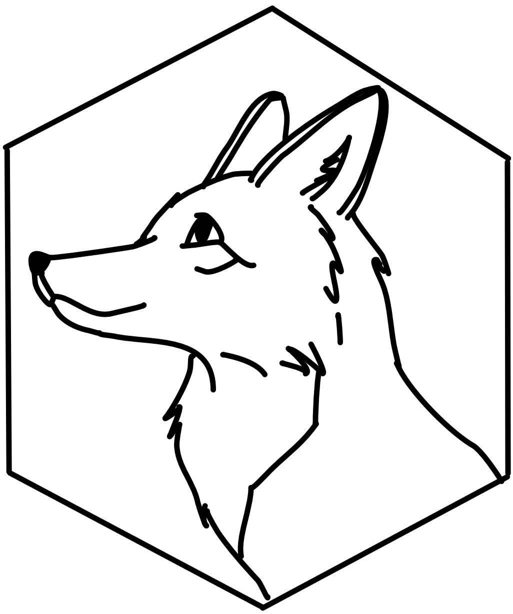
Zadání:

Nějakým způsobem udělejte edukovaný odhad, jakou maximální rychlostí smí vajíčko na nějaký povrch dopadnout, aby se nerozbilo. Určete velikost padáku vyrobeného z materiálu o hustotě ϱ_p, který by vajíčko zachránil při pádu z velké výšky. Uvažujte tvar padáku jako otevřenou polokouli, jejíž součinitel odporu je $C = 1{,}4$.

Řešení:

Problém 3.3 spočíval hlavně v určení mezní rychlosti, při které se vajíčko rozbije. Většina z vás problém řešila experimentálně a maximální, pro vajíčko bezpečná, výška vám vycházela kolem 20 cm. To však platilo jen pro tvrdé plochy jako třeba kuchyňská linka. Dr.$^{\mathrm{MM}}$ Lukáš Koma však své měření prováděl na trávníku, kde mu vyšla maximální bezpečná výška až 80 cm. Jako jediný také padák pro vajíčko opravdu sestrojil, za což bych mu chtěl pogratulovat. Jeho padák byl vyrobený z mikrotenových sáčků a oproti minimálnímu odhadnutému poloměru, který činil 11,5 cm, měl jím vyrobený padák poloměr přibližně 17 cm a vajíčko s přehledem zachránil z výšky až 4 m! Rovněž bych rád vypíchnul řešení Bc.$^{\mathrm{MM}}$ Mikuláše Hořenka, který problém atakoval přímým výpočtem potřebné kinetické energie k prasknutí vajíčka. Jeho výsledek maximální bezpečné výšky zhruba 16 cm výborně odpovídá hodnotám naměřeným ostatními řešiteli.

Honza Tregler; `jan.tregler@seznam.cz`



Konference Zlatá koruna 2026

Výroba papíru 10b

Dr.^{MM} Filip Dvořák, Bc.^{MM} Filip Gašparín, Bc.^{MM} Mikuláš Hořenek

Je pravděpodobné, že jste se již někdy pokoušeli něco nakreslit. Napadlo vás ale, že byste si ten papír, na který čmáráte, vyrobili sami? Či zda lze váš nepovedený obrázek znovu proměnit v neposkvrněný kousek čekající na božský dotek brku? Nebo jste třeba jen hloubali, z čeho všeho zajímavého by se dal papír udělat?

Právě toto jsme zkoumali v naší konfeře na letošním jarním soustředění M&M ve Zlaté Koruně. Zkusili jsme vyrobit papír z různých, někdy dost neobvyklých, materiálů a recyklovat ten starý. Suroviny jsme kombinovali, lisovali, sušili a získali mnoho rozličných kousků různých barev. Ke konci jsme zkoumali vlastnosti vyrobených papírů.

Teorie

Typická výroba papíru se skládá z několika základních kroků:

1. Příprava a rozmělnění surovin

   Toto je sám o sobě dosti komplexní proces, ovšem v domácích podmínkách ho není nutné rozebírat dopodrobna. Podstatné je, že výsledkem je jemná vodní suspenze vláknitého materiálu, jež je případně obohacena o přídavné látky (pojiva, barviva aj.).

2. Tvorba listu

   Pohybem síta ve směsi je na něj zachytáváno optimální množství papíroviny, čímž vzniká jakýsi zárodek listu (prvolist) ve formě tenké vrstvy takového vodnatého bláta.

3. Lisování

   Zde je většina vody mechanicky tlakem vypuzena a papír nabývá kompaktního tvaru.

4. Sušení

   Veškerá zbytková vlhkost (nad úroveň okolí) je odpařena.

V průmyslu pak je sekvence obohacena o další kroky, jako je válcování, klížení apod.; vzhledem k našim podmínkám jsme však od nich upustili.

Náš postup výroby

Nástroje a vybavení

Využívali jsme náčiní, které je běžně dostupné i doma. Potřebovali jsme: větší nádoby na vodu (nejlépe širší obdélníkové); tyčový mixér; jednovrstevné plátno (utěrka na nádobí, prostěradlo) – větší rozměry výhodou; tvrdé, rovné a ideálně hladké desky (pro lepší manipulaci s mokrým papírem); fén; nůž (cokoliv, čím se dá krájet/dlabat houba); improvizovaný lis na papír ve formě dvou rovných hladkých desek a vlastní tělesné tíhy; síto na papír (jediná účelově koupená věc, náklady v řádu nižších stovek, pro kutily domácí projekt).

Krok 0 – zisk vstupních surovin

Jako první jsme si sehnali vstupní suroviny. Vybrali jsme: použitý kancelářský papír; letákovinu (LIDL leták); karton z krabice. Kromě toho i materiály z přírody: suché listy; houbu, konkrétně troudnatec kopytovitý (*Fomes fomentarius*) – teoretizovali jsme, že chitin z buněčné stěny hub by mohl být využitelný podobně jako celulóza rostlin (oba jsou to polysacharidy) a držet papír pohromadě. Pokusili jsme se vyrobit papír i z kopřiv, ale to kvůli obtížnostem v kroku 2 (viz níže) nebylo možné. V určitých případech jsme zkusili použít PVA lepidlo jako pojivo.

Krok 1 – rozmělnění surovin

Do větší nádoby s vodou (dostatečný objem pro pojmutí nutného množství materiálu) jsme natrhali/nakrájeli/nažmoulali výchozí suroviny na menší kusy. Při procesu zvaném rozmňágání (náš vlastní vynález) jsme se ho snažili dále rozmělnit a efektivně dispergovat ve vodě. V případě papíru skoro stačilo jen směs žmoulat v rukou, pomohli jsme si ale tyčovým mixérem, který byl nezbytností pro jiné materiály. Byl velice nápomocný, a nebýt něj, nedosahoval by náš papír takových kvalit. Bohužel nebyl dostatečně výkonný, s papírem si lehce poradil, listí mu trvalo a troudnatec ho skoro zadusil (z technických důvodů byl na chvíli odstaven z provozu). Vlákna kopřiv namotaná na hřídel čepele ho zasekla, což ostatně znemožnilo jejich použití.

Výsledkem rozmňágání byla různě vodnatá kašovitá (blátivá) směs, v níž nebyly žádné větší částečky. Ty pouze v případě houby klesly ke dnu a v objemu vody plaval hnědavý kal. Takto připravené suroviny se daly hezky promísit.

Krok 2 – formace listu a jeho přenos na plátno

Do větší nádoby s dostačujícím objemem pro manipulaci se sítkem na papír (25×19 cm pro formát A5) jsme nalili rozmňáganou směs. Tu jsme různě kombinovali podle toho, o jaký papír mělo jít. K ní jsme dolili dostatek vody a pořádně ji promíchali, aby vznikla suspenze žádané koncentrace. Sítko jsme umístili ke dnu nádoby a zvedáním skrz objem kapaliny jsme nabrali materiál. Po vyzvednutí nad hladinu jsme nechali vytéct nadbytečnou vodu. Poté jsme jej převrátili a položili na utěrku.

Bylo nutné lepivou směs dostat ze sítka (ve kterém držela) na látku; toho jsme docílili *vyfrčkováním* – nadzvednutím jednoho okraje a přejížděním, dloubáním a cvrnkáním za pomoci prstů jsme odchlípli formující se list dolů. Také jsme si pomáhali proudem vzduchu z fénu. Bylo nutné toto provádět opatrně, aby se prvolist nepotrhal. Jakmile se jeden okraj odchlípl, koheze materiálu dále zajišťovala jakžtakž plynulý přenos.

Krok 3 – lisování

Po přenesení prvolistu jsme látku přeložili, aby jí byl z obou stran obklopen. Díky větším rozměrům utěrek se nám na jednu vešlo více vzorků, museli jsme si ovšem dávat pozor na dostatečné mezery pro ohýbání. Látku jsme posklápěli na velikost jednoho listu, při této činnosti byly velice nápomocné tvrdé desky pro podebírání navlhlé látky tak, aby se prvolist v ní nepotrhal.

Celý štůsek jsme odnesli do našeho improvizovaného lisu. To není ve své podstatě složité zařízení, je potřeba zajistit vodorovnou pevnou desku, na ni dát štůsek látky a nahoru přiložit další pevnou rovnou desku. Tu pak zatížit co nejvíce, v našem případě jsme použili zemskou tíži vlastních těl a poskakování, odhadem se nám tak podařilo vyvinout tlak až 42 kPa (spočítáno pomocí tělesné hmotnosti a rozměrů listu). Takto jsme vytlačili značné množství vody, čímž jsme urychlili proces schnutí a zpevnili formující se list.

Krok 4 – sušení

Vylisovaný štůsek jsme částečně rozložili a podle podmínek nechávali schnout na sluníčku, při nepřízni počasí vevnitř na vyhřívané podlaze koupelny. Po dni sušení bylo již možné vybrané listy papíru opatrně odebrat a sušit samostatně. Stále byly dost mokré, ale už držely při sobě a s pomocí jemné motoriky bylo možné je sundat z látky, aniž se potrhaly. Provedení rozvodu teplé vody objektu hrálo v náš prospěch, u stropu byly vedle sebe umístěné měděné trubky tak akorát, abychom na ně mohli položit listy papíru, jakž jsme i učinili. Po dalším (necelém) dni byl náš papír plně suchý. Teď už stačilo se jen pochlubit naším hotovým výrobkem.

Použité kombinace materiálů

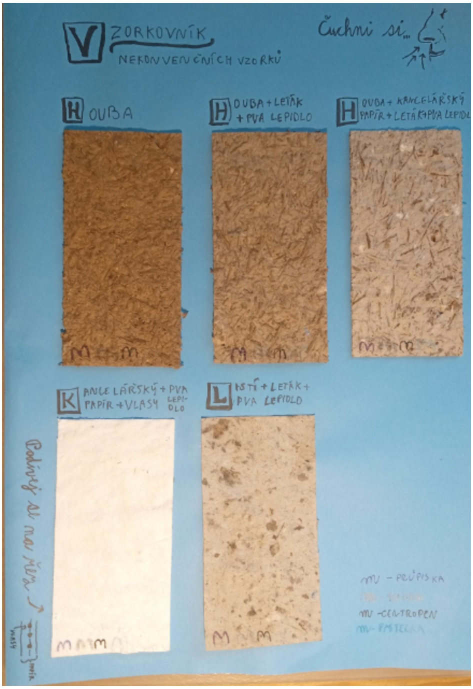
Začali jsme u prostého kancelářského papíru, letákoviny a kartonu. Tyto materiály již jsou samy o sobě papír, a tak nebylo moc pochyb o tom, že se z nich dá papír udělat. Taky se nám to povedlo, vyrobili jsme několik variant za použití čistých výchozích materiálů a dále jejich kombinací (kanc. papír + letákovina, všechny tři dohromady). Pro každou možnost jsme vytvořili dvě provedení, s lepidlem a bez. Zkusili jsme i jeden experimentální dvouvrstvý bílý papír, kdy na jeden prvolist jsme rozložili chomáč dlouhých vlasů, který jsme překryli další vrstvou. Vlasy jsou velice tenké, pevné a pružné, slibovali jsme si od toho zlepšení mechanických vlastností papíru (viz diskuze a závěr) formou vnitřní výztuže.

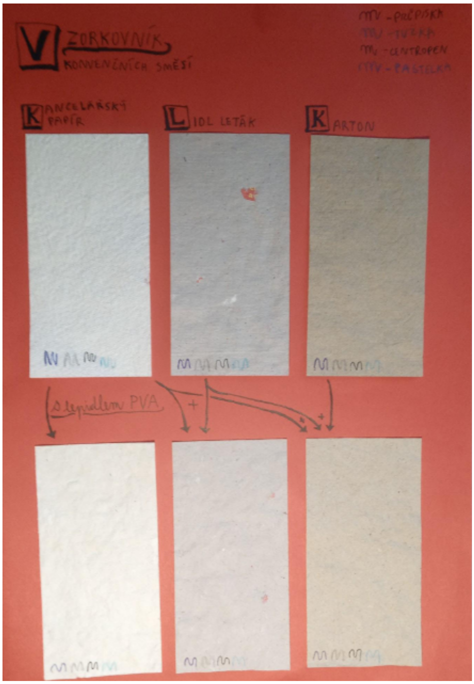
Pak jsme testovali neotřelejší materiál, a to výše zmiňovaný troudnatec kopytovitý, který bylo před mixováním potřeba nasekat. Většinu hmoty samotné houby tvořily výtrusné rourky v několika vrstvách (jedna přirůstá za rok), ty se daly docela dobře nadlabat nožem a krájet. Hmota houby byla celkem tvrdá a tuhá, avšak jednotlivé rourky se odlupovaly od sebe. Plstnatou masu z horní části klobouku jsme nepoužili, jelikož jí nebylo dostatečné množství a měli jsme obavu (zkušenostmi podloženou), že by tato tuhá soudržná hmota zadusila mixér. Troudnatcové rourky jsme použili jak samotné, tak s příměsí letákoviny a lepidla a také s bílým papírem, letákem a lepidlem.

Posledním (úspěšným, neb kopřivy se nezdařily) pokusem bylo suché listí, které jsme vzhledem k jeho tendenci vyplavat nad hladinu nechali něco přes den zatížené pod vodou. Směs po rozmixování a nabírání sítem nebyla moc kompaktní a nešlo by z ní udělat list papíru, a tak jsme přidali trochu letákoviny a pro druhou možnost i lepidlo.

Seznam kombinací materiálů

- kancelářský papír (+ lepidlo)
- letákovina (+ kancelářský papír + lepidlo)
- karton (+ kancelářský papír + letákovina + lepidlo)
- houba
- houba + letákovina + lepidlo
- houba + kancelářský papír + letákovina + lepidlo
- listí + letákovina + lepidlo
- kancelářský papír + vlasy + lepidlo





Výsledky

Hlavní cíl naší konfery byl splněn – vyrobili jsme papír! Rovněž se nám tak podařilo učinit i z pokusných materiálů (houba, listí). Potvrdili jsme hypotézu, že chitin z buněčné stěny hub by mohl být využitelný podobně jako celulóza rostlin, k níž nás vedla chemická podobnost látek.

Popis chování jednotlivých materiálů

Původně potištěný kancelářský papír byl po zpracování čistě bílý, takže by se na něj dalo znova hezky psát. Byl nejkvalitnější surovinou, stačilo přidat i malé množství, a výslednou směs výrazně zbělil. Letákovina, na začátku zcela barevně potištěná a zdánlivě na vyhození, byla po rozmňágání světle šedá a papír z ní vyrobený byl dobře použitelný na psaní. Tvořila malé a lepivé částečky, které rovněž našly uplatnění jako pojivo do směsí s většími kusy (houba, listí). Karton neprošel žádnou výraznější změnou, papír z něj byl nejtužší ze všech.

Papír z houby byl podstatně tlustší než ten z recyklovaných materiálů, avšak držel při sobě velmi dobře. Jeho vizáž (hnědý s černými jehličkami) neumožňovala příliš dobrou viditelnost psaného slova. Rovněž zajímavá byla jeho pěnovitá struktura na řezu. I menší příměs do papírové směsi dokázala výrazně zpevnit výsledný list, sám papír z čistých hub byl pevnější než karton. Hezké také bylo, že voněl.

Listí jako jediné nešlo použít samo o sobě (bez nějakého průmyslového řešení, které nemáme). Spolu s letákovinou bylo schopno tvořit celkem ohebný a použitelný papír, jenž ovšem nebyl úplně vhodný na psaní.

Použití lepidla nepřineslo valné výsledky, ba navíc ztěžovalo formaci prvolistu – kašovitá směs se více lepila na sítko. Výsledek byl jen o něco tužší a méně ohebný než ten bez lepidla.

Dvouvrstvý vlasový papír byl velice zajímavý; na pohled stejný jako běžný kancelářský papír (jen o trochu hrubší), ale při mechanickém dělení se projevily jeho unikátní vlastnosti. Vlasy mu dodaly pevnost a podstatně ztěžovaly stříhání, rovněž roztrhnout ho bylo náročnější. I když jich byla v papíře jen trocha, a to chaoticky rozmístěných, výrazně zlepšily jeho mechanické vlastnosti.

Na jeden vzorek poté, co byl zformován na plátně, jsme jako pokus umístili suché lískové jehnědy jako dekoraci. Během lisování se zapustily do struktury papíru a držely. Vzor byl pak hezky viditelný. Všechny papírovité vzorky byly jemně zvlněné. Ačkoliv by se to mohlo jevit jako vada, dodávalo to našim výrobkům řemeslný vzhled, což by jistě zvedlo cenu produktu jakožto pravého, podomácku vyrobeného, recyklovaného papíru.

Diskuze a závěr

Jak by šel náš postup vylepšit? Identifikovali jsme několik aspektů, na něž je radno se zaměřit. Jako první uveďme rozmělnění surovin – mixér opravdu nestačil pro ty náročnější. Zadruhé, lisování bylo v našich polních podmínkách silně suboptimální – vyšší tlak po výrazně delší dobu by znamenal kompatnější a kvalitnější výstup. Dále sušení – nekontrolované schnutí způsobovalo deformaci (výše zmíněné zvlnění). V neposlední řadě dodatečné zpracování – my jsme ho zanedbali, možná by mohlo pomoci.

Při práci jsme dokázali, že třídit papír má smysl. Z použitého šel docela snadno

vyrobit nový, kvalitní. Překvapivá byla výrazná proměna letákoviny umožňující opětovné použití. Takže ano, třiďte odpad.

Listí by se dalo uplatnit v papírenském průmyslu jako výplňová hmota. Kousky lístků se dobře lepily s papírem, takže kdyby se vyráběl nějaký spotřební papír, který by se už v budoucnu moc nedal recyklovat, přidaním listí by se zmenšily náklady na množství papíroviny, která by se dala využít dále někde jinde a jinak.

Co se týče papíru z hub, dospěli jsme k potvrzení hypotézy, že by se dal vyrábět a mohl by mít využití. Díky své pěnovité struktuře by mohl fungovat jako dobrý termoizolant; chtěli byste zateplení domu z polystyrenu nebo houbového papíru? Vzhledem ke schopnosti příměsi houboviny zvyšovat pevnost by houby mohly být použitelné coby přídavná výztužná hmota. Náš postup nebyl pro houbu nejvhodnější, nicméně ukázal potenciál. Průmyslové zpracování (jak mechanické, tak chemické) by z houboviny mohlo zvládnout udělat bílý a homogenní materiál s unikátními vlastnostmi.

Jak již bylo výše zmíněno, přidání vlasů zlepšuje papír mechanicky. Toho by se dalo využít v průmyslu pro získání potřebných vlastností. Navíc není zrovna nákladné si je opatřit, nebo alespoň jejich náhradu v podobě různých vláken. Mohly by se do vrstev papíru nanést v nějaké uspořádané mřížce, čímž by se normalizovaly mechanické vlastnosti. Výsledný kompozit by se dal více namáhat a snesl by větší zátěž.

Celá naše práce je snadno proveditelná doma. Bylo fascinující a nesmírně poutavé si něco takového zkusit, pro tvořivé lidi může tato činnost představovat úžasný odpočinek a zdroj hezkého papíru, který lze před lisováním různě ozdobit (vysušené květy, lišejník, …).

Výsledky 2. deadlinu 4. čísla a celého 5. čísla

Poř.	Jméno	R.	$\sum_{-1}$	Témata 1	26	5	6	O	$\sum_0$	$\sum_1$
1.	Prof.$^{\text{MM}}$ J. Klementová	4	509,6		4,5	9,8	4,0		18,3	148,5
2.	Dr.$^{\text{MM}}$ J. Thomitzek	1	133,2	11,0		23,0			34,0	133,2
3.	Dr.$^{\text{MM}}$ L. Koma	2	133,0	15,0		12,0	19,0		46,0	128,0
4.	Prof.$^{\text{MM}}$ M. Jarvis	4	502,5	6,8		10,9	12,5		30,2	127,2
5.	Dr.$^{\text{MM}}$ P. Barták	2	134,2		12,0	11,5			23,5	72,3
6.	Dr.$^{\text{MM}}$ S. Šimečková	4	101,2							58,7
7.	Mgr.$^{\text{MM}}$ M. Hrubá	2	52,3							52,3
8.	Doc.$^{\text{MM}}$ O. Nevěřil	4	315,2			15,5			15,5	50,8
9.	Dr.$^{\text{MM}}$ B. Salajová	4	169,3		3,0		0,5		3,5	49,1
10.	Mgr.$^{\text{MM}}$ Š. Swaczyna	1	84,6	1,0		14,4	4,2		19,6	47,4
11.	Bc.$^{\text{MM}}$ S. Bažantová	3	47,1	3,0		13,0			16,0	47,1
12.	Bc.$^{\text{MM}}$ R. Krzystek	3	46,6			18,5			18,5	46,6
13.–14.	Doc.$^{\text{MM}}$ M. Ambros	3	267,1							46,5
	Dr.$^{\text{MM}}$ F. Dvořák	3	101,6		4,5			6,0	10,5	46,5
15.–16.	Bc.$^{\text{MM}}$ P. Fiala	4	41,3	13,8					13,8	41,3
	Dr.$^{\text{MM}}$ P. Starý	4	155,6							41,3
17.	Bc.$^{\text{MM}}$ V. Kupilík	1	35,9	1,0	4,0	4,4			9,4	35,9
18.	Mgr.$^{\text{MM}}$ J. Fišerová	3	58,7							32,0
19.	Bc.$^{\text{MM}}$ M. Hořenek	3	30,6			4,9		6,0	10,9	30,6
20.	Bc.$^{\text{MM}}$ M. Vagner	3	29,5				14,0		14,0	29,5
21.	Bc.$^{\text{MM}}$ E. Ježek	4	28,8							28,8
22.	Dr.$^{\text{MM}}$ A. Gauchet	4	109,2			7,9			7,9	28,7
23.	Bc.$^{\text{MM}}$ T. Holásek	3	28,2							28,2
24.	Bc.$^{\text{MM}}$ F. Gašparín	3	26,3			4,9		6,0	10,9	26,3
25.	Mgr.$^{\text{MM}}$ F. Nouza	4	91,6							23,5
26.–28.	M. Stroff	4	19,8							19,8
	A. Mouchová	3	19,8							19,8
	V. Kubrycht	4	19,8							19,8
29.	L. Mihola	1	19,5							19,5
30.	Q. Liao	3	19,0							19,0
31.	V. Holuša	3	18,7			4,9			4,9	18,7
32.	O. Kočur	3	16,5			4,9			4,9	16,5
33.	Mgr.$^{\text{MM}}$ N. Jochová	3	62,2							15,1

Poř.	Jméno	R.	$\sum_{-1}$	Témata 1	26	5	6	O	$\sum_0$	$\sum_1$
34.–35.	M. Pavlas	2	15,0							15,0
	Mgr.$^{\text{MM}}$ V. Kučera	4	93,0							15,0
36.	M. Dvořák	3	14,9							14,9
37.	J. Štěchová	4	14,3							14,3
38.	J. Kaplický	4	18,6							13,9
39.	R. Stavarský	3	13,2		10,0	3,2			13,2	13,2
40.	A. Ježková	2	13,0							13,0
41.	Mgr.$^{\text{MM}}$ K. Bouchalová	1	61,6							12,0
42.	Dr.$^{\text{MM}}$ J. Jedlička	4	131,7							11,0
43.	T. Zatloukal	3	10,1							10,1
44.	Doc.$^{\text{MM}}$ D. Kaňka	4	211,9							10,0
45.	Dr.$^{\text{MM}}$ K. Kučerová	1	136,5							8,5
46.	D. Školař	1	7,7			7,7			7,7	7,7
47.–48.	M. Hošek	4	7,0							7,0
	Š. Hrdý	4	7,0							7,0
49.	Mgr.$^{\text{MM}}$ S. Ožanová	4	56,3							6,0
50.	R. Michálková	4	10,6							5,5
51.	F. Jančík	1	5,4	1,0		4,4			5,4	5,4
52.	K. Kučerová	1	3,0							3,0
53.–55.	L. Šemberová	Z9	1,2							1,2
	O. Plíšek	1	1,2							1,2
	M. Vojtěch	Z8	1,2							1,2
56.	J. Dingová	4	1,0							1,0
57.	J. Vospálek	3	0,6							0,6

Sloupeček $\sum_{-1}$ je součet všech bodů získaných v našem semináři, $\sum_0$ je součet bodů v těchto deadlinech a $\sum_1$ součet všech bodů v tomto ročníku. Sloupec **O** symbolizuje **Ostatní**, obvykle příspěvky za články. Tituly uvedené v předchozím textu slouží pouze pro účely M&M.

Výsledková listina 32. ročníku

				Číslo						
Poř.	**Jméno**	**R.**	$\sum_{-1}$	**1**	**2**	**3**	**4**	**5**	**6**	$\sum_{1}$
1.	Prof.$^{\mathsf{MM}}$ J. Klementová	4	509,6	64,0	17,7	35,5	21,5	9,8		148,5
2.	Dr.$^{\mathsf{MM}}$ J. Thomitzek	1	133,2	43,0	25,7	30,5	13,0	21,0		133,2
3.	Dr.$^{\mathsf{MM}}$ L. Koma	2	133,0	13,0	25,0	36,0	26,0	28,0		128,0
4.	Prof.$^{\mathsf{MM}}$ M. Jarvis	4	502,5	68,0	10,0	19,0	26,7	3,5		127,2
5.	Dr.$^{\mathsf{MM}}$ P. Barták	2	134,2	14,5	13,7	20,6	21,8	1,7		72,3
6.	Dr.$^{\mathsf{MM}}$ S. Šimečková	4	101,2	27,0	13,7	18,0				58,7
7.	Mgr.$^{\mathsf{MM}}$ M. Hrubá	2	52,3	46,0	3,3	3,0				52,3
8.	Doc.$^{\mathsf{MM}}$ O. Nevěřil	4	315,2	10,0	9,3	16,0	10,0	5,5		50,8
9.	Dr.$^{\mathsf{MM}}$ B. Salajová	4	169,3	16,8	6,0	22,8	3,5			49,1
10.	Mgr.$^{\mathsf{MM}}$ Š. Swaczyna	1	84,6	15,0	4,5	8,3	14,2	5,4		47,4
11.	Bc.$^{\mathsf{MM}}$ S. Bažantová	3	47,1	8,0	6,5	16,6	10,0	6,0		47,1
12.	Bc.$^{\mathsf{MM}}$ R. Krzystek	3	46,6	13,9	6,2	8,0	13,0	5,5		46,6
13.–14.	Doc.$^{\mathsf{MM}}$ M. Ambros	3	267,1	9,0	1,0	36,5				46,5
	Dr.$^{\mathsf{MM}}$ F. Dvořák	3	101,6	17,5	11,5	7,0	4,5	6,0		46,5
15.–16.	Bc.$^{\mathsf{MM}}$ P. Fiala	4	41,3	11,0	5,5	11,0	13,8			41,3
	Dr.$^{\mathsf{MM}}$ P. Starý	4	155,6	41,3						41,3
17.	Bc.$^{\mathsf{MM}}$ V. Kupilík	1	35,9			26,5	4,0	5,4		35,9
18.	Mgr.$^{\mathsf{MM}}$ J. Fišerová	3	58,7	32,0						32,0
19.	Bc.$^{\mathsf{MM}}$ M. Hořenek	3	30,6			19,7	4,9	6,0		30,6
20.	Bc.$^{\mathsf{MM}}$ M. Vagner	3	29,5			15,5	14,0			29,5
21.	Bc.$^{\mathsf{MM}}$ E. Ježek	4	28,8	28,8						28,8
22.	Dr.$^{\mathsf{MM}}$ A. Gauchet	4	109,2	4,5	8,3	8,0	7,9			28,7
23.	Bc.$^{\mathsf{MM}}$ T. Holásek	3	28,2	28,2						28,2
24.	Bc.$^{\mathsf{MM}}$ F. Gašparín	3	26,3			15,4	4,9	6,0		26,3
25.	Mgr.$^{\mathsf{MM}}$ F. Nouza	4	91,6	11,0		12,5				23,5
26.–28.	M. Stroff	4	19,8	19,8						19,8
	A. Mouchová	3	19,8	19,8						19,8
	V. Kubrycht	4	19,8	19,8						19,8
29.	L. Mihola	1	19,5	9,5		10,0				19,5
30.	Q. Liao	3	19,0	19,0						19,0
31.	V. Holuša	3	18,7			13,8	4,9			18,7
32.	O. Kočur	3	16,5			11,6	4,9			16,5
33.	Mgr.$^{\mathsf{MM}}$ N. Jochová	3	62,2	9,3	5,8					15,1
34.–35.	M. Pavlas	2	15,0		4,0	11,0				15,0

Poř.	Jméno	R.	$\sum_{-1}$	Číslo 1	2	3	4	5	6	$\sum_{1}$
	Mgr.$^{\mathsf{MM}}$ V. Kučera	4	93,0	15,0						15,0
36.	M. Dvořák	3	14,9			14,9				14,9
37.	J. Štěchová	4	14,3		3,8	10,5				14,3
38.	J. Kaplický	4	18,6	6,0		7,9				13,9
39.	R. Stavarský	3	13,2				13,2			13,2
40.	A. Ježková	2	13,0	13,0						13,0
41.	Mgr.$^{\mathsf{MM}}$ K. Bouchalová	1	61,6	12,0						12,0
42.	Dr.$^{\mathsf{MM}}$ J. Jedlička	4	131,7	9,0	2,0					11,0
43.	T. Zatloukal	3	10,1			10,1				10,1
44.	Doc.$^{\mathsf{MM}}$ D. Kaňka	4	211,9	10,0						10,0
45.	Dr.$^{\mathsf{MM}}$ K. Kučerová	1	136,5	8,5						8,5
46.	D. Školař	1	7,7				7,7			7,7
47.–48.	M. Hošek	4	7,0	7,0						7,0
	Š. Hrdý	4	7,0	7,0						7,0
49.	Mgr.$^{\mathsf{MM}}$ S. Ožanová	4	56,3			6,0				6,0
50.	R. Michálková	4	10,6			5,5				5,5
51.	F. Jančík	1	5,4					5,4		5,4
52.	K. Kučerová	1	3,0	3,0						3,0
53.–55.	L. Šemberová	Z9	1,2		1,2					1,2
	O. Plíšek	1	1,2		1,2					1,2
	M. Vojtěch	Z8	1,2		1,2					1,2
56.	J. Dingová	4	1,0	1,0						1,0
57.	J. Vospálek	3	0,6		0,6					0,6

Kontakty:

M&M, OPMK, MFF UK
Ke Karlovu 3
121 16 Praha 2

E-mail: `mam@matfyz.cz`
Web: `mam.matfyz.cz`
FB: `casopis.MaM`

matfyz M&M