STUDENTSKÝ ČASOPIS A KORESPONDENČNÍ SEMINÁŘ

Ročník XXXI
Číslo 2

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 řešitelko, milý řešiteli,

konečně je tady druhé číslo tvého oblíbeného časopisu M&M! Hned na začátku najdeš pokračování tématka *Vektory a matice*, ve kterém se tentokrát dozvíš něco o lineárních kombinacích a soustavách rovnic. Jidášova tématka *Programování v TEXu a LATEXu* a *Typografie* také pokračují a obě budou tentokrát o textu. V tom prvním se naučíš text správně skládat do řádků a sloupců a v tom druhém se dozvíš, co jsou to fonty a jejich řezy a k čemu se hodí. V rámci pravopisné části tématka *Typografie* vám navíc Dr.$^{\mathsf{MM}}$ Dominik Kaňka ve svém článku stručně vysvětlí, jak se v češtině správně píšou číslovky.

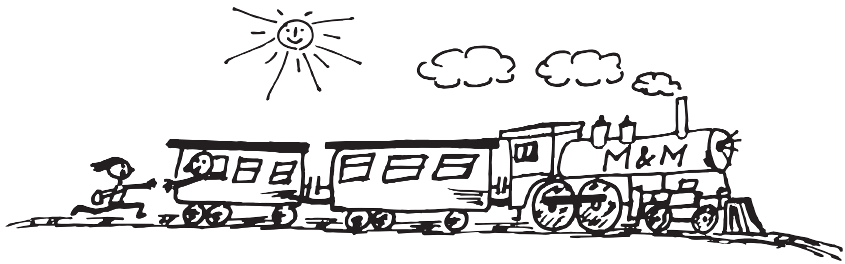
Kromě toho nám v tomto čísle přibyla hned dvě nová tématka. V tématku *Speciální teorie relativity* se naučíš ohýbat čas i prostor jenom pomocí matematických rovnic a v tématku *Strojíme matematiku puntíky a šipkami* budeš (překvapivě) budovat matematiku od začátku, ale pozor, úplně jinak než to dělá teorie množin!

A než se konečně vrhneš do řešení, máme pro tebe ještě dvě pozvánky. Ve čtvrtek 28. listopadu se koná v Praze na Matfyzu den otevřených dveří[1], kde si můžeš projít stánky různých částí Matfyzu a prohlédnout si na vlastní oči, co se tady vlastně děje.

No a v prosinci se jako již tradičně uskuteční M&Mí vánoční víkendovka, tentokrát v Hlinsku v Pardubickém kraji. Rezervuj si termín 6.–8. 12., těšit se můžeš na deskovky, kytarování, výlet a povídání s ostatními orgy i řešiteli. Přihlašovací formulář se ve správný čas objeví ve tvé e-mailové schránce spolu s dalšími podrobnými informacemi.

Krásný podzim ti přejí

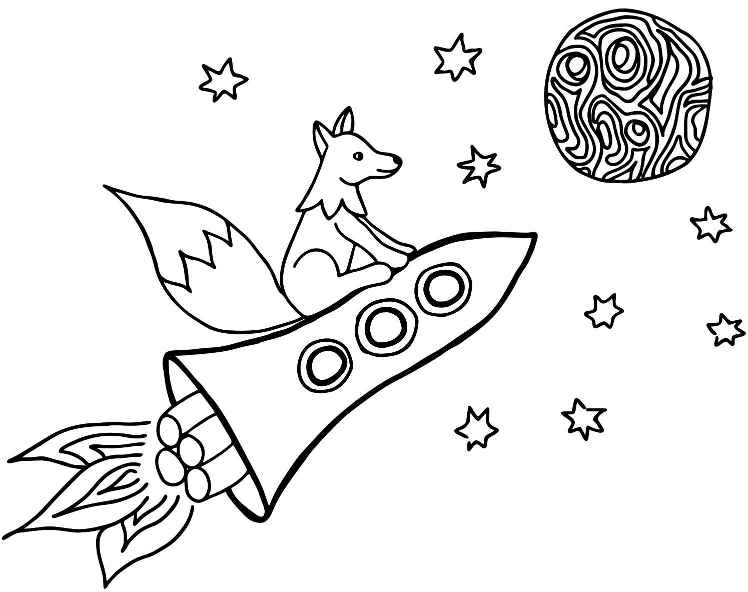
Tví organizátoři



P.S.: Týden před M&Mí víkendovkou (tedy 29. 11. – 1. 12.) pořádá spřátelený seminář KSP KSP. To není překlep, KSP značí nejen *Korespondenční seminář z programování*, ale také *Krutou Smršť Přednášek*, tedy víkendovku nacpanou (vesměs, ale ne výhradně informatickými) přednáškami od sklepa až na půdu. Více informací najdete přímo na stránce Smrště[2].

[1] `https://www.mff.cuni.cz/cs/uchazeci/dny-otevrenych-dveri/2024`
[2] `https://ksp.mff.cuni.cz/akce/smrst/2024/`

Obsah



Zadání a řešení témat

1. deadline: 12. listopadu 2024 | 2. deadline: 10. prosince 2024

Téma 2 – Vektory a matice

V tomto díle si ukážeme, že matice jsou navíc i užitečný nástroj pro popisování soustav rovnic, jejichž řešení nám pak snadno umožní dělat lineární transformace „pozpátku“. Postupně si kolem toho dokážeme několik tvrzení, některé důkazy budou zadány jako úlohy. Slibujeme, že *všechna tvrzení platí*, takže v řešení úloh smíte používat i tvrzení z předchozích úloh, které nemáte vyřešené.

Začneme užitečným pozorováním o násobení matic:

Úloha 2.1 [2b]: *Ukažte, že maticové násobení je asociativní, tedy že pro jakékoliv tři matice* $\mathbf{A}, \mathbf{B}, \mathbf{C}$ *platí* $(\mathbf{A} \cdot \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot (\mathbf{B} \cdot \mathbf{C})$ *(pokud je součin definován).*

Úloha 2.2 [2b]: *Ukažte, že maticové násobení je distributivní, tedy že pro jakékoliv tři matice* $\mathbf{A}, \mathbf{B}, \mathbf{C}$ *platí* $\mathbf{A} \cdot (\mathbf{B} + \mathbf{C}) = \mathbf{A} \cdot \mathbf{B} + \mathbf{A} \cdot \mathbf{C}$ *(pokud jsou součiny a součty definovány). (Stejně se dá ukázat, že* $(\mathbf{A} + \mathbf{B}) \cdot \mathbf{C} = \mathbf{A} \cdot \mathbf{C} + \mathbf{B} \cdot \mathbf{C}$*)*

Ještě se nám bude možná hodit operace *transpozice*: pro matici $\mathbf{A} \in \mathbb{R}^{m \times n}$ má její transpozice $\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{n \times m}$ prvky $a^{\mathrm{T}}_{ij} = a_{ji}$. Jedná se tedy o „překlopení“ matice podle hlavní diagonály (z levého horního rohu šikmo doprava dolů). Matice tedy prohodí rozměry a přeskládá prvky, čímž se z řádků stanou sloupce a naopak. Symbolicky tedy:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \qquad \mathbf{A}^{\mathrm{T}} = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{pmatrix}$$

A na konkrétním příkladu:

$$\mathbf{A} = \begin{pmatrix} 2 & 4 & 3 \\ -3 & 0 & 1 \\ 5 & 7 & -11 \\ 1 & -1 & 0 \end{pmatrix} \qquad \mathbf{A}^{\mathrm{T}} = \begin{pmatrix} 2 & -3 & 5 & 1 \\ 4 & 0 & 7 & -1 \\ 3 & 1 & -11 & 0 \end{pmatrix}$$

Zatím nebudeme prozrazovat, jak se transpozice chová k různým dřív zavedeným operacím, ale jistě budeme mít několik pozorování o chování sloupců, které budeme chtít aplikovat na řádky, a opačně; a v tom nám transpozice může pomoci.

Lineární kombinace

S lineárními kombinacemi jsme se setkali už minule v části o vyjadřování jedněch vektorů pomocí vektorů jiných, jen jsme tu operaci nepojmenovali. Teď to napravíme: *lineární kombinace* vektorů $\vec{v}_1, \dots, \vec{v}_n$ s koeficienty $\alpha_1, \dots, \alpha_n$ (reálná čísla) je vektor $\vec{v}$ daný předpisem

$$\vec{v} = \sum_{i=1}^{n} \alpha_i \vec{v}_i = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \cdots + \alpha_n \vec{v}_n.$$

Protože rozepisovat součty je trošku nepraktické, můžeme vektory zapsat jako sloupce matice a tuto matici zprava násobit vektorem koeficientů:

$$\vec{v} = \begin{pmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_n \\ | & | & & | \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}$$

Jako příklad jsme minule viděli, že vektor $\binom{7}{5}$ jde vyjádřit jako $7 \cdot \binom{1}{0} + 5 \cdot \binom{0}{1} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) \cdot \binom{7}{5}$. Stejně tak si můžeme všimnout, že z vektorů $\left(\begin{smallmatrix} 4 \\ 1 \\ 0 \end{smallmatrix}\right)$ a $\left(\begin{smallmatrix} 1 \\ -4 \\ 0 \end{smallmatrix}\right)$ vyjádřit vektor $\left(\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right)$ *nejde.*

To, jestli jde nebo nejde některý vektor vyjádřit jako lineární kombinace ostatních, se v tomhle díle ukáže jako užitečná vlastnost. Zavedeme pro to tedy formální definici a pojem: máme-li množinu alespoň dvou[3] vektorů $V = \{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$, řekneme, že tyto vektory jsou *lineárně závislé*, pokud některý vektor $\vec{v}_k \in V$ jde vyjádřit jako lineární kombinace ostatních vektorů $V \setminus \{\vec{v}_k\}$. Pokud žádný z vektorů nejde vyjádřit pomocí ostatních, jsou tyto vektory *lineárně nezávislé.*

Například, jak jsme viděli výše a minule, vektory $\binom{1}{0}, \binom{0}{1}, \binom{7}{5}$ jsou lineárně závislé: kromě zmíněného vyjádření vektoru $\binom{7}{5}$ jde vyjádřit třeba

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{7} \cdot \begin{pmatrix} 7 \\ 5 \end{pmatrix} + \frac{-5}{7} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Pro názornost ještě zmiňme, že vektory nemusí jít vyjádřit všechny. Vektory: $\left\{ \left(\begin{smallmatrix} 4 \\ 1 \\ 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 8 \\ 2 \\ 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 0 \\ 0 \\ 1 \end{smallmatrix}\right) \right\}$ lineárně závislé *jsou* (druhý je dvojnásobek prvního a nulanásobek třetího), přestože třetí vektor z předchozích vyjádřit nejde. Navíc si povšimněme, že každá množina obsahující nulový vektor je lineárně závislá.

Úloha 2.3 [2b]: *Najděte dvě různé lin. kombinace vektorů* $\left(\begin{smallmatrix} 1 \\ 2 \\ 2 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} -1 \\ 0 \\ -8 \end{smallmatrix}\right)$ *a* $\left(\begin{smallmatrix} 1 \\ -2 \\ 14 \end{smallmatrix}\right)$, *které dají výsledný vektor* $\left(\begin{smallmatrix} 3 \\ 5 \\ 9 \end{smallmatrix}\right)$.

Zajímavé matice

Teď od lineárních kombinací na chvíli odběhneme k zajímavým maticím, ale ke konci tohoto dílu se ukáže, že spolu úzce souvisí. Zajímavost matic bude typicky spočívat v jejich chování při maticovém násobení.

[3] Požadavek na aspoň dva vektory nám ušetří potřebu dodefinovat, jak se chovají prázdné lineární kombinace a jestli je prázdná množina lineárně nezávislá.

Jednotková matice

Jednotková matice tvaru $d \times d$ se značí $\mathbf{I}_d$ a vypadá následovně:

$$\mathbf{I}_d = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

Často víme, kterou jednotkovou maticí chceme násobit, takže jako zkratku budeme používat prostě $\mathbf{I}$ pro jednotkovou matici tvaru, *který zrovna jde použít* (vždycky je čtvercová).

Když jednotkovou maticí $\mathbf{I}$ násobíme libovolnou matici, původní matice se nezmění, tj. pro matici $\mathbf{A} \in \mathbb{R}^{m \times n}$ je $\mathbf{I}_m\mathbf{A} = \mathbf{A}\mathbf{I}_n = \mathbf{A}$.

Regulární matice

Regulární matice je každá čtvercová matice, která má lineárně nezávislé sloupce[4]. U nečtvercových matic tuhle vlastnost nezkoumáme, protože užitečnost regulárních matic je daná jak čtvercovitostí, tak lineární nezávislostí sloupců, jak si dokážeme v pozdějším dílu.)

Úloha 2.4 [3b]: *Nechť $\mathbf{A}$ je regulární matice. Ukažte, že z toho, co je napsáno výše, plyne, že nemohou existovat různé vektory $\vec{x}$ a $\vec{y}$, pro něž je $\mathbf{A}\vec{x} = \mathbf{A}\vec{y}$.*

Inverzní matice

Připomeňme si úlohu 1.3 z minulého čísla – v té jsme hledali matici $\mathbf{A}$ tak, aby $\left(\begin{smallmatrix}1&2\\3&4\end{smallmatrix}\right)\cdot\mathbf{A} = \left(\begin{smallmatrix}5&13\\19&25\end{smallmatrix}\right)$. Kdyby se jednalo o klasickou rovnici, bylo by přirozené soustavu vydělit $\left(\begin{smallmatrix}1&2\\3&4\end{smallmatrix}\right)$, čímž bychom dostali vyjádření $\mathbf{A} = \left(\begin{smallmatrix}5&13\\19&25\end{smallmatrix}\right)/\left(\begin{smallmatrix}1&2\\3&4\end{smallmatrix}\right)$. Můžeme něco takového ale dělat s maticemi?

Odpověď zní, že někdy ano. Stejně jako u klasické rovnice si musíme ale dát pozor na několik věcí:

- Ne každou maticí půjde „dělit“ (podobně jako nejde dělit nulou).
- Některé operace s rovnicí nám mohou přidat další řešení – takové operace buď nechceme dělat, nebo chceme vědět, že je děláme (podobně jako vynásobení obou stran nulou).

Nicméně v běžném případě dělit celou rovnici jde, a podobně to bude u matic. Konkrétně, násobit strany regulární maticí nám soustavu nerozbije ve smyslu dvou odrážek výše. Pozor na to, že maticové násobení (pořád ještě) není komutativní, takže musíme buď obě strany násobit zleva, nebo zprava.

[4] Čtvercové matice, které podmínku nesplňují, se nazývají *singulární*.

Zatím ale pořád nevíme, jak s takovou možností naložit – zatím umíme jen „přidávat“ matice na obě strany, ale my je chceme odebírat... Chceme proto najít matici $\mathbf{B}$, kterou když vynásobíme matici $\mathbf{A}$, vyjde nám jednotková matice.

Úloha 2.5 [2b]: *Ukažte, že v tomhle případě nezáleží na pořadí, tedy pokud je* $\mathbf{A}$ *regulární a* $\mathbf{AB} = \mathbf{I}$*, pak i* $\mathbf{BA} = \mathbf{I}$*.*

To pořád není úplně zlepšení: k vyřešení úlohy 1.3 teď chceme využít schopnost vyřešit úlohu 1.3', kdy pravá strana bude jednotková matice...

No, tak už to rozsekneme: protože se nám bude často hodit umět se zbavovat matic, tak naše matice $\mathbf{B}$ má speciální jméno: *matice inverzní k matici* $\mathbf{A}$, a proto se typicky značí $\mathbf{A}^{-1}$.

K hledání inverzních matic lze použít *Gaussovu–Jordanovu eliminaci*: k matici $\mathbf{A}$ „přilepíme“ jednotkovou matici (typicky zprava). Na příkladu z úlohy 1.3:

$$(\mathbf{A}|\mathbf{I}) = \left(\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{array}\right)$$

Následně pomocí *elementárních řádkových úprav* rozšířenou matici soustavy předěláváme tak, abychom na její levé straně dostali jednotkovou matici – vznikne $\mathbf{I}|\mathbf{B}$. Tvrdíme, že $\mathbf{B}$ je inverzní k $\mathbf{A}$.

A co že jsou elementární řádkové úpravy? Můžeme:

- vyměnit dva řádky.
- řádek vynásobit reálným číslem α, kdy $\alpha \neq 0$ (každé číslo v řádku se zvětší α-krát).
- k i-tému řádku přičíst α-násobek j-tého řádku (přičemž α může být 1).

Problém 2.6: *Ukažte, že každá elementární řádková úprava jde vyjádřit jako násobení zleva vhodnou maticí („maticí úpravy“). Zkuste ukázat, že matice úprav jsou regulární.*

V principu můžeme dělat libovolné řádkové úpravy v libovolném pořadí, a pokud se nám náhodou podaří dosáhnout cílového stavu, tak jsme vyhráli. Nicméně se osvědčil následující postup, kdy matici z levého horního rohu postupně přetváříme na jednotkovou.

Postupně procházíme řádky od prvního. Pokud je nejlevější nenulový prvek v některém nižším řádku víc vlevo než v aktuálním řádku, tyto řádky prohodíme. Označíme tuto nejlevější nenulu jako *pivot*. Tento potenciálně vyměněný řádek vydělíme hodnotou pivota, čímž se z něj stane jednička (a vlevo od ní jsou jen nuly). To nám snadno umožní vynulovat ostatní hodnoty ve sloupci s pivotem: ke každému jinému řádku i přičteme $(-a_{ip})$-násobek aktuálního řádku, kde p je číslo sloupce s pivotem. (Všimněte si, že na levější hodnoty to nemá vliv, protože k nim přičítáme jen nuly.) Posuneme se o řádek níž a postup opakujeme – nejlevější nenula je teď ve sloupci vpravo, zatím bez důkazu tvrdíme, že pro regulární matici bude vždy hned v tom dalším sloupci. Takto postupně odleva vyrábíme správné sloupce hledané jednotkové matice.

Úloha 2.7 [1b]: *Zdůvodněte, že postup skutečně hledá inverzní matici.*

Všimněte si několika dalších užitečných vlastností (můžete předpokládat, že inverzní matice je také regulární).

Úloha 2.8 [2b]: *Ukažte, že:*

- *každá regulární matice má nejvýš jednu inverzní matici.*
- *matice může být inverzní k nejvýš jedné matici. (Hint: co by se stalo, pokud by takové matice existovaly dvě?)*
- *inverzní matice k inverzní matici je původní matice.*

Soustavy rovnic

V minulém čísle jsme si matice popsali jako „tabulky plné čísel“, což nám pro předchozí účely stačilo. Existuje několik způsobů, kterými se na matice můžeme podívat. V tomto čísle si ukážeme, že matice pro nás může být i způsobem, jak zapsat soustavu lineárních rovnic. Soustavu

$$\begin{aligned} 2x + 6y + 5z &= 0 \\ 3x + 5y + 18z &= 33 \\ x + 2y + 5z &= 8 \end{aligned} \quad \text{lze napsat taktéž jako} \quad \left(\begin{array}{ccc|c} 2 & 6 & 5 & 0 \\ 3 & 5 & 18 & 33 \\ 1 & 2 & 5 & 8 \end{array}\right).$$

Užitečnost tohoto zápisu se ukáže, když na matici rovnice použijeme Gaussovu–Jordanovu eliminaci. Velmi se hodí, že elementární řádkové úpravy přesně odpovídají povoleným operacím při řešení soustav rovnic součtovou metodou, takže vždycky můžeme přepsat soustavu do klasického rovnicového tvaru a dostat platné rovnosti.

Nicméně tentokrát jsou možné i různé další výsledky:

- Pokud vyjdou nějaké nulové řádky, tak je prostě dál nebudeme uvažovat.

1. Vyjde $\mathbf{I}|\vec{c}$. Pak přepsáním do rovnicového tvaru rovnou máme rovnosti s výsledky: $x = c_1$, $y = c_2$, ... To je ten příznivý případ. Tento případ nastává, pokud je matice soustavy regulární. Příklad:

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 2 & 3 & 6 & 4 \\ 3 & 5 & 8 & 4 \end{array}\right) \sim \ldots \sim \left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 1 \end{array}\right), \text{řešení je tedy} \quad \begin{aligned} x &= 2 \\ y &= -2 \\ z &= 1 \end{aligned}$$

2. V některém řádku vyjdou před svislítkem nuly a za svislítkem nenula ℓ. To odpovídá rovnosti $0 = \ell$, tedy soustava nemá řešení.

3. Výpočet „uskočí“ a některý sloupec se nevynuluje (žádný pivot se v tomhle sloupci nikdy nenacházel – tedy nastala přesně ta situace, o které jsme v algoritmu výše bez důkazu slíbili, že pro regulární matice nenastane). Pak má soustava rovnic nekonečně mnoho řešení, protože si pro proměnnou odpovídající nevynulovanému sloupci (resp. proměnné, pokud jich je víc) můžeme hodnotu zvolit libovolně a vždy dostaneme platný vztah mezi těmito proměnnými a pivotickou proměnnou. (Přímočaré řešení je prostě za takové proměnné dosadit nulu, kdy pro pivotickou proměnnou vyjde pravá strana matice.)

   Pro ukázku uvažme následující průběh Gaussovy–Jordanovy eliminace (rozmyslete si, jak vypadala původní soustava rovnic pro proměnné x, y, z):

$$\left(\begin{array}{rrr|r} 1 & -2 & 2 & 4 \\ 1 & -1 & 3 & 6 \\ 0 & 1 & 1 & 2 \end{array}\right) \sim \ldots \sim \left(\begin{array}{rrr|r} 1 & 0 & 4 & 8 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right).$$

   Z výsledné matice vyčteme, že proměnná z může nabývat libovolné reálné hodnoty; tu označme t. Pak platí $y = 2 - t$, $x = 8 - 4t$. Příklad jednoho konkrétního řešení třeba pro $t = 0$ je $\left(\begin{smallmatrix} 8 \\ 2 \\ 0 \end{smallmatrix}\right)$.

Ještě si všimněme jedné věci: Pokud původní matice soustavy byla $\mathbf{A}|\vec{b}$ a odmyslíme si, jaká přesně byla původní písmenka proměnných, můžeme též soustavu napsat pomocí maticového násobení jako: $\mathbf{A} \cdot \vec{x} = \vec{b}$, kde $\vec{x}$ je vektor proměnných:

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \text{ resp. v našem případě } \vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

Tento zápis nám však nápadně může připomínat maticový zápis lineární kombinace. A skutečně, vlastně se v soustavě ptáme na to, jak vypadají koeficienty lineární kombinace sloupců levé strany soustavy, které dají konkrétní vektor na pravé straně soustavy. A navíc, pro regulární matice můžeme rovnou spočítat vektor výsledků $\vec{x} = \mathbf{A}^{-1} \cdot \vec{b}$ (byť si Gaussovu–Jordanovu eliminaci neušetříme).

Úloha 2.9 [1,5b]: *Vypočítejte Gaussovou–Jordanovou eliminací následující soustavu rovnic:*

$$\left(\begin{array}{ccc|c} 2 & 6 & 5 & 0 \\ 3 & 5 & 18 & 33 \\ 1 & 2 & 5 & 8 \end{array}\right)$$

V příštím díle se opět podíváme na obrázkové chápání regulárních matic, zavedeme souřadnice a budeme se bavit o lineárních prostorech. A také se můžete těšit na vzorová řešení z obou dílů tématka.

Pavel a Marian; `lingebra@ledoian.cz`
odevzdávejte do odevzdávátka

Téma 3 – Programování v TEXu a LATEXu

V minulém díle jsme si ukázali, jak v TEXu definovat makra, používat registry typu `\count` a pracovat s podmínkami, a hlavně jsme si řekli, co jsou tokeny a jejich kategorie (tj. jak TEX „vidí“ to, co my napíšeme) a jak pomocí změny kategorie znaku donutit TEX, aby vnímal svět jinak. Tohle všechno si zopakujte ve vzorových řešeních na straně 16.

Zatím tedy umíme z toho, co TEXu předloží člověk, získat nějaké povely, které buď mění to, jak funguje tento překlad[5] (např. `\catcode` nebo `\def`), a přenastavují hodnoty registrů (např. `\newcount\cislo \cislo=1`), nebo něco vykreslují do výsledného PDF (jistě jste si všimli, že písmena a čísla, pokud neznamenají něco jiného, se nějak vypisují). Získáváme je tím způsobem, že nejprve[6] znaky (případně skupiny znaků) nahradíme tokeny (*tokenizace*), následně makra a aktivní znaky (opakovaně) nahradíme jejich definicemi (*expanze*).

Díl 2: Kreslení

Jak už jste si možná všimli, TEX je v tom, co zatím umíme, relativně nepraktický (a možná jste již narazili na to, že je i pomalý).[7] Částečně je to způsobeno tím, že je (na program) velmi starý, ještě připravený na práci s počítači s minimální pamětí a se systémy s řádky doplněnými mezerami na fixní délku. Částečně tím, že TEX se snaží být co nejobecnější. Hlavně jsme ho ale zatím nepoužívali k tomu, k čemu byl určen – sázení textu. (I když k tomu se často právě budou hodit věci, které jsme se už naučili.)

Pro sázení textu používáme dva směry, horizontální (řádek, do něj se skládají písmena) a vertikální (odstavce a poté stránky nebo sloupce, složené z jednotlivých řádků). Například pokud máme více sloupců na stránce (např. v novinách), pak máme písmena seskládaná horizontálně do řádků, ty vertikálně do sloupců, pak sloupce zase položíme horizontálně vedle sebe a nakonec je toto sloupořadí vertikálně umístěno s nadpisem a patičkou (a možná s dalšími sloupořadími) na jednu stránku novin.

Tomu také odpovídá, že TEX má (dva) horizontální módy, kdy skládá prvky vedle sebe (vytváří tzv. horizontální seznam), a (dva) vertikální módy, kdy skládá věci pod sebe (vytváří tzv. vertikální seznam).[8] Některé prvky se mohou vyskytovat pouze v některých módech, nebo se mohou chovat jinak, podle toho, v jakém módu se TEX nachází.

---

[5] Důležité řídicí sekvence, které ještě nezazněly, jsou `\expandafter` a `\noexpand`, které upravují, jak se expandují (konkrétně neexpandují) makra. Naučit se je používat však vyžaduje obrovské množství příkladů a praxe, na které zde není místo.

[6] Ve skutečnosti ještě máme nultou fázi, a to získání vstupu (*input*), kdy načteme řádek, místo jeho konce vložíme svůj znak (`\endlinechar`) a zahodíme všechny mezery před koncem.

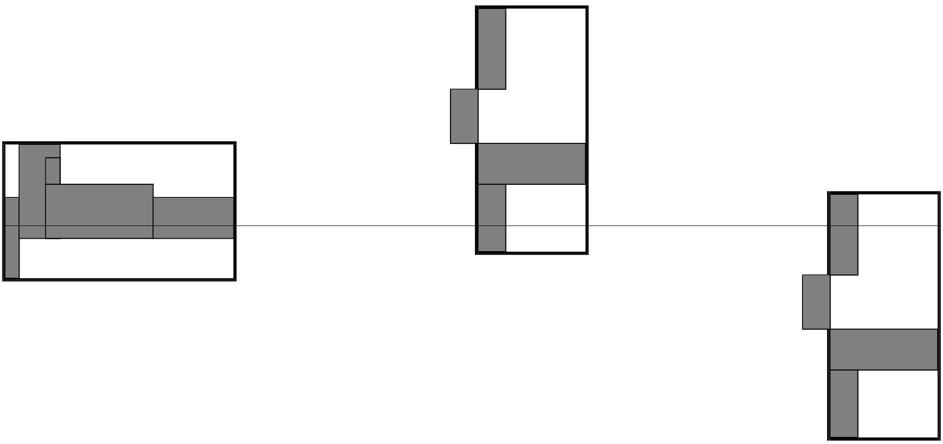
[7] Zlí jazykové tvrdí, že je nepraktický na úplně všechno a že by sami zvládli naprogramovat praktičtější sázecí program. No, zatím se jim to nejspíše nepovedlo.

[8] Ještě má matematické módy (konkrétně vnitřní a display), ale ty jsou kapitola sama pro sebe a matematiku tu budeme probírat až tak, jak ji dělá LATEX.

Vnitřní módy aka boxy

Jednodušší jsou vnitřní módy (vnitřní horizontální a vnitřní vertikální). Odpovídají totiž tzv. boxům, které si můžeme představit jako pomyslné obdélníky usazené na účaří (linku) s šířkou, výškou (nad linkou) a hloubkou (pod linkou), do kterých je obsah naskládán vedle/pod sebe tak, aby „se držel jedné přímky". Vnitřní horizontální mód je uvnitř `\hbox{...}`, skládá prvky na účaří. Vnitřní vertikální mód je uvnitř `\vbox{...}` nebo `\vtop{...}`, zarovnává prvky doleva (jejich levý okraj je na levém okraji vertikálního boxu). Rozměry výsledných boxů se počítají:

- `\hbox`:
  - výška (hloubka): maximum z výšek (z hloubek) jednotlivých prvků (a nuly, tj. je vždy kladná)
  - šířka: součet šířek jednotlivých prvků (může být i záporná, pokud seznam obsahuje např. zápornou mezeru)
- `\vbox` a `\vtop`:
  - šířka: maximum z šířek jednotlivých prvků (a nuly, tj. je vždy kladná)
  - výška a hloubka: u `\vbox` je hloubka rovna hloubce „posledního prvku" (konkrétně posledního boxu, čáry nebo výplně, o výplních později, ale pokud je to výplň, nebo není obsaženo ani jedno, pak je hloubka rovna 0) a výška rovna součtu výšek a hloubek prvků minus tato hloubka; u `\vtop` je to opačně, tj. výška je rovna výšce „prvního prvku" (pokud je to výplň, tak je zase nulová) a hloubka je součet hloubek a výšek prvků minus tato výška.



Obrázek 1: Na účaří usazený `\hbox` obsahující (horizontální) materiál o rozměrech (šířka|výška|hloubka): (5|10|**20**), (15|**30**|5), (-*5*|25|5), (40|15|5), (30|10|5); dále `\vbox` a `\vtop` s materiálem o rozměrech: (**10**|10|20), (-*10*|10|10), (**40**|10|5), (10|15|**10**).

Úloha 2.1 [1b]: *Proč je „výška" rozdělena na výšku a hloubku, kdežto u šířky se nic takového neděje? (Připomínám, že hlavním účelem TEXu je sázet dokumenty.)*

Před `\hbox`, `\vbox` nebo `\vtop` (dále `\*box`) můžeme napsat `\lower <dimen>`, `\raise <dimen>`, `\moveleft <dimen>` nebo `\moveright <dimen>`, kde `<dimen>` je (desetinné) číslo a jednotka, a tím box posunout daným směrem. Také můžeme mezi `\*box` a složenou závorku ještě vložit `to <dimen>` nebo `spread <dimen>`, čímž říkáte, aby se TEX ten rozměr, ve kterém se rozměry sčítají, pokusil roztáhnout na (u `to`) nebo o (u `spread`) danou délku. Pokud se to nepovede (třeba proto, že žádný prvek nelze roztáhnout), tak TEX při kompilaci vypíše varování `under/overfull \*box` a v případě, že jste chtěli `\hbox` menší než lze, pak se vloží černý obdélníček (tzv. slimák).

Například `\hbox to 1pt{x}y` dává `Overfull \hbox (4.2778pt too wide)`, protože se box nezvládl smrštit z `5.2778pt` na `1pt`. Nicméně box má šířku 1 pt, což si můžete ověřit na tom, že se x a y překrývají. Naopak `\hbox spread 1pt{x}y` vypíše `Underfull \hbox (badness 10000)`, neboť roztažení boxu o 1 pt má badness 10000 (maximální), protože zde není žádná mezera, která se roztáhla. Box samotný však bude o 1 pt větší (y bude o 1 pt dál než v `\hbox{x}y`).[9]

Úloha 2.2 [2b]: *Nalezněte (např. v* TEXbook naruby*) jaké máme jednotky délek v TEXu. Kde a proč byste použili ty absolutní, a kde ty závislé na fontu?*

Vnější a hlavní mód aka lámání

Druhé dva módy jsou obtížnější o to, že v nich probíhá lámací algoritmus. Konkrétně hlavní vertikální mód je ten, v kterém TEX začíná a ve kterém probíhá algoritmus na zalámání stránek. Vertikální seznam, který je v něm vytvořen, je taktéž zarovnán vlevo, ale (způsobem, který je popsán např. v knize *TEXbook naruby*) je rozdělen do jednotlivých `\vbox` odpovídajících stránkám. Podobně odstavcový (horizontální) mód je mód, do kterého se přepne buď pomocí `\indent`/`\noindent`, nebo vždy, když se ve vertikálním módu objeví něco, co ve vertikálním módu nelze zpracovat (např. písmeno), a ve kterém probíhá algoritmus na zalámání řádků. Obdobně i zde je horizontální seznam „TEXovskou magií" rozlámán na jednotlivé řádky (`\hbox`y) tak, aby byly všechny správně široké (to může zase vést na `under/overfull \hbox`) a ty jsou pak vloženy do vertikálního módu, ze kterého se přešlo na tento odstavcový.

Obsah horizontálního/vertikálního seznamu

Co tedy může být v boxech, řádcích „textu" nebo „na stránkách"?

- Písmeno, případně slitek (ligatura): pouze v horizontálních seznamech.
- Box.

[9] Pokud si tento odstavec budete zkoušet, nezapomeňte řádek začít např. `\noindent` a pak až `\hbox`, neboť jinak se `\hbox{x}` interpretuje jako prvek vertikálního seznamu, tedy bude nad `y`.

- Pevná nezlomitelná mezera (i záporná): vytvoří se pomocí `\kern <dimen>`, podle toho, zda je v horizontálním nebo vertikálním seznamu, udělá mezeru tím směrem.

- `\penalty` následované číslem: umožní zlom v daném místě (při zalomení jsou odstraněny všechny následující mezery), číslo udává, jak moc je špatné v daném místě zalomit: nula říká podobně špatné jako v mezeře, -10000 (místo `\penalty-10000` pište zkratku `\break`) zlomí vždy, 10000 (zkratka `\nobreak`) nezlomí nikdy.

- Potenciálně pružná zlomitelná mezera: vytvoří se pomocí tokenu kategorie 10 (pouze v horizontálním módu; to je pružná mezera s rozměry spočítanými z fontu) nebo pomocí `\vskip` (případně `\hskip`) následovaným `<glue>`; umožňuje zlom, pokud ji nepředchází jiná zlomitelná mezera nebo penalta; v případě zlomu zmizí a odstraní se všechny mezery za ní a případně jedna pevná nezlomitelná mezera před ní.

  `<glue>` (česky lepidlo) má tvar `<dimen> plus <dimen*> minus <dimen*>`, kde povinný je pouze první `<dimen>`, zbytek tam být nemusí. Znamená, že mezera má základní rozměr první `<dimen>` a může se roztahovat „o" rozměr následující za `plus` a smršťovat „o" rozměr za `minus`. Přičemž „o" znamená, že mezery v jednom boxu s danou velikostí (nebo na jednom řádku/stránce) se roztahují/smršťují v poměru svých roztažností, a pokud toto roztáhnutí nebo smrštění přesáhne rozměr za `plus` nebo `minus`, roztáhnou/smrští se i více, ale vypíší varování `over/underfull h/vbox`.

  `<dimen*>` může obsahovat i jednotky `fil`, `fill` a `filll`, což jsou tři „různě velká nekonečna". Roztahuje/smršťuje se vždy jen v nejvyšším nekonečnu (tj. pokud máte na stejném řádku mezeru `\hskip 0pt plus 5cm` a mezeru `\hskip 0pt plus 1fil`, roztáhne se pouze druhá mezera) a v takovém případě nemůže nastat `over/underfull h/vbox`.

- `\discretionary{pred}{po}{bez}`: pouze horizontální mód, umožní zlom v daném místě, pokud se nezlomí, vloží `bez`, pokud se zlomí, bude na konci řádku `pred` a na začátku dalšího `po` (TEX ví, kde lámat anglická slova, a tohle do nich vkládá automaticky, LATEX s balíčkem babel to umí i pro jiné jazyky).

- Čára: v horizontálním módu `\vrule`, ve vertikálním módu `\hrule`, za čímž může následovat `height<dimen>`, `depth<dimen>` a `width<dimen>`; vykreslí (plný) obdélník – čáru – s rozměry height, depth a width; s tím, že `\vrule` má implicitně šířku 0,4 pt a implicitní výška a šířka odpovídá výsledné výšce a šířce `\hbox`, ve kterém je; `\hrule` má pak implicitní výšku 0,4 pt, hloubku 0 pt a šířku odpovídající šířce `\vbox`.

- Další věci (jako zápis do souboru, značka, insert atd.).

V horizontálním seznamu taktéž můžou být některé vertikální věci, ty se pak přesunou do vnějšího seznamu. (\vadjust{...} vloží vertikální prvky ... při nejbližší příležitosti do vertikálního seznamu. Tj. například \vadjust{\hbox{...}} můžete použít, pokud chcete vložit daný \hbox pod aktuální řádek místo do něho.)

Úloha 2.3 [2b]: *Vytvořte makro, které vysune text doleva od aktuálního místa. Bude se vám hodit* \hbox *s nastavenou šířkou, a nějaká pružná výplň.*

Dále vycentrujte Hello world *na střed řádku. Začněte* \noindent *a bude se vám hodit pružný výplněk a něco, co se nezahodí (např.* \penalty*,* *box*, nebo* \kern*).*

Úloha 2.4 [2b]: *Co udělá horizontální (vertikální) čára nulové tloušťky s šířkou (hloubkou, výškou) větší než šířka (hloubka, výška) ostatních prvků v daném seznamu?*

*Můžete si nakreslit rámeček kolem boxu pomocí (*TeXbook naruby*, str. 95):*

```
\def\obvod #1{\vbox{\hrule \hbox{\vrule #1\vrule}\hrule}}
\obvod{testovaný text}
```

nebo (pro více boxů vnořených do sebe) použít kód ze strany 87 knihy TeXbook naruby*.*

Tuto úlohu určitě zkuste řešit, toto je velmi často využíváno, když TeXáte v „neLaTeXu“ (a i v LaTeXu se vám může hodit).

Úloha 2.5 [2b]: *Poskládejte boxy jako v motivaci výše: horizontální řádky (obsahující písmenka) ve vertikálních sloupcích, které jsou horizontálně vedle sebe a ještě je nad nimi vertikálně nadpis a pod nimi patička.*

*Pro vykreslení celé sestavy použijte (*TeXbook naruby*, str. 95, modifikováno):*

```
\offinterlineskip \def\,{\kern1pt}
\def\Vbox#1{\vbox{\,\hbox{\,\vrule\vbox{\hrule #1\hrule}\vrule\,}\,}}
\def\Vtop#1{\vbox{\,\hbox{\,\vrule\vtop{\hrule #1\hrule}\vrule\,}\,}}
\def\Hbox#1{\hbox{\,\vbox{\,\hrule\hbox{\strut\vrule #1\vrule}\hrule\,}\,}}
```

a místo \hbox*,* \vbox *nebo* \vtop *tedy pište* \Hbox*,* \Vbox *nebo* \Vtop*.*

Registry

V minulém díle jsme si ukázali registr („proměnnou“) typu \count, tedy celé číslo. V tomto díle jsme si ukázali další věci, které by stály za uložení, a to <dimen>, <glue> a boxy. Také jsme se bavili o tokenech a pro několik funkcionalit TeXu jsou potřeba jejich seznamy.

Tudíž máme registry typu: \count, \dimen, \skip (pro <glue>; jeho matematickou variantu je pak \muskip, kde lze používat pouze jednotky mu), \toks (\toks42={...} přiřadí do tokenového registru 42 tokeny ... bez expanze) a \box.

Registrů každého typu je 256,[10] pomocí `\new...\nazev` (`...` je typ registru) je vám automaticky jeden z nich přiřazen pod `\nazev`. `\nazev` se pak expanduje na hodnotu tohoto registru, `\the\nazev` vypíše obsah jako text[11], `\nazev=...` přiřadí do registru. A to s jednou výjimkou, `\box` se totiž chová divně, tudíž zde musíme použít `\box\nazev` (resp. `\copy\nazev`) pro získání hodnoty (`\box` navíc vymaže registr, `\copy` ne), `\setbox\nazev=\*box{}` pro nastavení hodnoty a `\the` nefunguje.[12]

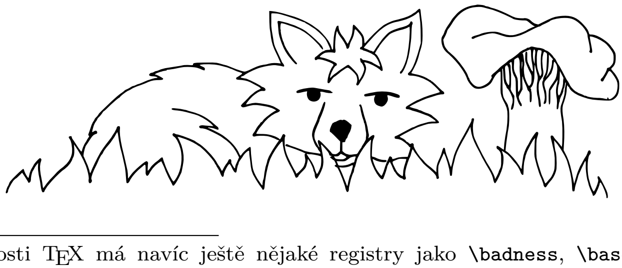
Závěr

Problém 2.6: *Vytvořte něco praktického (ať pro vás, nebo vaše spoluřešitele) v TEXu. Nemusí to být nic složitého, až budeme umět v LATEXu, budou se složité věci dělat o dost jednodušeji.*

Problém 2.7: *Je něco, co byste v TEXu rádi uměli? Třeba vás zajímá, jak funguje stránkový/řádkový lámací algoritmus? Nastudujte si to (ať už na internetu, v knize* TEXbook naruby *nebo konzultací s námi) a popište to čtenářům M&M.*

Například jsme neprobrali ***matematiku****, inserty (plovoucí obrázky), práci se soubory,* `\mark`*,* `\leaders` *nebo to, jak v TEXu fungují různé fonty.*

Pokud jste se dočetli až sem, možná se ptáte, proč bychom měli chtít vytvářet PDF jen s takto základními povely. Odpověď je jednoduchá, neměli! (Já vás varoval.) Ty, kteří mají radši méně složité funkce, do kterých je vidět, odkážu na csplain (konzervativní rozšíření TEXu podporující češtinu, které však neovládám), a případně na heslo „Plain TeX“, pod kterým na internetu najdete opravdu TEX (a ne LATEX). Ty, kterým vyhovují černé krabičky, které většinou udělají tu složitou věc, kterou po nich člověk chce, ale jindy se bohužel zle rozsypou, pozvu k dalšímu dílu v příštím čísle, který už konečně bude o LATEXu (i když stále více o programování než o psaní v něm). Obě možnosti však staví na TEXu, takže na důsledky toho, co jsme se naučili o fungování TEXu, stále budete narážet.



[10] Ve skutečnosti TEX má navíc ještě nějaké registry jako `\badness`, `\baselineskip` nebo `\output`.

[11] U `\dimen` a `\skip` vypíše hodnotu přepočítanou do `\t`, neboť vnitřní reprezentace zapomíná původní jednotku.

[12] Naopak u boxu můžeme použít `\ht\nazev`, `\dp\nazev` a `\wd\nazev` na získání výšky, hloubky a šířky, a můžeme použít `\unhbox\nazev`, `\unhcopy\nazev`, `\unvbox\nazev` a `\unvcopy\nazev` pro získání obsahu boxu (např. pokud potřebujeme vnitřek znovu pružný, nebo ho chceme udělat znovu zalamovatelný).

Řešení 1. dílu

Řešení si také můžete stáhnout v podobě zdrojového kódu na `https://mam.mff.cuni.cz/media/prilohy/31-2-t3-reseni.zip`.

Úloha 1.1

Zadání:
Proč nejde při předefinování makra s použitím původního významu vždy použít `\edef`*?*

Řešení:
Pokud použijeme `\edef\staremakro{\makro}` místo `\let\staremakro\makro`, pak pokud `\makro` mělo parametry, tak to selže rovnou, jelikož při definování `\staremakro` se celý vnitřek složených závorek expanduje a při expanzi `\makro` se k němu nenajdou parametry. To lze samozřejmě obejít tím, že makru `\staremakro` přidáme parametry a předáme je makru `\makro`. To však musíme znát přesné parametry makra `\makro`, což je u spousty maker (např. `\section` v LaTeXu) téměř nemožné. Navíc pokud se někde parametry třeba porovnávaly `\if`em, pak `\edef` způsobí, že se místo parametru porovnají tokeny `#n` (což může skončit například v nekonečné rekurzi).

Pokud makro `\makro` nemělo parametry (a ani se neexpanduje na nic, co by potřebovalo dostat další parametry), tak stále mohlo využívat něčeho proměnlivého, např. čísla zadefinovaného pomocí `\newcount` nebo jiného makra, u kterého se čeká, že ho uživatel předefinuje. Potom `\edef` expanzí obsahu složených závorek navždy do `\staremakro` zadrátoval aktuální hodnoty.

Úloha 1.2

Zadání:
Jak spočítat třetí mocninu čísla? Neboli doplňte program tak, aby vypsal třetí mocninu čísla zadaného na začátku:

```
\newcount\cislo
\cislo=3
...
\the\cislo
\bye
```

Řešení:

```
\newcount\zaklad
\zaklad=\cislo
\multiply \cislo by \zaklad
\multiply \cislo by \zaklad
```

Úloha 1.3

Zadání:
Doplňte podmínky `\ifnum` *spolu s* `\cislojerozumnetrue` *a* `\cislojerozumnefalse` *tak, aby tvrzení na konci kódu bylo správně (pro libovolnou hodnotu* `\cislo`*):*

```
\newcount\cislo
\cislo=42
\newif\ifcislojerozumne
...
Cislo \the\cislo{} \ifcislojerozumne je \else neni \fi vetsi nez
-100 a zaroven mensi rovno 100.
\bye
```

Co je třeba doplnit, pokud bychom chtěli „je/není rovno 42 nebo rovno -42“ místo „je/není větší než -100 a zároveň menší rovno 100“?

Řešení:
Toto je úloha na to, jak udělat logické operace AND (vnoříme podmínky), NOT (použijeme větev `\else`) a OR (dáme podmínky zvlášť). Tedy

```
\ifnum \cislo >-100
 \ifnum \cislo >100 \else \cislojerozumnetrue \fi
\fi
```

v prvním případě, v druhém pak:

```
\ifnum #1 =42 \cislojerozumnetrue \fi
\ifnum #1 =-42 \cislojerozumnetrue \fi
```

Úloha 1.4

Zadání:
Jak zařídit, aby se místo každé mezery nacházející se ve zdrojovém souboru vypsal do PDF znak podtržítko (vypsaný pomocí `\_`)?

Co kdybychom navíc chtěli, aby místo každé mezery bylo nejen podtržítko, ale mezera+podtržítko+mezera, a věděli, že ve zdrojovém souboru není znak `*`?

Řešení:

```
\catcode`\ =13
\def {\_}% Mezera je ted aktivni, takze ji defem priradime vyznam \_
```

Nejjednodušším způsobem jak vyřešit druhou část je uvědomit si, že každému znaku můžeme přiřadit kategorii „mezera“. (Další najdete ve vzorových řešeních v podobě zdrojového kódu: `https://mam.mff.cuni.cz/media/prilohy/31-2-t3-reseni.zip`.)

```
\catcode`\ =13
\catcode`\*=10
\def {*\_*}
```

Úloha 1.5

Zadání:

Řekněme, že velmi jednoduchá tabulka se vytvoří pomocí

```
\halign{\vrule\strut{} # \vrule&&{} # \vrule\cr\noalign{\hrule}
```

následovaného jednotlivými řádky tabulky, kde mezi jednotlivými buňkami v řádku je symbol `&` *a každý řádek končí* `\cr\noalign{\hrule}`*. Vše končí uzavírací závorkou.*

My však odněkud dostáváme tabulku ve formátu CSV (comma-separated values). Tedy (zjednodušeně) každý řádek tabulky je (pouze) na samostatném řádku a buňky jsou od sebe odděleny čárkou. Vzorový soubor ve formátu CSV je k dispozici na našich stránkách[13]*.*

Jak zařídit, abychom nemuseli v tabulce změnit ani znak, a přesto se vypsala podle postupu výše (`\halign...`*)?*

Program může vypadat následovně:

```
...%
\halign{\vrule\strut{} # \vrule&&{} # \vrule\cr\noalign{\hrule}%
\input{tabulka.csv}%
}%
\bye%
```

Řešení:

Jedna z možností je chovat se ke konci řádku jako by to byl libovolný jiný znak. (Vypadá to sice trochu divně, ale hezky to vyjadřuje, jak se TEX staví ke znakům na vstupu.) Pozor, znak `%` musí být přesně na těch řádcích, na kterých je (tam a nikam jinam chceme vložit znak konce řádku).

```
\catcode‘,=4 % zmeni carku na "tabulator"
\catcode‘\
=13 % udela z konce radku aktivni znak
\def
{\cr\noalign{\hrule}} % zadefinuje, co se ma vlozit misto konce radku
```

Jiná řešení této úlohy taktéž najdete v `https://mam.mff.cuni.cz/media/prilohy/31-2-t3-reseni.zip`.

Úloha 1.6

Zadání:

Jak lze naprogramovat cykly (`for` *i* `while`*)?*

Řešení:

Využijeme toho, že v expanzi makra může být ono samo.

[13] `https://mam.mff.cuni.cz/media/prilohy/soustredeni.csv`

```
\newcount\n
\def\zopakujnkrat{
...
\advance \n by -1 % inkrement
\ifnum \n = 0 \else \zopakujnkrat \fi % podminka
}
```

Problém 1.7

Zadání:
Naprogramujte svůj oblíbený prográmek v TEXu.

Řešení:
Jak Doc.MM Michael Jarvis vyřešil načítání matic ze souborů, aby je mohl vynásobit, Doc.MM Ondřej Sedláček počítá rozšířený Eukleidův algoritmus, Bc.MM Svatava Šimečková třídí pole, Mgr.MM Petr Starý vyhledává v poli (které je navíc definované pomocí `\edef`, tudíž není omezen počtem registrů), jak Mgr.MM Anna Trnková hledá prvočísla (pomocí Eratosthenova síta) a jak spolu s Doc.MM Lucií Zůnovou zvládly převádění mezi číselnými soustavami, můžete zjistit v `https://mam.mff.cuni.cz/media/prilohy/31-2-t3-reseni.zip`.

Speciálně doporučujeme řešení Mgr.MM Anny Trnkové, ve kterém definuje docela obecný for cyklus (pro procházení aritmetických posloupností).

Pokud jste něco nepochopili, nesmutněte, TEX je docela složité monstrum, kterému málokdo rozumí dokonale. Nebojte se zeptat na Discordu, e-mailem, konzultovat s kamarády nebo s českou knihou o celém fungování TEXu od Petra Olšáka nesoucí název *TEXbook naruby*[14]. Název knihy je narážkou na *TEXbook*, což je manuál TEXu přímo od jeho autora, profesora Donalda Knutha.

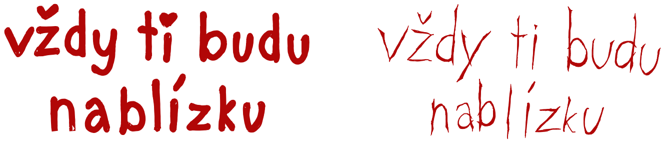
Jidáš; `jonas.havelka@volny.cz`
odevzdávejte do odevzdávátka
PDF i soubory `.tex`

[14] `https://petr.olsak.net/ftp/olsak/tbn/tbn.pdf`

Téma 4 – Typografie a pravopis

V tomto díle se budeme zabývat mikrotypografií, tedy písmem, nebo přesněji fontem. Pokud vás zajímá více pravopis, můžete si přečíst článek Dr.[MM] Dominika Kaňky na straně 27 o tom, jak správně psát číslovky. Případně si můžete počkat do dalšího dílu nebo si přečíst řešení úloh z toho minulého na straně 24.

Díl druhý: Na fontu záleží



K zápisu řeči lidstvo již dlouhou dobu používá různých písem (latinky, cyrilice, run, hieroglyfů atd.), avšak u každého písma existuje nespočet možností, jak ho psát (například u rukou psaného textu existuje psací a tiskací písmo, a i u těch existuje spousta variant). S tímto problémem se musíme v sazbě vyrovnat.

Základní pojem v této problematice je *font* (česky často písmo), což je kompletní sada znaků abecedy[15] jedné velikosti a stejného stylu. *Rodina písma* (anglicky typeface nebo také font family) je pak skupina fontů vycházejících z jednoho stylu, avšak lišících se velikostí a *řezem*, tj. tloušťkou (tloušťka písma je odborně duktus), sklonem (stojaté písmo vs. kurzíva) nebo dalšími variantami (např. kapitálky, monospace atd.), a v některých případech i patkami.

Problém 2.1: *K tvorbě nebo úpravě písma lze využít například program FontForge*[16]*. Můžete zkusit upravit nějaké existující písmo, nebo si nakreslit své vlastní. (Například morseovku, nezapomínejte však, že pokud pošlete dokument v elektronické podobě a nevložili jste font do souboru, pak se příjemci text buď nezobrazí, nebo se použije nějaký standardní font. Stejně tak většinou může člověk font změnit, nebo vykopírovat text jinam, pokud jste nepřevedli text na křivky.)*

Volbou správné rodiny písma můžeme změnit (speciálně zlepšit) dojem, kterým náš dokument působí na čtenáře. Také můžeme zvýšit čitelnost (patkové písmo) a tím udělat náš text atraktivnější. Volbou správného fontu (většinou ze zvolené rodiny) můžeme:

- zvýraznit důležitá slova a pojmy (*kurzíva*);
- oddělit části textu (jako názvy knih nebo kód) od zbytku (*kurzíva*, *monospace*);

[15] Abeceda ≠ všechny znaky (ani všechny znaky např. latinky), tudíž se spoustou fontů může být problém, že nepodporují diakritiku nebo nějakou složitější interpunkci (např. pomlčku).

[16] `https://fontforge.org/`. Pokud vám (stejně jako mně) aktuální verze na Windows nefunguje, odinstalujte ji a nainstalujte starší verzi: `https://github.com/fontforge/fontforge/releases/download/20201107/FontForge-2020-11-07-Windows.exe`

- rozčlenit text (nadpisy jsou *bezpatkovým* větším a často *tučným* písmem, případně *verzálkami* nebo *kapitálkami*, naopak poznámky pod čarou a horní a dolní indexy jsou menším fontem; *tučné* písmo lze také použít pro zvýraznění, kde se v rámci odstavce/kapitoly bavíme o konkrétních pojmech);

- přidat textu určitý význam (citovaný text *kurzívou*, obdobně matematické věty, zadání úloh, . . .).

Problém 2.2: *Operační systémy většinou obsahují mnoho různých rodin písma (které pak můžete vidět např. v Microsoft Wordu). Vyberte si nějaké slavné rodiny (např. Arial, Calibri, Courier, Comic Sans, Computer Modern, DejaVu, Helvetica, Lucida, Papyrus, Times New Roman) a popište, jaký je v nich rozdíl a kdy kterou použít.*

Řezy

Tloušťka (neboli **duktus**) písma odkazuje na tloušťku tahů. Rodina písma obsahuje často mnoho různých tlouštěk od ultra-light po extra-bold, běžně se však používá pouze základní tloušťka (ta však závisí na rodině písma) a trochu tučnější varianta (anglicky bold).

Sklon písma vypovídá o tom, jak je bod, o kterém intuitivně řekneme, že je nad jiným bodem, reálně posunut doleva nebo doprava. V textu psaném zleva doprava jsme zvyklí na sklon směrem vpravo. Běžně se používá kurzíva (také italika, anglicky italic, podle Itálie, odkud pochází), která se však, na rozdíl od skloněného písma (oblique), liší od rovného písma nejen sklonem, ale i vzhledem některých znaků.

Proporciální fonty mají různé znaky různě široké (např. i je úzké a m široké). Naopak **neproporcionální** (také **monospace**) fonty mají všechny znaky stejně široké (což znamená, že n-té písmeno na jednom řádku bude přesně pod n-tým písmenem na předchozím řádku).

Kapitálky (česky také mediuskule, anglicky small caps jako zkrácenina small capitals) pro malá písmena (minusky nebo miniskule) i velká písmena (verzálky nebo majuskule) používají tvar velkých písmen, rozlišují je pouze tím, že malá písmena jsou mírně menší než velká.

Úloha 2.3 [1b]: *Najděte text mimo M&M, kde jsou použity kapitálky.*

Font také může (patkové, serif) a nemusí (bezpatkové, sans-serif) obsahovat **patky**. Patky jsou způsob zakončení tahů, který pomáhá udržovat čtenáře v jedné linii a tím zlepšit čitelnost textu. Při nízkém rozlišení však mohou mít opačný efekt.

Mnoho řezů se dá vytvořit uměle ze základního fontu (např. transformací, která písmo zkosí, lze měnit sklon nebo lze uměle „roztáhnout“ černou plochu a tím dosáhnout tlustějšího písma, případně použít velká písmena jako kapitálky), většinou je však výsledek „divný“ a nepřirozený, tedy je lepší, pokud rodina písem obsahuje tyto řezy ručně připravené.

Technické okénko

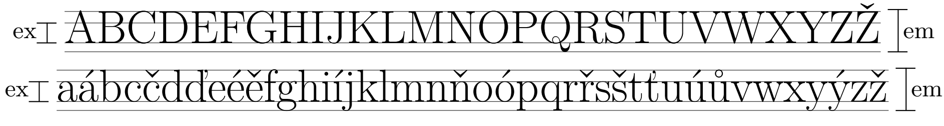
Pro ukládání jednotlivých glyfů (obrázek písmene/znaku) fontu používáme obdobně jako u obrázků buď rastrový nebo vektorový (obrysový) formát, navíc však můžeme použít čárový formát.

- Rastrové fonty glyf popisují diskrétně (nespojitě) tak, že si glyf rozdělí na čtverečky a udají, jakou barvu má (tedy spíše jak moc šedý je) každý čtvereček.
- Vektorové fonty naproti tomu glyf popisují spojitě, většinou za pomocí Beziérových křivek. Tedy o každém bodu můžeme přesně říct, zda leží, nebo neleží v daném glyfu.
- Čárové fonty pouze popisují, jak znak nakreslit pomocí „tahů štětce/pera“. Program vykreslující takový font si pak může trochu vybrat, jak budou jednotlivé tahy vypadat (např. jejich tloušťku).

Problém 2.4: *Zamyslete se nad důvody, proč se každý z těchto formátů používá.*

Font má několik důležitých vodorovných linií, tzv. dotažnic, u kterých končí tahy tvořící jednotlivé glyfy. Nejdůležitější je *základní dotažnice* (účaří), na kterou se glyfy „pokládají“. Směrem dolů je pak *dolní dotažnice*, tedy nejnižší místo, kam se dotahují tahy vedoucí dolů (např. u písmen `jpqy`). Směrem nahoru je *střední dotažnice*, ke které se dotahují tahy malých písmen (vyjma `bdfhlkt`, i když i u těch je střední dotažnice patrná), poté *horní dotažnice*, kam se dotahují písmena `bdfhlkt`, diakritika nad malými písmeny a většinou i velká písmena (občas však mají vlastní dotažnici trochu níže). Nad ní je pak *akcentová dotažnice*, kam se dotahuje diakritika nad velkými písmeny.

Vzdálenost mezi účařím a střední dotažnicí je tzv. *střední výška* a odpovídá jednotce 1 ex (výška `x`). Nejmenší výška řádku, do kterého se glyfy vejdou (často bez diakritiky nad velkými písmeny), se nazývá *kuželka*. Tento rozměr se pak vyskytuje v popisu fontu (např. 12pt font, neboli dvanáctibodový font znamená, že font má kuželku 12 bodů, kde bod je délková jednotka, která záleží na stylu sazby, často okolo 0,35 mm). Z kuželky pak vychází *čtverčík* (anglicky *quad*), historicky čtvercová plocha o straně délky kuželky používaná při tisku, dnes se však používá jako jednotka značená 1 em (em, protože podle jedné z alternativních definic měl být čtverčík plocha písmene `M`).



Obrázek 2: Dotažnice u fontu Latin modern (všimněte si, že 1 em je trochu větší než vzdálenost dolní a horní dotažnice, ale ne dostatečně, aby se tam vešla i diakritika u verzálek)

V předchozích odstavcích jsme mluvili o vertikálním umístění glyfů. Horizontální umístění je však ještě zajímavější. Mohli bychom glyfy pokládat jen tak vedle sebe, avšak některé kombinace písmen by vypadaly, že mezi sebou mají mezeru (např. AV), a některé naopak téměř splývaly (např. gj). Proto často nestačí mít jen pevné mezírky kolem glyfů, což se např. v TeXu řeší tabulkou překládající dvojice glyfů na velikost mezery (třeba i zápornou) mezi nimi. (Viz AV a gj.) Upravování mezer mezi písmeny se říká *kerning* (také vyrovnání či podřezávání).

Kromě mezer mezi znaky máme v textu další mezery, zejména mezislovní a okolo (většinou za) interpunkcí. Ani ty nejsou úplně jednoduché. Například při sázení do bloku je vhodné mezislovní mezery smršťovat a roztahovat. V americké sazbě navíc za koncem věty bývá větší mezera (ale ne každá tečka je konec věty, což nám automatizaci velikosti mezery dost stěžuje). V češtině je naopak zvykem kompenzovat „bílou plochu“ kolem tečky a čárky tím, že je mezera menší. Proto má font v TeXu kromě velikosti ex a em ještě velikost mezislovní mezery, její roztažnost a smrštitelnost a její „zvětšení“.[17]

A to nemluvíme o mezerách v zápise matematiky nebo kolem závorek, které mění velikost podle výšky jejich obsahu…

Problém 2.5: *Kde se ještě (v češtině) zpravidla píše užší mezera?*

Ligatury

Některé dvojice (trojice, čtveřice, …) písmen vedle sebe mohou působit rušivým dojmem. Proto se v profesionální sazbě nahrazují ligaturou (česky slitkem), tedy jedním glyfem, který v sobě nese obě písmena. Nejčastěji uváděnými příklady jsou ﬁ (z písmen fi), ﬂ (fl), ﬀ (ff) a ﬃ (ffi). Nejznámější ligaturou je však znak &, který vznikl jako ligatura písmen e a t (latinské et je spojka „a“).

V jazycích, které používají ae nebo oe jako jednu hlásku (v latině se čtou jako dlouhé é a můžeme se s nimi setkat například na náhrobních kamenech v kostele), narazíme také na ligatury Æ, Œ, æ a œ. Úplně stejným způsobem vzniklo německé ostré s (ß) jako spojení dlouhého s (ſ) a goticky psaného z (ʒ).

TeX (nebo přesněji rodina fontů Computer modern) používá ligaturu k jednomu pohodlnému triku, považuje totiž pomlčku (–) a dlouhou pomlčku (—) jako ligatury spojovníků (-- a ---). Navíc dlouhá pomlčka samotná okolo sebe nemá mezery, tedy ji lze „slít“ do dlouhé čáry: ————————— (čáry je však lepší dělat jiným způsobem). Obdobně ze dvou uvozovek (‘‘) vytvoří pravou dvojitou anglickou uvozovku ” (nepoužívat v češtině[18]) a ze dvou menšítek nebo většítek (<< nebo >>) vytvoří boční (francouzské) uvozovky « nebo ».

[17] Tyto rozměry v TeXu zjistíte pomocí `\fontdimen*\font`, kde * = 2 dá velikost mezery, 3 a 4 dá její roztažnost a smrštitelnost, 5 a 6 dá ex a em, 7 dodatečnou mezeru a 1 sklon písma, např. kvůli umístění diakritiky.

[18] Čeština primárně používá uvozovky „. . . “ psané příkazem `\uv{...}` nebo `\clqq` a `\crqq`. Dále jsou v některých případech (např. přímá řeč v přímé řeči) přípustné jednoduché uvozovky ‚. . . ‘ a boční uvozovky ». . . « (či dokonce jednoduché boční uvozovky ›. . . ‹). Použití jiných uvozovek je chyba!

Do extrému pak dotáhla ligatury rodina fontů Fira Code.[19] Ta z všemožných operátorů, které se v programování píší více znaky, udělá ligatury, které vypadají jako daný operátor zapsaný „jedním znakem“.

Problém 2.6: *Znáte nějaké jiné zajímavé ligatury? Podělte se s námi!*

Inspirováno prezentací `https://www.gymkl.cz/e_download.php?file=data/editor/308cs_2.pdf&original=Typograficke%CC%81%20minimum.pdf`.

Jidáš; `jonas.havelka@volny.cz`
odevzdávejte do odevzdávátka
PDF i zdrojové soubory

Řešení 1. dílu

Problém 1.1

Zadání:
Přesvědčte nás o tom, že právě váš oblíbený textový procesor / sázecí program je ten nejlepší.

Řešení:
K tomuto problému nám přišlo několik řešení popisujících TEX, Google dokumenty a Microsoft Word. Bohužel většina nás moc nepřesvědčila, neboť popisovala, co daný textový procesor / sázecí program umí, a úplně ignorovala fakt, že to umí i ostatní.

Nicméně vychvalovaná vlastnost Wordu je jeho intuitivní používání, Google dokumentů jejich možnost spolupráce a TEXu jeho základní typografické nastavení a přesnost.

Problém 1.2

Zadání:
Na `https://mam.mff.cuni.cz/media/prilohy/31-4.html` *máte nepoužitý draft článku do M&M. Upravte ho tak, aby šel co nejlépe číst.*

Bonus: Po druhém deadlinu prvního čísla budeme (anonymně, proto prosím nesdílejte svá řešení s ostatními) na Discordu hlasovat o nejlépe čitelných řešeních.

Řešení:
Snad všichni z vás našli v rámci úlohy 1.5 v tématku poučku, že text zarovnáváme do bloku. Bohužel více než polovina došlých řešení do bloku zarovnána nebyla. Další častou chybou bylo nechání neslabičných předložek na konci řádku.

Samostatnou kapitolou jsou pak mezery. Za čárkou (pokud se nejedná o desetinnou čárku) se vždy píše mezera. Tedy například správně je `A, B` místo `A,B`. Naopak před čárku nebo před ukončovací závorku se mezera nepíše, takže v souboru byl jeden řádek, kde bylo třeba takové mezery odstranit.

[19] `https://github.com/tonsky/FiraCode`

Brzy po vydání tohoto čísla obdrží všichni řešitelé této úlohy naše podrobné komentáře jejich hodnocení, a jakmile doladíme pravidla hlasování, budete moci rozhodnout o nejlepším řešení i vy. Připojte se k našemu Discordu[20] a očekávejte brzký start hlasování.

Problém 1.3

Zadání:
Najděte ve svých učebnicích češtiny pravopisné (a případně typografické) chyby a podělte se o ně s námi. Dobrý odrazový můstek jsou pomlčka–spojovník, lomítka, uvozovky atd.

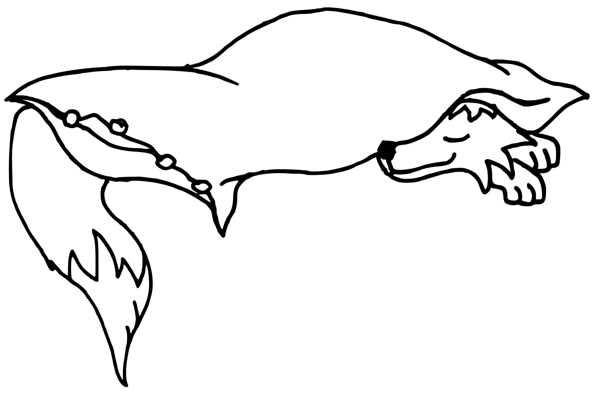
Lehčí varianta: v učebnicích čehokoliv.

Námi nejoblíbenější nález byl:

Řešení od Petry Bělušové a Bc.^MM Svatavy Šimečkové:

> Poslední věc, na kterou bychom rády upozornily, jsme nenašly v učebnici, nýbrž v samotném časopise M&M. Při úpravě textu je vhodné dbát na to, aby odstavec nebyl mezi stránky rozdělen tak, že na některé je pouze jeden řádek (`cs.wikipedia.org/wiki/Vdovy_a_sirotci`). V prvním čísle, na hranici stran 6 a 7, se nachází vdova – poslední řádek 6. stránky je prvním řádkem odstavce pokračujícího na straně 7.

Mají samozřejmě pravdu. A přestože v některých případech není zbytí (a to očividně i když TEX přesvědčíte, aby vdovy a sirotky dělal dvojnásob nerad), v tomto případě bylo nejspíše vhodné upravit velikosti obrázků, aby se toto nestalo.



[20] `https://discord.com/invite/MxPjKsDFWG`

Problém 1.4

Zadání:
Pravopisné a typografické vtipy: Nalezněte textíky, které se vtipným způsobem snaží upozornit na pravopisné jevy (např. čárky) nebo si dělají srandu z typografické úpravy nějakého textu. Nezapomeňte nám poslat jejich zdroj.

Řešení:
Stejně jako jsme se my moc pobavili, můžete se pobavit sami kompilací vašich řešení na `https://mam.matfyz.cz/media/prilohy/31-typovtipy.pdf`.

Úloha 1.5

Zadání:
Už v tomto díle byla spousta typografických (a pravopisných) pouček. Naleznete je všechny?

Řešení:
Typografické poučky, které zazněly:

- Na řádku by nemělo být víc než 80 znaků.
- Delší text by měl být zarovnaný do bloku a patkovým písmem. Naopak kde to má význam (mnoho krátkých sdělení), použijte odrážky.
- Vizuálně oddělujte nadpisy (např. bezpatkovým tučným a větším fontem).
- Vyznačit důležitá slova (ale pamatujte na to, že tučné písmo a barvy spíše upoutají oči, takže je použijte spíše pro členění textu a na důležitá slova použijte kurzívu nebo monospace).

Další poučky (které nejsou moc typografické/pravopisné jako spíš obecné):

- Použijte spell check a zkontrolujte si gramatiku a pravopis.
- Používejte formát PDF (nejvíce zaručuje zachování formátování).
- Zkontrolujte si, zda vaše práce nemusí splňovat typografickou normu (a případně ji dodržujte).

Problém 1.6

Zadání:
Máte nějaký oblíbený pravopisný / typografický jev? Napište nám o něm článek!

Řešení:
Bc.$^{\text{MM}}$ Stela Ožanová zjistila, že když zadáte do Google vyhledávání Comic Sans, Times New Roman nebo Garamond font, dostanete výsledky vyhledávání v daném fontu. To funguje i s jinými fonty (viz `https://en.wikipedia.org/wiki/List_of_Google_Easter_eggs`) a taktéž s pojmy „kerning“ a „keming“.

Dr.$^{\text{MM}}$ Dominik Kaňka nám zaslal hezký článek o tom, jak psát číslovky:

Psaní číslovek

5b

Dr.[MM] Dominik Kaňka

Jednou z částí našeho pravopisu, kterou lidé často nepíší správně, je bezpochyby psaní číslovek. Přitom to není nikterak složité a pravidla v tomto odvětví jsou jasně definovaná. (A navíc mi připadalo, že se to sem docela hodí.)

Jako první bych rád zmínil skloňování číslovek dva a oba. V prvních pádech všech rodů si obě zachovávají jejich koncovky (mužský – dva, ženský – dvě, střední – dvě; obdobně oba, obě, obě). Ve 2. a 6. pádě není nikde žádné -ch, takže máme dvou (nikoli dvouch) a obou (nikoli obouch). Ve 3. a 7. pádě jsou přípustné pouze varianty dvěma a oběma. U složených slov můžeme psát dvou- i dvoj- (např. dvoubarevný i dvojbarevný).

Číslovky tři a čtyři se skloňují dle vzoru kost se dvěma výjimkami. Číslovka tři má v 7. pádě tvar tř**e**mi, číslovka čtyři má ve 2. pádě tvar čtyř. U obou číslovek jsou možné ve 2. pádě tvary třech, respektive čtyřech. Tyto tvary jsou ovšem považovány za hovorové (tzn. spisovné, ale vhodné spíše do mluveného projevu). Pokud se tyto číslovky vztahují k částem těla, mají koncovku duálovou (např. mezi čtyřma očima). U složených slov jsou opět dvě možnosti, troj- i tří- (např. trojbarevný i tříbarevný).

	rod mužský	**rod ženský**	**rod střední**
1. pád	dva, oba	dvě, obě	dvě, obě
2. pád	dvou, obou	dvou, obou	dvou, obou
3. pád	dvěma, oběma	dvěma, oběma	dvěma, oběma
4. pád	dva, oba	dvě, obě	dvě, obě
6. pád	dvou, obou	dvou, obou	dvou, obou
7. pád	dvě**ma**, obě**ma**	dvě**ma**, obě**ma**	dvě**ma**, obě**ma**

	tři	**čtyři**
1. pád	tři	čtyři
2. pád	tří, třech	čtyř, čtyřech
3. pád	třem	čtyřem
4. pád	tři	čtyři
6. pád	třech	čtyřech
7. pád	tře**mi**	čtyř**mi**

Tabulka 1: Tabulky shrnující skloňování číslovek dva, oba, tři, čtyři.

Psaní řadových číslovek je ještě jednodušší. Stačí, pokud chceme psát číslem, napsat číslo následované tečkou (např. 5. – „pátý“). Je možné také zapsat slovem (např. pátý). Jakékoli zápisy typu 5ý, 5tý, 5-tý jsou špatně.

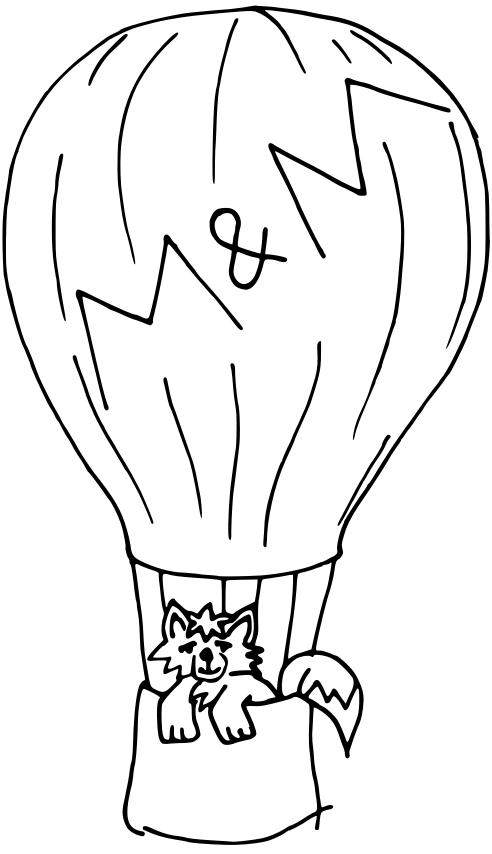
Psaní mezer také mění význam číslovek. Je rozdíl, pokud napíšu 8kg a 8 kg. Pokud mezeru nenapíšu, jedná se o jedno slovo (osmikilogramový), pokud mezeru napíšu, mám dvě slova (osm kilogramů). Složená slova můžu napsat i pomocí číslice a dalšího slova (8kilogramový, 4metrový atd.).

Vyjádření číslovky slovem má také vícero možností. Například 475 můžeme zapsat jako čtyři sta sedmdesát pět nebo čtyři sta pětasedmdesát (pokud chci napsat desítky a jednotky v opačném pořadí, musím je napsat jako jedno slovo, jinak se všechny části píší zvlášť). V psaných textech můžeme tyto číslovky skloňovat dvěma způsoby. Prvním je skloňování každého členu (např. bez čtyř set sedmdesáti pěti, respektive pětasedmdesáti). Ve druhém případě můžeme skloňovat pouze desítky a jednotky, zbytek výrazu skloňovat nemusíme (např. bez čtyři sta sedmdesáti pěti, respektive pětasedmdesáti).

Doufám, že se mi povedlo ukázat, že psaní číslovek opravdu není složité a pravidel není mnoho.

Zdroje:

- https://didacticus.cz/psani-cislovek
- https://prirucka.ujc.cas.cz/?id=785&dotaz=%C4%8D%C3%ADslovky
- https://prirucka.ujc.cas.cz/?id=790&dotaz=%C4%8D%C3%ADslovky
- https://prirucka.ujc.cas.cz/?id=791&dotaz=%C4%8D%C3%ADslovky
- https://prirucka.ujc.cas.cz/?id=670&dotaz=%C4%8D%C3%ADslovky
- https://prirucka.ujc.cas.cz/?id=671&dotaz=%C4%8D%C3%ADslovky
- https://www.umimecesky.cz/cislovky-sklonovani



Téma 5 – Strojíme matiku puntíky a šipkami

Úvod

Mám rád obrázky. Máš rád obrázky. Má rád obrázky. Máme rádi obrázky. Máte rádi obrázky. Mají rádi obrázky.

Teď vážně, kdo nemá rád obrázky. Věřili byste nám, drazí čtenáři, tvrdili-li bychom vám, že celou matematiku umíme překreslit do obrázků? Ovšem, ta sama o sobě není nikterak odvážná výpověď; vždyť písmena, čísla i vše ostatní na papíře jsou vlastně, obrázky. My zde nechováme na mysli ledajaké obrázky, nýbrž obrázky vzniknuvší kresbou pouze dvou věcí – puntíků ($\bullet$) a šipek ($\longrightarrow$) mezi puntíky. Co však míníme slovy „překreslit celou matematiku“, zůstane ještě nějakou chvíli oděno v rouchu tajemství.

V tomto čísle se rozhovoříme o tak zvané *teorii kategorií* – jednom z nejoblíbenějších způsobů, kterak kreslení pramnoha matematických struktur vnuknout formální ráz. Místo i vůle stačiti budou na styk se základními idejemi; jejich účel a užitečnost odkryjeme s časem.

Teorie kategorií

Teorie kategorií je matematická disciplína, jež postupně vznikala v průběhu 20. století. Za její debut v širém matematickém světe lze považit knihu *Categories for the Working Mathematician*[21] od Saunderse Mac Lana vydanou roku 1971. Dnes se vskutku štědře využívá především v oblasti algebry.

Jsouc někdy přezdívána *metamatematikou*, teorie kategorií zkoumá vpravdě matematiku jako takovou. Je vlastně jakýmsi pohledem shůry na to rašeliniště struktur a konceptů, které vypadají, že mají mnoho společného, ale i mnoho rozdílného. Teorie kategorií zkoumá právě ty vlastnosti, které spolu všechny sdílejí.

Tímto tématkem se s ní velmi pravděpodobně setkáte poprvé. Setkání bude to zprvu neformální, intuitivní a obrázkové; ne však na úkor opominutí mnohých rigorózně kutých myšlenek. Na řadu technických podrobností leč schází času i smyslu. Ba, před samotným seznámením se stroze rozhovoříme o jednom zásadním rozumovém kroku v pojetí matematiky (a částečně světa), jejž radíme pro zdárné pochopení učinit.

V teorii kategorií nebývá zvykem definovat věci určením toho, „co jsou“, nýbrž výčtem „vlastností, které splňují“ či „informací, které nesou“. Místo aby řekli: „Toto je hrnek,“ kategorici dějí spíše: „Toto je věc, která slouží k uskladnění tekutiny během pití.“ A … pak dalších mnoho stran dokazují, že všechny takové věci jsou sobě podobné. Tento velmi svobodný přístup má podíl na úspěchu, který teorie kategorií zaznamenala. Totiž, logika i teorie množin (nikoli explicitně, ale svým výkladem) v sobě již přirozeně skýtají jistou *interpretaci* – logická spojka $\wedge$ znamená „a zároveň“, množina je „souhrn prvků, jež spolu souvisejí“ a podobně.

[21] V době vzniku této knihy byla teorie kategorií považována za abstraktní „hrátky“ matematiků, již měli dostatek majetku, aby nemuseli pracovat. Užitečnost této teorie i v principu praktičtějším odvětvím ukázal právě S. Mac Lane – odtud přívlastek *working* v názvu.

Interpretace objektu podvědomě zužujíc náš obzor nás v zásadě vede s ním pracovat konkrétním způsobem. Proto vám teď neprozradíme, co kategorie *je* (neb to nevíme), ale jenom a pouze, *co ji tvoří.* Následující „definice“ je šita tak, aby v sobě shrnovala jen ty zcela nejvšednější vlastnosti všemožných matematických struktur (možná jste slyšeli o *grupách*, *okruzích* či *vektorových prostorech*[22]). To není bez jejich hlubší znalosti snadné prohlédnout, prosíme pročež shovívavé čtenáře o důvěru.

Jakoby definice kategorie

Nyní, **kategorie** $\mathcal{A}$ je dána následujícími daty:

- **puntíky** (formálně **objekty**), které budeme značit symbolem $\bullet$ či malými písmeny latinské abecedy (a, b, c, $\ldots$), bude-li třeba formálního zápisu;
- **šipkami** (formálně **morfismy**), které vždy vedou z puntíku do puntíku. Ty budeme vždy značit piktogramem $\longrightarrow$, popřípadě navíc malými písmeny řecké abecedy (α, β, γ, $\ldots$). Formální zápis faktu, že šipka α vede z puntíku a do puntíku b vypadá takto:

$$a \xrightarrow{\alpha} b \quad \text{nebo} \quad \alpha : a \to b.$$

  Všechny šipky z a do b v kategorii $\mathcal{A}$ značíme $\mathcal{A}(a,b)$.

Před dokončením definice kategorie se, snad příliš široce, vyjádříme ke značení (ovšem z filosofických důvodů). Fakt, že objekt a patří do kategorie $\mathcal{A}$, se obvykle píše stylem $a \in \mathcal{A}$. To je ovšem pouze drzé zneužití značení z teorie množin! *V žádném smyslu* neznamená, že a je *prvkem* kategorie $\mathcal{A}$, protože $\mathcal{A}$ rozhodně nemusí být množina. Ježto si silně nepřejeme, abyste, nebozí čtenáři, přemýšleli o kategoriích jako o *množinách* a o objektech jako o *prvcích*, tohoto tradičního značení se s neoprávněnou hrdostí buřičů zdržíme a budeme psát například $a \blacktriangleleft \mathcal{A}$, abychom vyjádřili, že a je objekt kategorie $\mathcal{A}$. Ještě jednou a znovu – to neznamená, že *a patří do* či *leží v* $\mathcal{A}$. Onen vztah pouze vyjadřuje, že objekt a je součástí dat popisujících kategorii $\mathcal{A}$. Podobně, skutek, že α je šipka z a do b v kategorii $\mathcal{A}$, budeme někdy psát jako $\alpha \blacktriangleleft \mathcal{A}(a,b)$.

Předchozí odstavec částečně vysvětluje původ názvu *kategorie.* Objekty kategorie $\mathcal{A}$ (ve smyslu *čí* objekty) jsou totiž přesně … objekty kategorie $\mathcal{A}$ (ve smyslu *jaké* objekty). Podobně jako jsou třeba *okurky* objekty kategorie $\mathcal{ZELENINA}$ nebo *kočár* objekt kategorie $\mathcal{VOZIDLO}$. Shrnuto, značení $a \blacktriangleleft \mathcal{A}$ znamená, že objekt a má vlastnosti, které ho zařazují do kategorie $\mathcal{A}$. Na druhou stranu však jsou tyto vlastnosti, podle nichž je do $\mathcal{A}$ zařazen, určeny právě všemi objekty, které je mají. K uchopení významu předchozí věty kéž poslouží následující paradigma: jakési konkrétní vozidlo (řekněmež automobil) je vozidlem pro to, že sdílí jisté

[22] Matematické články anglické wikipedie jsou obvykle velmi dobře psané a srozumitelné. Ambiciózní zájemci buďtež zváni sobě přečíst třeba článek o grupách na `en.wikipedia.org/wiki/Group_(mathematics)` a hledat podobnosti se záhy představenými kategoriemi.

vlastnosti s ostatními vozidly – slouží k převážení, má kola apod. Ovšem, fakt, že jsou tyto vlastnosti typické pro vozidla, víme jenom tak, že je všechna vozidla mají. Neexistuje žádná „vnější“ podmínka, žádný archetyp, dle nějž soudit, je-li, či není-li, objekt vozidlem. Ptáte-li se nás, je to takový filosofický uzel, který ale umožňuje vnímat kategorie pokaždé tak, jak v danou chvíli nejvíce záhodno.

Zpět k definici. Požadujeme, aby šipky měly další omezující vlastnosti, které je připodobňují *zobrazením* či *funkcím* mezi objekty. Konkrétně,

(SK) šipky lze **skládat**. To znamená, že kdykoli v kategorii $\mathcal{A}$ nastává situace

$$\begin{tikzcd} & \bullet \arrow[dr] & \\ \bullet \arrow[ur] & & \bullet, \end{tikzcd}$$

pak existuje šipka, která tento trojúhelník doplňuje. Schematicky, (a toto značení budeme používat i nadále)

$$\begin{tikzcd} & \bullet \arrow[dr] & \\ \bullet \arrow[ur] \arrow[rr, dashed, "\exists"] & & \bullet. \end{tikzcd}$$

Formálně píšeme, že pro libovolné puntíky a, b, c a šipky $a \xrightarrow{\alpha} b, b \xrightarrow{\beta} c$, existuje šipka $a \xrightarrow{\alpha\beta} c$, které se říká jejich **složení**. Posloupnosti navazujících šipek někdy přezdíme **cesta** v kategorii $\mathcal{A}$. Přirozeně, složení šipek existuje jedině, když první šipka končí tam, kde druhá začíná;

(AS) toto skládání je **asociativní**. Nezáleží na tom, jestli prve složíme α s β a po $\alpha\beta$ pokračujeme šipkou γ, nebo nejdříve vyrobíme $\beta\gamma$ a před touto jdeme šipkou α. V obou případech je výsledkem stejná šipka v kategorii $\mathcal{A}$. To je, soudíme, velmi přirozená podmínka. Formálně zapsáno, $(\alpha\beta)\gamma = \alpha(\beta\gamma)$. Obě cesty vizte nakreslené níže.

$$\begin{tikzcd} & \bullet \arrow[r, "\beta"] \arrow[drr, "\beta\gamma"] & \bullet \arrow[dr, "\gamma"] & \\ \bullet \arrow[ur, "\alpha"] \arrow[rrr, dashed, "\alpha(\beta\gamma)"'] & & & \bullet \end{tikzcd} \qquad \begin{tikzcd} & \bullet \arrow[r, "\beta"] & \bullet \arrow[dr, "\gamma"] & \\ \bullet \arrow[ur, "\alpha"] \arrow[urr, "\alpha\beta"] \arrow[rrr, dashed, "(\alpha\beta)\gamma"'] & & & \bullet \end{tikzcd}$$

Onu přirozenost podmínky asociativity lze nastínit třeba pohybem mezi městy. Jednoho chmurného jesenního večera cestujete z Ostravy do Brna, projíždějíce obcemi Bílovec a Vyškov. Troufáme si tvrdit, že pod výpověďmi „Jedu z Ostravy do Bílovce a pak přes Vyškov do Brna“ a „Jedu z Ostravy přes Bílovec do Vyškova a pak z Vyškova do Brna“ sobě všední posluchač představí touž trasu.

(SM) každý puntík má kolem sebe jednu speciální **smyčku** – šipku, jež v něm začíná i končí. Speciální v tom smyslu, že její složení s libovolnou jinou šipkou nic nedělá. Je to vlastně taková „zůstaň na místě“ šipka. Formálně, ke každému puntíku a existuje šipka $1_a : a \to a$ taková, že kdykoli vezmeme šipku α, která začíná v a, pak $1_a\alpha = \alpha$, a kdykoli vezmeme šipku β, která končí v a, pak $\beta 1_a = \beta$. Její „oficiální“ název zní *identita* na a.

$$\begin{tikzcd} \bullet \arrow[loop left, "1_a"] \arrow[r, "\alpha"] & \bullet \end{tikzcd} \qquad \begin{tikzcd} \bullet \arrow[r, "\beta"] & \bullet \arrow[loop right, "1_a"] \end{tikzcd}$$

Vlastnost identity říká, že v levém obrázku jsou *úplně všechny* cesty z levého puntíku do pravého ve skutečnosti stejné. Všimněme si, že takových cest je nekonečně mnoho, protože po smyčce 1_a mohu jít kolikrát chci a teprve potom pokračovat šipkou α. Všechny cesty v pravém obrázku jsou taktéž stejné.

Pro úsporu místa i přehlednost obrázků budeme místo

$$\begin{tikzcd} \bullet \arrow[loop left] \end{tikzcd} \text{ kreslit pouhé } \odot.$$

Kolečko kolem puntíku bude vždy symbolizovat *výhradně* identitu na tomto puntíku. Případné ostatní šipky vedoucí z puntíku do něj samého vykreslíme řádně.

Z právě uvedených pravidel, jež šipky musejí splňovat, plyne úvahou i mnoho jejich dalších vlastností. Kupříkladu, ačkoliv jsme to přímo neuvedli, identita na puntíku existuje vždy jen jedna. Důkaz se pokuste sestrojit v první úloze témátka.

Úloha 2.1 [0,5b]: *Dokažte, že pro každý puntík* $a \blacktriangleleft \mathcal{A}$ *existuje* pouze jediná *šipka* 1_a *s vlastností identity.*

Doporučujeme přeformulovat výrok takto: „Ať je $\alpha : a \to a$ *smyčka kolem puntíku* $a \blacktriangleleft \mathcal{A}$ *majíc vlastnost identity. Pak* $\alpha = 1_a$*.“*

Kategorie s rozumným počtem objektů můžeme celé zakreslit jedním obrázkem; obrázkem obsahujícím všechny jejich puntíky i šipky. Důsledkem právě zakončené definice není ale *každý* obrázek z puntíků a šipek kategorií. Třebaže puntíky na papíře jsou volné jak ptáci v oblacích, šipky jsou svázány podmínkami vypsanými v definici kategorie. Například

$$\bullet \longrightarrow \bullet$$

celou kategorií *není*, protože levý ani pravý puntík nemají kolem sebe smyčku.

Naopak,

$$\odot \qquad \odot$$

celou kategorií dozajista je. Totiž, nic v definici kategorie nevynucuje existenci šipek mezi *různými* puntíky. Jediné šipky, jež v každé kategorii existovat musejí, jsou identity na puntících.

Jmeme se představit další příklady a vysvětlit, jak visuální reprezentace kategorií implicitně vynucují některé vlastnosti šipek.

Příklady kategorií

Pohlédněme na příklad složitější kategorie o čtyřech objektech:

$$\begin{tikzcd} \odot \arrow[r, "\alpha"] \arrow[d, "\beta"'] \arrow[dr, "\varepsilon"] & \odot \arrow[d, "\delta"] \\ \odot \arrow[r, "\gamma"'] & \odot \end{tikzcd}$$

Po krátkém zamyšlení si možná říkáte: „Tohle přece celá kategorie není. Já musím umět šipky skládat, a nikde nevidím ani $\beta\gamma$, ani $\alpha\delta$.“ Opravdu, mezi levým horním a pravým dolním puntíkem musejí vést ještě další dvě šipky. Takový obrázek představuje celou kategorii jedině v případě, kdy jsou všechny tři šipky vskutku jedna a ta samá, symbolicky $\varepsilon = \alpha\delta = \beta\gamma$. Ta není situace vůbec neobvyklá. Snad uveďme konkrétní příklad. Řekněme, že všechny čtyři objekty jsou prostě reálná čísla – pozor, nikoli nějaká konkrétní čísla, ale celá *množina* reálných čísel. Za šipky volmež pro jednoduchost nějaké lineární funkce $\mathbb{R} \to \mathbb{R}$. Nesmíme ale vybírat úplně libovolně; složení $\alpha\delta$ (nebo $\delta \circ \alpha$, jak se složení funkcí obyčejně zapisuje) musí být totožné s $\beta\gamma$. Jedna taková správná volba je třeba

$$\begin{aligned} \alpha(x) &= 2x + 3, & \beta(x) &= x - 1, \\ \gamma(x) &= -2x - 5, & \delta(x) &= -x. \end{aligned}$$

Při té platí

$$\varepsilon(x) = \alpha\delta(x) = \delta(\alpha(x)) = -(2x + 3) = -2x - 3 = \gamma(\beta(x)) = \beta\gamma(x).$$

Tuto konkrétní kategorii můžeme nakreslit třeba tak, že místo puntíků použijeme symbol $\mathbb{R}$ a šipky popíšeme výstupy příslušných funkcí. Učinili jsme tak obrázkem 5.

$$\begin{tikzcd}
\mathbb{R} \arrow[r, "2x+3"] \arrow[d, "x-1"'] \arrow[dr, "-2x-3"] & \mathbb{R} \arrow[d, "-x"] \\
\mathbb{R} \arrow[r, "-2x-5"'] & \mathbb{R}
\end{tikzcd}$$

Obrázek 5: Příklad konkrétní kategorie.

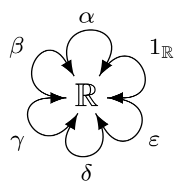
Úloha 2.2 [0,5b]: *Jaký má v kategorii na obrázku 5 předpis funkce $1_{\mathbb{R}}$? Je jednoznačně určený (tedy, může mít tato funkce i více předpisů)?*

Obrázkem 5 narážíme na první „lapálii“ teorie kategorií. Totiž, člověk by si řekl, že když jsou všechny objekty množinou reálných čísel, tak jsou přece všechny stejné a šipky α, β, γ, δ i ε prostě smyčky kolem tohoto jednoho objektu. Není tomu tak. Přestože všechny objekty *představují* stejnou věc, jsou rozdílné. Na podobnou myšlenku narazíme znovu později v podobě *isomorfismu* objektů. Nedokonalým příměrem je představa existence dvou (či více) do poslední mrtě stejných židlí umístěných kolem stolu. Židle to sice naprosto stejné jsou, ale liší se svým umístěním. Na obrázku 5 jsou všechny objekty „zevnitř“ stejné, ale vyjímají sebe šipkami, které z nich nebo do nich vedou.

Nyní se zamyslete, čím *přesně* je jeden objekt reálných čísel s šesti smyčkami rozdílný od kategorie na obrázku 5.

Úloha 2.3 [0,5b + 1b]:

1. *Vysvětlete, čím je kategorie*

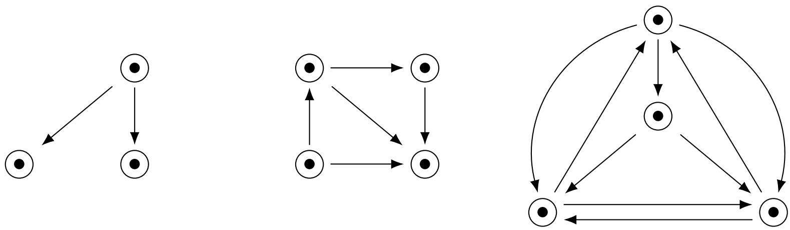


jiná než ta na obrázku 5.

2. *Upravme kategorii na obrázku 5 tak, aby mezi každými jejími dvěma objekty nevedla jen jedna lineární funkce, ale úplně všechny. Znovu a lépe: šipky mezi dvěma objekty jsou nyní všechny lineární funkce $\mathbb{R} \to \mathbb{R}$. Je toto vůbec kategorie? Ano-li, je stejná jako kategorie s* jedním *objektem reálných čísel a smyčkou opět pro každou lineární funkci?*

Není lepšího způsobu, jak ověřit pochopení definice, než příkladem. Proto si hned v další úloze vyzkoušíte rozpoznat, zda daný obrázek může být (za ztotožnění některých šipek) celou kategorií.

Úloha 2.4 [1b]: *Rozhodněte, které z následujících obrázků* mohou *reprezentovat celé kategorie. Tím myslíme, zda lze splnit všechny podmínky z definice kategorie, považujeme-li některé šipky za stejné. Odpověď aspoň stroze odůvodněte.*



Diagramy

Většinu užitečných kategorií v plné kráse nakreslit dost dobře nemůžeme. Důvodem je často fakt, že mají nekonečno objektů nebo nekonečno šipek. Případně jich mají prostě konečnou hromadu a jsou nepřehledné. *Diagramy* jsou elegantním způsobem, jak si lidé (dosaďte abstraktní algebraici) představují, co se v které kategorii děje. Jsou to v principu výřezy pouze těch objektů a šipek, jež nás v daný moment zajímají. Jejich užitečnost vyjde najevo již v sekci o universálních vlastnostech.

Příklady diagramů jsme dokonce viděli v definici kategorie. V podmínce *složitelnosti* šipek jsme představili tento *diagram*:

$$\begin{tikzcd} & \bullet \arrow[dr] & \\ \bullet \arrow[ur] & & \bullet. \end{tikzcd}$$

Slovy: „nastává-li v $\mathcal{A}$ situace," jsme pak přesně mínili: „dosadíme-li za puntíky libovolné objekty kategorie $\mathcal{A}$ a za šipky libovolné morfismy mezi příslušnými objekty." Vyřízli jsme z obrázku kategorie $\mathcal{A}$ jakýsi „podobrázek" tří objektů a dvou morfismů. Ona podmínka složitelnosti dí, že vždy najdu v kategorii $\mathcal{A}$ šipku z levého do pravého puntíku, která zařídí, aby diagram

$$\begin{tikzcd} & \bullet \arrow[dr] & \\ \bullet \arrow[ur] \arrow[rr] & & \bullet \end{tikzcd}$$

tzv. *komutoval* – to znamená, že cesty (posloupnosti šipek) mezi libovolnými dvěma puntíky v diagramu jsou si rovny (jako šipky v kategorii $\mathcal{A}$).

Při kreslení diagramů je obyčejně pohodlné označit puntíky a šipky písmeny, abychom se na ně mohli snadno odkazovat. Než ji konečně opustíme, podmínku složitelnosti ještě jednou přeformulujeme do její zcela rigorózní podoby: Pro libovolné objekty $a, b, c \blacktriangleleft \mathcal{A}$ a morfismy $a \xrightarrow{\alpha} b, b \xrightarrow{\beta} c$, existuje morfismus $a \xrightarrow{\gamma} c$ takový, že diagram

$$\begin{tikzcd} & b \arrow[dr, "\beta"] & \\ a \arrow[ur, "\alpha"] \arrow[rr, "\gamma"] & & c \end{tikzcd}$$

komutuje. Takovému morfismu řekneme *složení* α a β a píšeme $\gamma = \alpha\beta$.

Uvedeme pro názornost ještě jeden příklad. Tvrzení, že diagram

$$\begin{tikzcd} \bullet \arrow[r, "\alpha"] \arrow[d, "\beta"'] \arrow[dr, "\varepsilon"] & \bullet \arrow[d, "\delta"] \\ \bullet \arrow[r, "\gamma"'] & \bullet \end{tikzcd}$$

komutuje, v sobě skrývá platnost rovností $\varepsilon = \alpha\delta = \beta\gamma$. Slovy, všechny cesty z levého horního do pravého dolního rohu se musejí rovnat.

Úloha 2.5 [1,5b + 1,5b]: *Přeformulujte podmínky (AS) a (SM) přes komutativitu vhodně zvolených diagramů, podobně jako jsme my učinili s podmínkou (SK). Může se vám hodit poznámka, že dva i více různých puntíků v diagramu nemusí vůbec odpovídat různým objektům. Například*

$$a \xrightarrow{1_a} a,$$

je naprosto validní diagram pro jakýkoli objekt $a \blacktriangleleft \mathcal{A}$.

Universální vlastnost

Kterési vlastnosti objektů jsou pro nás výjimečné a zajímavé. Vyznačují se tím, že jsou v silně kategoriálním smyslu jedinečné – všechny objekty, které jimi oplývají, už musejí tímto být v podstatě totožné (formálně *isomorfní*). Takovým vlastnostem pak kategorici říkají *universální*[23] a objekty, jež je mají, pojmenováváme různě – podle povahy oné vlastnosti či podle slavných matematiků, jejichž výzkum a tvorba s ní ideálně nemají nic společného.

Abychom vykreslili, které vlastnosti objektů nám mohou připadat „zajímavé“, uvažme například – snad, čtenáři, souhlasíte – výjimečné přirozené číslo $0 \in \mathbb{N}$ (otázkou, jak vidět přirozená čísla kategoriálními zornicemi, se budeme zabývat jindy). Číslo 0 má tu *jedinečnou* vlastnost, že jeho přičtením k libovolnému jinému přirozenému číslu se hodnota tohoto čísla nezmění. Je to vlastnost universální.

Jak bývá v teorii kategorií zvykem, dokážeme tento fakt sporem. Totiž, představme si, že jsou v přirozených číslech nuly dvě – označme je třeba $\oslash$ a $\obslash$. Pak platí

$$\oslash + \obslash = \obslash, \quad \text{ale taky} \quad \oslash + \obslash = \oslash,$$

[23]Intuitivní čtenáři sobě mohli povšimnout, že v jednu chvíli nazýváme vlastnosti *jedinečné*, v druhou *universální*, tedy v zásadě antonymy. Vinen jest překlad z angličtiny, kde slovo *universal* znamená též *zahrnující všechny členy jisté třídy či skupiny* (v tomto případě třídy vzájemně isomorfních objektů). V češtině slovo *universální* tento význam nenese; vhodnějším (avšak nepoužívaným) jménem je pročež v této situaci již užité *jedinečný*.

protože $\rotatebox{-45}{$0$}$ ani $\rotatebox{45}{$0$}$ nemění hodnotu čísla, ke kterému je přičtena. Obě rovnosti dohromady znamenají, že $\rotatebox{-45}{$0$} = \rotatebox{45}{$0$}$, takže si vpravdě jsou všechny nuly rovny.

Přirozená čísla jsou žel bohu příliš naivním příkladem. V obecných kategoriích se nevyplácí požadovat, aby dva objekty se stejnou universální vlastností byly doslova *stejné*. Rovnost je pevný vztah, který v matematických strukturách nastává jen ojediněle. Spokojíme se s tím, že přechodem od jednoho k druhému „neztratíme žádnou informaci“. Párům takových objektů se říká *isomorfní* (z řec. *isos*, „stejný“ a *morphé*, „tvar“). Formálně, dva objekty a,$b \blacktriangleleft \mathcal{A}$ jsou *isomorfní* (zapisujeme $a \cong b$), existují-li šipky $a \xrightarrow{\alpha} b$ a $b \xrightarrow{\beta} a$ takové, že jejich složení je v jednom pořadí identita na a a v druhém pořadí identita na b. To jest, platí jak $\alpha\beta = 1_a$, tak $\beta\alpha = 1_b$. Pomáhá se na isomorfismus dívat tak, že jít do isomorfního objektu a zase zpátky je totéž, jako zůstat na místě.

V jazyce diagramů se dá isomorfismus $a \cong b$ popsat existencí šipek $a \xrightarrow{\alpha} b$ a $b \xrightarrow{\beta} a$, které zařizují, že následující diagramy

$$\begin{tikzcd} a \arrow[r, "\alpha"] \arrow[dr, "1_a"'] & b \arrow[d, "\beta"] \\ & a \end{tikzcd} \qquad \begin{tikzcd} b \arrow[r, "\beta"] \arrow[dr, "1_b"'] & a \arrow[d, "\alpha"] \\ & b \end{tikzcd}$$

komutují.

Úloha 2.6 [2b]: *Ukažte (diagramy, úvahou, ...), že „býti isomorfní s“ je universální vlastnost. Šířeji řečeno, pro libovolný objekt $a \blacktriangleleft \mathcal{A}$ ukažte, že všechny objekty s ním isomorfní jsou rovněž vzájemně isomorfní.*

Jsouce vyzbrojeni pojmem isomorfismu, podáme poněkud formálnější popis universality: vlastnost X je v kategorii $\mathcal{A}$ *universální*, když pro každé dva objekty $a, b \blacktriangleleft \mathcal{A}$ mající vlastnost X platí, že $a \cong b$.

Iniciální a terminální objekt

Snad nejjednodušším příkladem objektů s universální vlastností jsou tzv. *iniciální* (chcete-li, *počáteční*) a *terminální* (chcete-li, *koncové*) objekty. Jak jeho přídomek částečně napovídá,

- *iniciální* objekt je takový objekt, ze kterého vede přesně jedna šipka do každého objektu;
- *terminální* objekt je takový objekt, do kterého vede přesně jedna šipka z každého objektu.

Například, v kategorii

$$\begin{tikzcd} & \odot \arrow[dr] & \\ \odot \arrow[ur] \arrow[rr] \arrow[dr] & & \odot \\ & \odot \arrow[ur] & \end{tikzcd}$$

je puntík nalevo iniciální a puntík napravo terminální objekt.

Rozmyslíme si, že „býti iniciální“ je universální vlastnost. To znamená, že každé dva iniciální objekty jsou isomorfní. Ať jsou tedy $i, j \blacktriangleleft \mathcal{A}$ dva iniciální objekty v kategorii $\mathcal{A}$. Pak máme přesně jednu šipku $i \xrightarrow{\eta} j$ (protože i je iniciální) a přesně jednu šipku $j \xrightarrow{\kappa} i$ (protože j je iniciální). Chceme, aby $\eta\kappa = 1_i$ a $\kappa\eta = 1_j$, čili, aby i a j byly isomorfní prostřednictvím šipek η a κ. To ale musí být pravda, jelikož $\eta\kappa$ je šipka $i \to i$ a taková existuje jenom jedna (neboť i je iniciální), konkrétně 1_i. Rovněž, $\kappa\eta$ je šipka $j \to j$, takže musí být rovna 1_j.

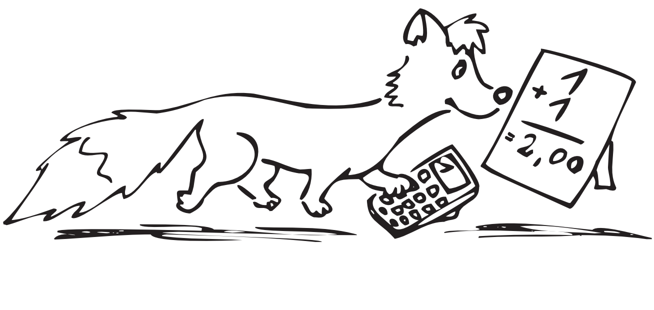
Podobná úvaha vede i k universalitě terminálního objektu. Podrobný argument zkuste sestrojit v rámci následující úlohy.

Úloha 2.7 [1b]: *Argumentujte, že „býti terminální“ je v kategorii $\mathcal{A}$ universální vlastnost, tj. všechny terminální objekty v $\mathcal{A}$ jsou sobě isomorfní.*

Universální vlastnosti jsou snad nejdůležitějším konceptem teorie kategorií a brzy je začneme používat ke stavbě zajímavých věcí z logiky a teorie množin.

Problém 2.8: *Jednou významnou kategorií je kategorie obvykle značená* ***Set*** *– kategorie množin. Jejími objekty jsou množiny*[24] *(tj.* všechny *množiny) a šipkami mezi dvěma množinami jsou přesně všechna zobrazení*[25] *(či funkce), která mezi nimi vedou. Vysvětlete, co jsou v kategorii* ***Set*** isomorfismy *a* iniciální a terminální objekty. *Napadne vás příklad nějaké jiné universální vlastnosti množin?*

Adam; `kategoricketematko@gmail.com`
odevzdávejte do odevzdávátka



[24]Pro připomenutí či seznámení doporučujeme článek `cs.wikipedia.org/wiki/Množina`.

[25]O zobrazeních si můžete přečíst například v článku na Wikipedii: `https://cs.wikipedia.org/wiki/Zobrazení_(matematika)`.

Téma 6 – Speciální teorie relativity

Díl 1: Lorentzovy transformace

Úvod

V tomto tématku si představíme speciální teorii relativity, což je jedna z teorií, které proslavily nikoho menšího než Alberta Einsteina. Naštěstí pro nás je tato jeho teorie, nebo alespoň její základy, poměrně jednoduchá. Speciální teorie relativity (dále jen STR) nám říká, jak se chovají objekty, jejichž rychlost se blíží rychlosti světla. Než však začneme tuto teorii budovat, je nutné si ujasnit pár pojmů.

Nejdříve si přiblížíme, co myslíme pojmem *pozorovatel*. Pozorovatel je entita (můžeme si například představit člověka), která dokáže měřit různé veličiny (například čas nebo vzdálenost).

Referenční soustava (vztažná soustava) je zvolená skupina těles (nebo jen jedno referenční těleso), která jsou vzájemně v klidu nebo ve známém vzájemném pohybu. Poloha a pohyb těles jsou pak vztahovány vzhledem ke zvolené referenční soustavě. Mějme například dva pozorovatele stojící kus od sebe na chodníku, ve kterém je díra. Od pozorovatele A je díra jeden metr napravo a od pozorovatele B zase tři metry nalevo. Je jasné, že každý pozorovatel nám o poloze díry řekne něco jiného. To je právě proto, že používají jiné referenční soustavy.

Inerciální referenční soustava je taková soustava, ve které pro volnou částici platí Newtonovy pohybové zákony v nejjednodušším tvaru, neboli volná částice se pohybuje rovnoměrně přímočaře. Všechny soustavy, které se vůči nějaké inerciální referenční soustavě pohybují rovnoměrně přímočaře nebo jsou vůči ní v klidu, jsou také inerciální. Naopak soustava pohybující se vůči inerciální referenční soustavě se zrychlením je neinerciální.

Nyní, když jsme vysvětlili důležité pojmy, můžeme začít budovat STR. Ta vychází z jednoduchého principu.

Na relativním počátku času Albert Einstein postuloval, že fyzikální zákony platí ve všech inerciálních soustavách stejně. Je dobré zmínit, že to se týká i zákonů popisujících elektromagnetismus, což znamená, že i rychlost světla je ve všech inerciálních soustavách stejná.

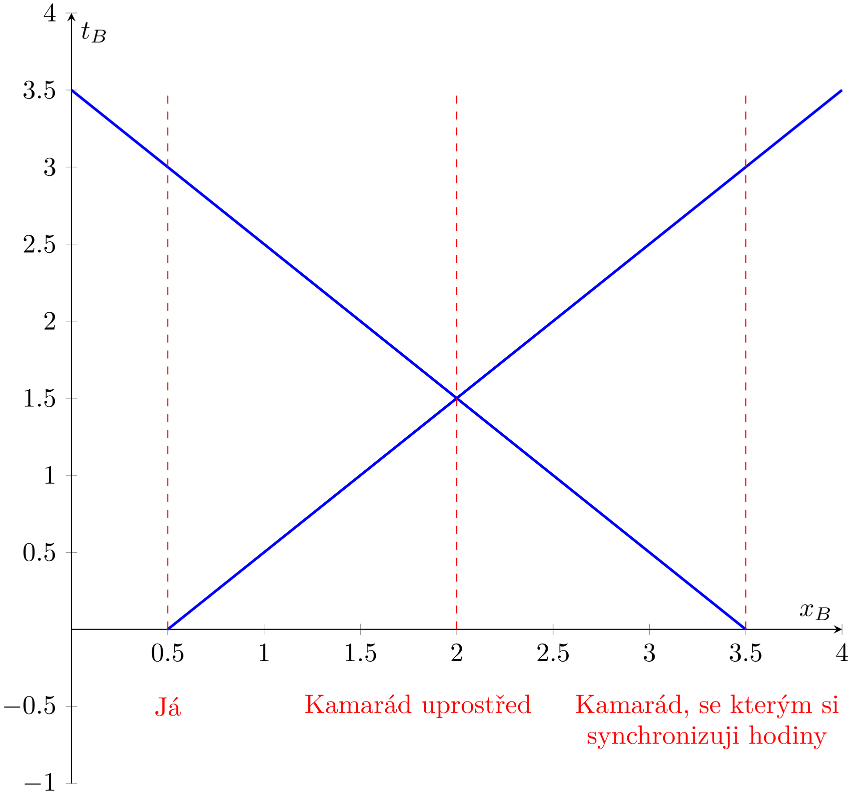
Tento postulát má ve fyzice nedozírné následky, jejichž malou část si nyní představíme.

Prostoročasové diagramy

V STR uvažujeme o klasickém prostoru a čase jen jako o složkách tvořících prostor o čtyřech dimenzích, kterému říkáme prostoročas. Prostoročas má jednu časovou osu a tři prostorové (alespoň v kartézských souřadnicích). Když řešíme problém v STR, většinou je prvním krokem si nakreslit prostoročasový diagram. Přestože neumíme kreslit ve více než třech dimenzích, ve většině případů nám to ani nevadí. Dobrou intuici lze získat i z obyčejného 2D diagramu o jedné časové a jedné prostorové ose.

Obvykle zavádíme svislou osu jako časovou a vodorovnou osu jako prostorovou. Pozorovatelé se vždy pohybují v takových diagramech po přímkách. Pokud bychom s takovými diagramy pracovali v nějakých „klasických" jednotkách jako třeba metrech a sekundách, přímka představující světelný paprsek by byla prakticky nerozlišitelná od prostorové osy. Proto většinou zavádíme jednotky tak, aby $c = 1$, kde c značí rychlost světla. Příkladem takových jednotek by mohly být třeba sekundy a světelné sekundy, tedy vzdálenost, kterou světlo urazí za jednu sekundu. Příklad jednoho diagramu je na obrázku 8.

Vzhledem k užitečnosti rovnosti $c = 1$ je tato rovnost konvencí, kterou budeme dodržovat i my.



Obrázek 8: Přímky se sklonem 45° představují světelné paprsky a svislé přerušované přímky pak nehybné pozorovatele. (V PDF verzi jsou obrázky barevné a možná budou lépe čitelné.)

Koncept simultaneity a seřizování hodin

Postulát(y) STR jde využít k definování konceptu synchronicity. Představme si, že chceme synchronizovat svoje hodiny s hodinami kamaráda. Jsme ale trochu líní a nechceme k němu prostě dojít. Zavoláme si na pomoc dalšího kamaráda, ať si stoupne přesně mezi nás (což si musí odměřit, takže se zeptáme někoho s metrem). S kamarádem jsme se domluvili, že přesně ve 12:00 vyšleme k osobě doprostřed světelný paprsek. Pokud naměří, že paprsky doletěly ve stejný okamžik, naše hodiny jsou synchronizované. Tuto situaci zachycuje obrázek 8. Všechny body v prostoročase, ze kterých by byly ve 12:00 vyslány paprsky, které by se s tím naším potkaly v naší referenční soustavě v polovině vzdálenosti mezi námi a tímto bodem, nazveme simultánní nebo také synchronní.

Pokud takto definujeme synchronní události, je jasné, že pohybující se pozorovatel bude jako synchronní vnímat něco jiného, protože by se k němu jeden paprsek dostal dříve a mohl by prohlásit, že synchronizované hodiny nemáme.

Rozmyslete si, proč by tento experiment fungoval pouze se světelnými paprsky a ne například s kulkami z pistole nebo se zvukem.

Odvození Lorentzových transformací

Představme si nyní nejdůležitější koncept STR, Lorentzovy transformace. Lorentzovy transformace jsou relativistický způsob, jak přecházet mezi různými inerciálními referenčními soustavami. Mějme dva pozorovatele, Alici a Boba, kteří mají synchronizované hodiny podle postupu výše. Alice jede ve vlaku rovnoměrně přímočaře a Bob stojí na nádraží. Koleje vlaku budou sloužit jako x-ová osa a směr jízdy vlaku zvolíme jako směr, kde x roste. Označme v rychlost vlaku, x_B polohu Alice v Bobově referenční soustavě a x_A polohu Alice v její referenční soustavě. Nyní z klasické fyziky:

$$x_B = x_A + vt_A, \tag{1}$$

$$t_B = t_A,$$

kde čas Boba a Alice se z naší zkušenosti shoduje.

Pokud ovšem Alice vyšle světelný paprsek, nemůžeme použít rovnici 1 k popisu polohy paprsku, protože by Bob vnímal rychlost světla jinou než c. Musíme tedy svoje rovnice upravit takovým způsobem, aby se shodovaly s našimi postuláty.

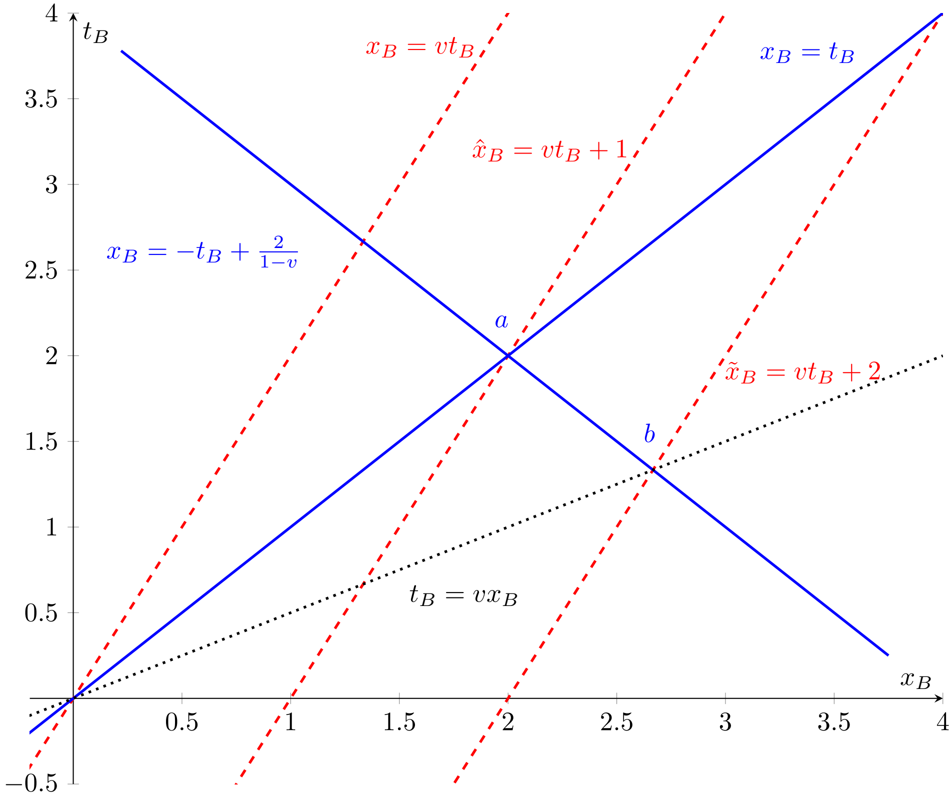
Začneme synchronizací hodin ve vagonu, ale z pohledu Boba. Ještě řekněme, že na $t = 0$ se shodnou oba pozorovatelé. Mějme dvoje hodiny, každé na jedné straně vagonu, a Alici stojící uprostřed. Z Bobova pohledu by šlo zapsat polohy hodin a Alice takto:

$$x_B = vt_B,$$

$$\hat{x}_B = vt_B + 1,$$

$$\tilde{x}_B = vt_B + 2,$$

kde jedna jednotka je polovina délky vagonu, x_B a $\tilde{x}_B$ jsou polohy hodin a $\hat{x}_B$ je poloha Alice. Nyní od hodin vyšleme světelné paprsky v takové časy, aby se



Obrázek 9: Diagram znázorňující synchronizaci hodin v pohybujícím se vlaku z pohledu stojícího pozorovatele. Přímky se sklonem 45^o jsou světelné paprsky. Přerušované přímky jsou polohy hodin a Alice. Tečkovaná přímka pak značí simultánní body v soustavě vlaku. (V PDF verzi jsou obrázky barevné a možná budou lépe čitelné.)

potkaly u Alice, tedy aby je Alice prohlásila za synchronizované. Z diagramu na obrázku 9 je vidět, že tyto časy pro Boba nejsou simultánní. Bod v prostoročase, kde se paprsky protnou, označme a. Je jasné, že bod a leží na přímce $x_B = t_B$ $(c = 1)$ a na $x_B = vt_B + 1$. Tedy pro bod a:

$$x_B = \frac{1}{1-v},$$

$$t_B = \frac{1}{1-v}.$$

Bod na konci vlaku označme b. Bod b musí ležet na přímce $x_B = vt_B + 2$ a zároveň na nějaké přímce popisující dráhu paprsku. Ta obecně vypadá (pro paprsek mířící do záporu):

$$x_B + t_B = \text{konst.}$$

Víme, že na této přímce leží bod a. Po dosazení:

$$x_B + t_B = \frac{2}{1-v}.$$

Nyní už jen vyřešíme soustavu dvou rovnic:

$$x_B = \frac{2}{1-v^2},$$

$$t_B = \frac{2v}{1-v^2}.$$

Tedy přímka, kde $t_A = 0$, je $t_B = vx_B$. Všimněme si, že to je symetrické s přímkou, kdy $x_A = 0$, tedy $x_B = vt_B$.

Dobrá, z toho plynou nové transformační rovnice (protože když $t_A = 0$, tak $t_B - vx_B = 0$, tedy $t_A = t_B - vx_B$, a analogicky u pozice), jen si musíme dát pozor při jejich zápisu, protože je obecně můžeme vynásobit ještě nějakým faktorem f, resp. g, aniž by to změnilo to, co jsme zatím vyvodili (protože $0 \cdot f = 0$):

$$x_A = (x_B - vt_B) \cdot f,$$

$$t_A = (t_B - vx_B) \cdot g.$$

Dobrá, ale pro paprsek světla musí platit:

$$x_A = t_A,$$

$$x_B = t_B.$$

Dosazením:

$$(x_B - vx_B) \cdot f = (x_B - vx_B) \cdot g,$$

$$f = g.$$

Nyní nám zbývá určit faktor f. Na to využijeme fakt, že jsme doteď uvažovali to, že Alice se pohybuje s rychlostí v a Bob stojí. Ale z pohledu Alice se přece pohybuje Bob, akorát že s rychlostí $-v$. Tedy:

$$x_B = (x_A + vt_A) \cdot f,$$

$$t_B = (t_A + vx_A) \cdot f.$$

Nyní jen dosadíme vztahy pro x_A a t_A:

$$x_B = ((x_B - vt_B) \cdot f + v \cdot (t_B - vx_B) \cdot f) \cdot f.$$

Vyřešíme:

$$x_B = (x_B - vt_B + vt_B - v^2 x_B) \cdot f^2,$$

$$x_B = (1 - v^2) \cdot x_B f^2,$$

$$f = \frac{1}{\sqrt{1 - v^2}},$$

kde volíme pozitivní odmocninu, abychom při transformaci zachovali směr růstu našich souřadnic. Finální verze naší práce:

$$t_A = \frac{t_B - vx_B}{\sqrt{1 - v^2}}, \tag{2}$$

$$x_A = \frac{x_B - vt_B}{\sqrt{1 - v^2}}.$$

Rovnice 2 se nazývají Lorentzovy transformace. Ty budeme občas označovat zkratkou LT.

Dilatace času

Z Lorentzových transformací plyne několik zajímavých efektů. Jedním z nich je fakt, že stacionárnímu pozorovateli se zdá, jako kdyby pohybujícímu pozorovateli plynul čas pomaleji. Tento jev nazýváme dilatace času. Představme si, že kolem nás projíždí velmi rychlý vlak rychlostí v. Na našich hodinách během jeho průjezdu uplyne Δt času. Kolik času uplyne na hodinách ve vlaku?

Nejdříve si pojmenujeme svoji referenční soustavu A a soustavu vlaku V. Platí:

$$t_V = \frac{t_A - vx_A}{\sqrt{1 - v^2}},$$

$$x_V = \frac{x_A - vt_A}{\sqrt{1 - v^2}}.$$

Pro jednoduchost řekneme, že naše poloha je $x_A = 0$, a vzhledem k tomu, že se nepohybujeme, se nebude měnit. Víme, že $\Delta t = t_{1A} - t_{2A}$, kde $t_{1A} > t_{2A}$ jsou libovolné časy, například čas, kdy nás minula lokomotiva, a čas, kdy nás minul poslední vagon. Jak tento rozdíl bude vypadat v referenční soustavě vlaku? Jednoduše provedeme Lorentzovu transformaci:

$$\frac{t_{1A}}{\sqrt{1 - v^2}} - \frac{t_{2A}}{\sqrt{1 - v^2}} = \frac{\Delta t_A}{\sqrt{1 - v^2}} = \Delta t_V.$$

Tedy ve vlaku uplyne $\frac{1}{\sqrt{1-v^2}}$krát času méně. To znamená, že jako pozorovatel v klidu bychom viděli hodiny ve vlaku tikat $\frac{1}{\sqrt{1-v^2}}$krát pomaleji.

Skládání rychlostí

Na začátku odvozování LT jsme si ukázali, že ve STR nemůžeme tak naivně skládat rychlosti jako v klasické fyzice. Nyní bychom chtěli odvodit relativistickou verzi vzorce pro skládání rychlostí.

Mějme pozorovatele A stojícího v klidu, pozorovatele B jedoucího ve vlaku a pozorovatele C jedoucího v autě, které je v již zmíněném vlaku. Vlak jede oproti pozorovateli A rychlostí v a auto jede oproti pozorovateli B rychlostí u. Platí:

$$t_A = \frac{t_B - vx_B}{\sqrt{1-v^2}}, \quad t_B = \frac{t_C - ux_C}{\sqrt{1-u^2}},$$

$$x_A = \frac{x_B - vt_B}{\sqrt{1-v^2}}, \quad x_B = \frac{x_C - ut_C}{\sqrt{1-u^2}}.$$

Hledáme rychlost w, aby platilo:

$$t_A = \frac{t_C - wx_C}{\sqrt{1-w^2}},$$

$$x_A = \frac{x_C - wt_C}{\sqrt{1-w^2}}.$$

Dosadíme do vztahů výše:

$$t_A = \frac{\frac{t_C - ux_C}{\sqrt{1-u^2}} - v\left(\frac{x_C - ut_C}{\sqrt{1-u^2}}\right)}{\sqrt{1-v^2}}.$$

Dostáváme:

$$t_A = \frac{(1+uv)t_C - (u+v)x_C}{\sqrt{1-u^2-v^2+u^2v^2}},$$

$$t_A = \frac{(1+uv)(t_C - \frac{u+v}{1+uv}x_C)}{(1+uv)\sqrt{\frac{1-u^2-v^2+u^2v^2}{(1+uv)^2}}},$$

$$t_A = \frac{t_C - \frac{u+v}{1+uv}x_C}{\sqrt{1-\left(\frac{u+v}{1+uv}\right)^2}}.$$

Vidíme tedy, že:

$$w = \frac{u+v}{1+uv}. \tag{3}$$

Úlohy

Úloha 2.1 [2b]: *Mějme dva pozorovatele – Alici a Boba. Alice je ve vlaku, který jede rovnoměrně přímočaře rychlostí $v = 0{,}5$ ($c = 1$ podle konvence výše). Alice stojí přesně uprostřed vlakového vagonu, který měří 16 m, a Bob stojí na nádraží.*

Alice ve stejný okamžik vyšle dva laserové paprsky, každý na jednu stranu vagonu. Spočítejte, za jak dlouho dopadnou paprsky na stěny vagonu z pohledu Alice a z pohledu Boba. Zamyslete se nad tím, jestli dopadnou současně.

Úloha 2.2 [3b]: *Analogickým postupem, jako jsme odvodili dilataci času, odvoďte takzvanou kontrakci délek, neboli jak se bude měnit metr pozorovatele v pohybu vůči pozorovateli v klidu.*

Úloha 2.3 [5b]: *Skalární součin je velmi praktická matematická operace a my bychom ho chtěli používat i pro čtyřvektory (tj. vektory v prostoročase). Úplně obecně se skalární součin mezi vektory* **u** *a* **v** *spočte jako (v maticové notaci):*

$$\boldsymbol{u}^T A \boldsymbol{v} = k\,,$$

kde matice A je symetrická matice a k je konstanta.

Při změně vztažné soustavy by se neměly měnit hodnoty skalárů (pouhých čísel, dobrým příkladem je třeba teplota tělesa). Když měníme vztažnou soustavu v prostoročase, používáme výše uvedené Lorentzovy transformace. Odvoďte, jak musí vypadat matice A ve dvou dimenzích, jedné časové a jedné prostorové, pokud chceme, aby vektor ukazující jednu jednotku v čase dopředu a nula v prostoru měl velikost 1. (Skalární součin vektoru se sebou samým berte jako čtverec velikosti tohoto vektoru.)

Hint: Lorentzova transformace lze zapsat maticí:

$$\Lambda = \gamma \begin{bmatrix} 1 & -v \\ -v & 1 \end{bmatrix},$$

pro $c = 1$ *a* $\gamma = \frac{1}{\sqrt{1-v^2}}$.

Bonus [1b]: Odvoďte tvar matice A ve čtyřech dimenzích.

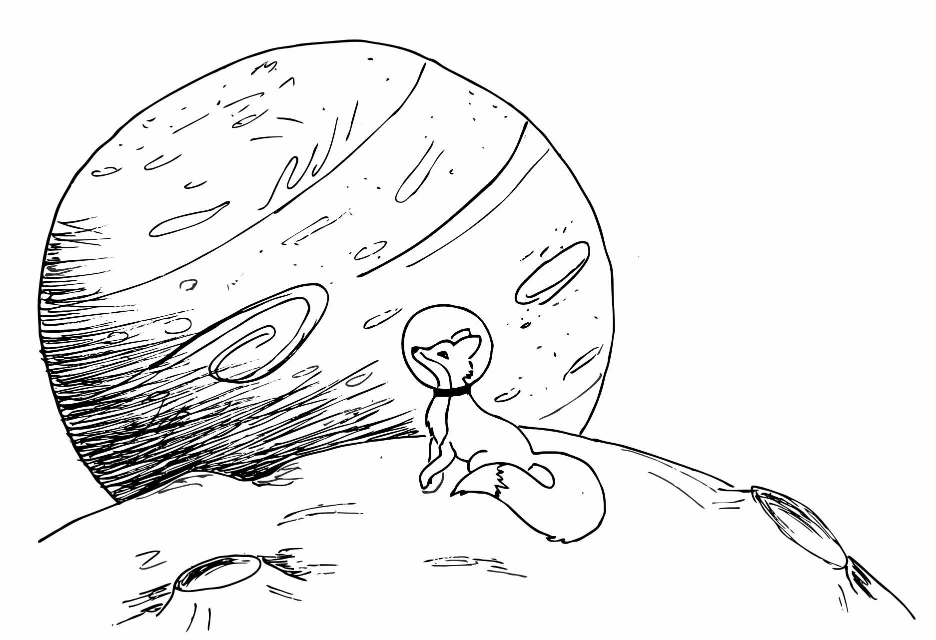
Problém 2.4: *Jaké relativistické efekty a jak silně bychom zažívali, kdybychom se pohybovali rychlostí světla? Předpokládejme, že jsme této rychlosti dosáhli pomocí magie. Hint: Je to chyták.*

Radim N.; `radim05@post.cz`
Honza T.; `jan.tregler@seznam.cz`
odevzdávejte do odevzdávátka

Výsledky 1. deadlinu 1. čísla

				Témata						
Poř.	Jméno	R.	$\sum_{-1}$	1	2	3	4	O	$\sum_0$	$\sum_1$
1.	Mgr.$^{\mathsf{MM}}$ P. Starý	3	50,2	7,7	11,4	20,1	11,0		50,2	50,2
2.	Doc.$^{\mathsf{MM}}$ J. Uglickich	3	376,0	5,5	12,4	17,8		12,0	47,7	47,7
3.	Doc.$^{\mathsf{MM}}$ L. Zůnová	4	228,2	3,4	9,0	23,8	8,8		45,0	45,0
4.	Mgr.$^{\mathsf{MM}}$ A. Trnková	4	71,2	5,3	12,4	25,6			43,3	43,3
5.	Doc.$^{\mathsf{MM}}$ O. Sedláček	4	263,0		10,5	20,0			30,5	30,5
6.–7.	Dr.$^{\mathsf{MM}}$ D. Kaňka	3	173,8	4,1	10,6		14,5		29,2	29,2
	Doc.$^{\mathsf{MM}}$ M. Jarvis	3	217,4		8,7	20,5			29,2	29,2
8.	Doc.$^{\mathsf{MM}}$ P. Šimová	4	209,5	6,1	13,0		8,0		27,1	27,1
9.	Bc.$^{\mathsf{MM}}$ M. Pustková	4	27,0	2,8	9,1	15,1			27,0	27,0
10.	Mgr.$^{\mathsf{MM}}$ K. Kučerová	Z9	98,2	4,5	10,5	1,0	9,6		25,6	25,6
11.	Bc.$^{\mathsf{MM}}$ S. Ožanová	3	24,6	6,5	9,3		8,8		24,6	24,6
12.	Bc.$^{\mathsf{MM}}$ F. Nouza	3	27,4	3,5	9,9	10,0			23,4	23,4
13.	Bc.$^{\mathsf{MM}}$ S. Šimečková	3	21,9			20,0	1,9		21,9	21,9
14.	Doc.$^{\mathsf{MM}}$ M. Těšitel	4	250,9			18,1	3,0		21,1	21,1
15.	Mgr.$^{\mathsf{MM}}$ J. Jedlička	3	79,7		9,0	7,0	2,5		18,5	18,5
16.	Bc.$^{\mathsf{MM}}$ K. Maxera	4	49,3		9,7		8,1		17,8	17,8
17.	P. Bělušová	3	17,5	2,5	5,0		10,0		17,5	17,5
18.	A. Gauchet	3	17,0		9,5		7,5		17,0	17,0
19.	Doc.$^{\mathsf{MM}}$ J. Klementová	3	223,8		11,0		3,0		14,0	14,0
20.	V. Šitra	3	13,5	2,9	9,1	1,5			13,5	13,5
21.	F. Dvořák	2	13,3	2,5	10,8				13,3	13,3
22.	N. Jochová	2	13,0	2,5	10,5				13,0	13,0
23.	Dr.$^{\mathsf{MM}}$ M. Urbanová	2	120,7					12,0	12,0	12,0
24.–25.	K. Bouchalová	Z9	11,4		11,4				11,4	11,4
	Mgr.$^{\mathsf{MM}}$ V. Kučera	3	62,0	1,0	5,2		5,2		11,4	11,4
26.	Dr.$^{\mathsf{MM}}$ M. Ambros	2	122,0		11,3				11,3	11,3
27.	Bc.$^{\mathsf{MM}}$ R. Petit	4	48,1		8,7	2,4			11,1	11,1
28.	R. Zelený	4	14,2		10,4				10,4	10,4
29.	Mgr.$^{\mathsf{MM}}$ J. Kadlec	4	57,6		7,3		3,0		10,3	10,3
30.	Doc.$^{\mathsf{MM}}$ O. Nevěřil	3	219,6		10,2				10,2	10,2
31.	M. Kramešová	4	5,4		5,4				5,4	5,4
32.	M. Štěpán	4	3,0		3,0				3,0	3,0
33.	Bc.$^{\mathsf{MM}}$ K. Česká	4	28,7	1,5					1,5	1,5
34.	M. Těšitel	Z1	1,0		1,0				1,0	1,0

Sloupeček $\sum_{-1}$ je součet všech bodů získaných v našem semináři, $\sum_0$ je součet bodů v tomto deadlinu 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.



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