Algebra s aplikacemi (studijní opora)

Martin Kuřil

Obsah

1 Úvod **2**

2 Celá čísla **6**
2.1 Úvod 6
2.2 Indukce 7
2.3 Algoritmus dělení: GCD a LCM 12
2.4 Prvočísla 22
2.5 Modulární aritmetika 32
2.6 Řešení lineárních kongruencí 35
2.7 Eulerova věta 43
2.8 Kryptografie s veřejným klíčem 52
2.8.1 RSA kryptosystém 52
2.8.2 Analýza kryptosystému RSA 60

3 Grupy **66**
3.1 Základní pojmy teorie grup 67
3.1.1 Definice grupy 67
3.1.2 Mocniny 67
3.1.3 Homomorfismy 67
3.1.4 Podgrupy 67
3.1.5 Součiny grup 67
3.2 Příklady grup 67
3.2.1 Aditivní grupa okruhu 67
3.2.2 Grupa jednotek okruhu 67
3.2.3 Symetrická grupa 67
3.2.4 Alternující grupa 67
3.2.5 Obecná lineární grupa 68

3.2.6 Grupa symetrií obrazce 68
3.2.7 Kvaterniony 68
3.3 Lagrangeova věta a její důsledky 68
3.3.1 Lagrangeova věta 68
3.3.2 Věty Fermatova a Eulerova 68
3.4 Cyklické grupy 68
3.4.1 Popis všech cyklických grup 68
3.4.2 Podgrupy cyklických grup 68

4 Algebraické struktury 68
4.1 Definice algebraické struktury 68
4.2 Pologrupy a monoidy 70
4.3 Okruhy a tělesa 72
4.4 Algebra polynomů 79
4.5 Euklidův algoritmus pro polynomy 86

5 Samoopravné kódy 96
5.1 Opakovací kódy a kódy kontroly parity 96
5.2 Lineární kódy 99
5.3 Hammingovy kódy 105
5.4 Úvod do BCH kódů pro opravu dvojnásobných chyb 109

6 Kořeny polynomů 110
6.1 Násobnost a počet kořenů polynomu 110
6.2 Základní věta algebry 113
6.3 Binomické rovnice 114
6.4 Kvadratické a kubické rovnice 115
6.5 Kořeny polynomů nad celými čísly 116
6.6 Hornerovo schéma 117

7 Pokyny (doporučení) k tempu studia 118

1 Úvod

Tento text má sloužit studentům jako pomůcka při studiu předmětu Algebra s aplikacemi.

Jednotlivé kapitoly (části) této studijní opory jsou zpracovány dvojím či vlastně trojím způsobem. V některé kapitole je probíraná látka v textu přímo

vyložena a výklad je doplněn několika cvičeními. Jindy je čtenáři (studentovi) po krátkém úvodu do problematiky uloženo, kde a co přesně má nastudovat. Někdy dokonce tento úvod do problematiky chybí a čtenáři je přímo uloženo samostudium, avšak takto postupuji pouze v tom případě, kdy čtenáře odkazuji na svůj studijní text [10]. Někdy následují další doporučení, například co dalšího by bylo dobré si přečíst. Ve většině případů je uloženo studium z textu [10]:

Martin Kuřil: *Základy algebry.*
`https://kma.ujep.cz/administrace/uploads/8f878f1.pdf`

Text je volně dostupný na internetu. V případě problémů s jeho získáním mi napište (adresu najdete na konci této studijní opory). Jde o studijní text, který je zatím ve fázi přípravy, avšak některé kapitoly jsou již hotové, například je hotová celá část věnovaná grupám (pochopitelně, i hotové části ještě mohou být změněny – hlavně budou opravovány případné chyby). Text je vhodný pro samostudium a jako studijní opora (nejen) pro studenty distanční a kombinované formy studia. Výklad je veden ve volném tempu a je provázen mnoha příklady. Důkazy tvrzení a vět jsou až nezvykle podrobné. Výhodou také jistě je, že studijní text, jak jsem se již zmínil, je volně dostupný na internetu, a to na mé stránce na stránkách Katedry matematiky Přírodovědecké fakulty UJEP.

Ve dvou případech je čtenáři uloženo studium z vysokoškolské učebnice [5]:

Jaroslav Blažek, Milan Koman, Blanka Vojtášková: *Algebra a teoretická aritmetika, II. díl.* Státní pedagogické nakladatelství, Praha, 1985.

Jde sice o knihu starší, avšak stále (jak se domnívám) dobře dostupnou – například Vědecká knihovna UJEP má ve svém fondu celkem 12 exemplářů této učebnice (stav ke dni 4.7.2022). Kniha byla v roce 1985 vydána jako celostátní vysokoškolská učebnice pro studenty matematicko-fyzikálních, přírodovědeckých a pedagogických fakult studijního oboru Učitelství všeobecně vzdělávacích předmětů aprobačního předmětu matematika.

Jak je na příslušných dvou místech uvedeno, alternativně je místo učebnice [5] možno použít knížku [11]:

Miroslav Šisler: *O řešení algebraických rovnic.* Mladá fronta, Praha, 1966.
https://dml.cz/handle/10338.dmlcz/403551

Jde sice opět o publikaci starší, avšak velmi dobře dostupnou – knížka je volně dostupná na internetu na stránce Czech Digital Mathematics Library `https://dml.cz/`. Je to webová stránka nabízející otevřený přístup k plným textům matematických časopisů, sborníků a knih vydávaných v historii v českých zemích. Velmi vám doporučuji využívat zmíněnou stránku!

Publikace Miroslava Šislera o řešení algebraických rovnic je součástí edice „Škola mladých matematiků". Tato edice byla založena roku 1961 na podnět Ústředního výboru Matematické olympiády, která tehdy v Československu již deset let úspěšně probíhala. Byla určena středoškolským studentům, rozšiřovala a prohlubovala jejich matematické vědomosti a přinášela řadu zajímavých příkladů.

Je jasné, že budete také potřebovat úlohy k procvičení probírané látky. Zde v této studijní opoře najdete řadu cvičení, a to především v těch kapitolách, ve kterých je látka přímo vyložena. Před chvíli jsem uvedl, že bude uloženo samostudium z několika výše uvedených textů. Z nich se úlohy k procvičení vyskytují v učebnici [5] a v knížce [11]. Kde vzít další cvičení (úlohy)? Jistě vám úlohy dá přímo váš vyučující nebo vám doporučí konkrétní (doufejme, že dobře dostupné) zdroje úloh. Já vám rozhodně doporučuji sbírku

David Stanovský: *Příklady z algebry.* Pracovní verze sbírky příkladů k základní přednášce z obecné algebry.
`https://www2.karlin.mff.cuni.cz/~stanovsk/vyuka/sbirka.pdf`

Ve sbírce najdete celkem 741 úloh. K některým jsou na konci uvedeny návody, k mnohým jsou uvedena řešení. Sbírka je volně dostupná na internetu na stránkách jejího autora (stav ke dni 4.7.2022).

Existují také další dobré a rozsáhlé sbírky úloh z algebry, například

A. K. Faddejev, J. S. Sominskij: *Zbierka úloh z vyššej algebry.* ALFA, Bratislava 1968.

Jde o slovenský překlad z ruského originálu. Sbírka obsahuje 1184 cvičení a také návody a výsledky k většině uvedených příkladů. Bohužel se však k této sbírce pravděpodobně dostanete jen s velkými obtížemi, například ve fondu Vědecké knihovny UJEP se nachází pouze jediný výtisk.

Dobrá je také sbírka úloh

J. Weil, J. Hocquemillerová, D. Allouch, A. Mézard, J.-C. Vaillant, Ch. Delorme, Ch. Lavitová, J.-C. Raoult: *Rozpracovaná řešení úloh z vyšší algebry.* Academia, Praha, 1987.

Jak název napovídá, řešení úloh jsou podrobně rozpracována. Fond Vědecké knihovny UJEP však obsahuje pouze tři výtisky.

Někteří z vás možná rádi čtou anglicky psanou literaturu. Těm mohu doporučit následující dva texty – jsou dobré a snadno se k nim dostanete:

Frederick M. Goodman: *Algebra: Abstract and Concrete.* Edition 2.6. Semi-Simple Press, Iowa City, IA, 2015.
`https://homepage.divms.uiowa.edu/~goodman/algebrabook.dir/book.2.6.pdf`

Text poskytuje důkladný úvod do abstraktní algebry. Kniha se věnuje tématům grupy, okruhy, tělesa. Kniha obsahuje spoustu cvičení. První a druhé vydání textu bylo publikováno nakladatelstvím Prentice-Hall. Současná verze je volně dostupná na internetu na stránkách autora knihy.

Druhým textem pak je

Thomas W. Judson: *Abstract Algebra: Theory and Applications*
`http://abstract.ups.edu/`

Judsonův text je určen pro jedno- nebo dvousemestrální bakalářský kurz abstraktní algebry. Text obsahuje aplikace, jako je teorie kódování a kryptografie. Na konci každé kapitoly je sada cvičení. Povaha cvičení se pohybuje v několika kategoriích: výpočetní, koncepční a teoretické problémy. Na konci textu je část s radami a řešeními mnoha cvičení. Často je v řešeních důkaz pouze načrtnutý a je na studentovi, aby uvedl podrobnosti. Cvičení se pohybují v obtížnosti od velmi jednoduchých až po velmi náročné.

Jednotlivé číselné obory budeme značit následovně:

- $\mathbb{N}$ – množina všech přirozených čísel, $\mathbb{N} = \{0, 1, 2, 3, \dots\}$
- $\mathbb{Z}$ – množina všech celých čísel, $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
- $\mathbb{Q}$ – množina všech racionálních čísel, $\mathbb{Q} = \{\frac{a}{b} |\ a, b \in \mathbb{Z}, b \neq 0\}$
- $\mathbb{R}$ – množina všech reálných čísel

- $\mathbb{C}$ – množina všech komplexních čísel, $\mathbb{C} = \{a + bi |\ a, b \in \mathbb{R}\}$

Připomeňme: Pro množiny A, B zápis $A \subseteq B$ znamená, že množina A je podmnožinou množiny B (tedy: pro každý prvek $x \in A$ platí, že $x \in B$). Zápis $A \subset B$ znamená, že $A \subseteq B$ a současně $A \neq B$.

Platí:

$$\emptyset \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$$

Také budeme používat následující značení:

- $\mathbb{Z}^+$ – množina všech kladných celých čísel, $\mathbb{Z}^+ = \{x \in \mathbb{Z} |\ x > 0\}$
- $\mathbb{Z}^-$ – množina všech záporných celých čísel, $\mathbb{Z}^- = \{x \in \mathbb{Z} |\ x < 0\}$
- $\mathbb{Q}^+$ – množina všech kladných racionálních čísel, $\mathbb{Q}^+ = \{x \in \mathbb{Q} |\ x > 0\}$
- $\mathbb{Q}^-$ – množina všech záporných racionálních čísel, $\mathbb{Q}^- = \{x \in \mathbb{Q} |\ x < 0\}$
- $\mathbb{R}^+$ – množina všech kladných reálných čísel, $\mathbb{R}^+ = \{x \in \mathbb{R} |\ x > 0\}$
- $\mathbb{R}^-$ – množina všech záporných reálných čísel, $\mathbb{R}^- = \{x \in \mathbb{R} |\ x < 0\}$

Nezapomeňte na to, že pokud budete na univerzitě studovat, pak budete mít svého konkrétního vyučujícího tohoto předmětu. V kontaktní části výuky vám některou část látky vyloží sám, může vám dát jiné odkazy na literaturu, než jsou ty zde uvedené, můžete s ním konzultovat.

V poslední kapitole najdete pokyny (doporučení) k tempu studia.

Toto je první verze studijní opory. Budou asi následovat verze další, snad lepší. Jistě v textu najdete nějaké chyby, nepřesnosti, ... Omlouvám se za ně a budu rád, když mi o nich dáte vědět.

2 Celá čísla

2.1 Úvod

V této kapitole budeme studovat vlastnosti množiny **celých čísel**

$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

a podmnožiny $\mathbb{N} \subset \mathbb{Z}$ **přirozených čísel** neboli nezáporných celých čísel

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}$$

Předpokládám, že znáte elementární logiku, množinovou notaci a základní vlastnosti **binárních operací sčítání** $(+)$ a **násobení** $(\cdot)$ a **relaci uspořádání menší nebo rovno** $(\leq)$ na množině celých čísel.

Tudíž hlavním objektem našeho studia v této kapitole bude uspořádaná čtveřice

$$(\mathbb{Z}, +, \cdot, \leq)$$

Tři další relace uspořádání souvisejí s $\leq$: $<$, $\geq$, $>$.

Princip dobrého uspořádání:
Každá neprázdná podmnožina množiny $\mathbb{N}$ má nejmenší prvek.

Poznámka. Princip dobrého uspořádání lze formulovat také pro množinu $\mathbb{Z}$. Jestliže $S \subseteq \mathbb{Z}$, $S \neq \emptyset$, S je zdola omezená, pak S má nejmenší prvek.

Cvičení.

1. Zformulujme princip dobrého uspořádání pro množinu $\mathbb{Q}$: Jestliže $S \subseteq \mathbb{Q}$, $S \neq \emptyset$, S je zdola omezená, pak S má nejmenší prvek. Ukažte na příkladu (včetně řádného zdůvodnění), že tento "princip"neplatí.

2. Pomocí operace $+$ definujte relaci $<$ na množině $\mathbb{N}$.

2.2 Indukce

Všimli jste si asi v předchozí části a všimnete si asi také v této části, že zde nejsou číslovaná tvrzení, věty, poznámky. Je tomu tak proto, že teď se zabýváme základy, principy, nevytváříme zatím sérii tvrzení. V další části již číslovat začneme.

Princip dobrého uspořádání je ekvivalentní s principem indukce.

Princip indukce:
Nechť $S \subseteq \mathbb{N}$. Předpokládejme, že S splňuje

1. $0 \in S$

2. $(\forall n \in \mathbb{N})\ n \in S \Longrightarrow n+1 \in S$

Pak $S = \mathbb{N}$.

Z principu dobrého uspořádání plyne princip indukce:
Nechť $S \subseteq \mathbb{N}$, S splňuje

1. $0 \in S$

2. $(\forall n \in \mathbb{N})\ n \in S \Longrightarrow n+1 \in S$

Chceme: $S = \mathbb{N}$.

Předpokládejme, že $S \neq \mathbb{N}$. Položme $T = \mathbb{N} - S$. Je $T \subseteq \mathbb{N}$, $T \neq \emptyset$ (jelikož $S \subseteq \mathbb{N}$, $S \neq \mathbb{N}$). Dle principu dobrého uspořádání existuje $a = \min T$. Protože $a \in T$, $a \notin S$, $a \neq 0$ (je totiž $0 \in S$). Pak existuje $b \in \mathbb{N}$, $a = b + 1$. Jsou dvě možnosti, obě dají spor:

- $b \notin S$: Pak $b \in T$, $a \leq b$, $b + 1 \leq b$, $1 \leq 0$, spor.
- $b \in S$: Pak $b + 1 \in S$, $a \in S$, spor.

Nutně tedy $S = \mathbb{N}$. To jsme chtěli.

Z principu indukce plyne princip dobrého uspořádání. To dokazovat nebudeme.

Princip dobrého uspořádání a princip indukce jsou axiomy – podstatně charakterizují přirozená čísla.

Princip indukce se často používá k důkazu tvrzení o přirozených číslech.
Nechť pro každé $n \in \mathbb{N}$ je dáno tvrzení $P(n)$.

Ukážeme:

1. $P(0)$ (platí, je pravdivé)

2. $(\forall n \in \mathbb{N})\ P(n) \Longrightarrow P(n+1)$

Pak $P(n)$ platí pro všechna $n \in \mathbb{N}$.

Proč to tak je? Proč to funguje?

Ke zdůvodnění použijeme princip indukce.

Buď $S = \{n \in \mathbb{N} |\ P(n)\}$, tj. S je množina všech přirozených čísel n, pro něž platí $P(n)$. Je $S \subseteq \mathbb{N}$. Dále,

1. $0 \in S$, protože $P(0)$ platí
2. $(\forall n \in \mathbb{N})\ n \in S \Longrightarrow n+1 \in S$

   Zdůvodnění: Buď $n \in \mathbb{N}$, $n \in S$. Takže $P(n)$ platí. Pak ovšem $P(n+1)$ platí, $n+1 \in S$.

Máme tedy:

1. $0 \in S$
2. $(\forall n \in \mathbb{N})\ n \in S \Longrightarrow n+1 \in S$

Princip indukce dává $S = \mathbb{N}$, $\{n \in \mathbb{N} |\ P(n)\} = \mathbb{N}$, $P(n)$ platí pro všechna $n \in \mathbb{N}$.

Poznámka. Obdobně postupujeme v případě, kdy $a \in \mathbb{Z}$ a pro každé $n \in \mathbb{Z}$, $a \leq n$, je dáno tvrzení $P(n)$. Chceme dokázat, že $P(n)$ platí pro každé $n \in \mathbb{Z}$, $a \leq n$. Důkaz lze udělat tak, že ukážeme dvě věci:

1. $P(a)$ platí
2. $(\forall n \in \mathbb{Z})\ a \leq n \wedge P(n) \Longrightarrow P(n+1)$

Princip dobrého uspořádání (a tedy také princip indukce) je ekvivalentní s následujícím principem:

Princip silné indukce:
Nechť pro každé $n \in \mathbb{N}$ je dáno tvrzení $P(n)$. Předpokládejme, že

1. $P(0)$ platí (je pravdivé)
2. $(\forall n \in \mathbb{N})\ P(0) \wedge P(1) \wedge \cdots \wedge P(n) \Longrightarrow P(n+1)$

Pak $P(n)$ platí pro všechna $n \in \mathbb{N}$.

Indukce je technika dokazování tvrzení $P(n)$ o přirozených číslech (či o celých číslech n, kde $a \leq n$ pro nějaké dané celé číslo a). Nejdříve vám musí někdo (třeba vyučující) dát $P(n)$ a pokyn "Dokažte, že pro všechna přirozená čísla n platí $P(n)$!". Jistě, tento pokyn se vám často nebude líbit (například v písemce). Nebo sami přidete k tomu, že by se hodilo něco dokázat indukcí. Při jaké příležitosti? Indukce není technika objevování. Nějak, třeba zkoumáním mnoha konkrétních případů, stanovíte hypotézu "Pro všechna přirozená čísla

n platí tvrzení $P(n)$."a pak teprve se indukcí pokusíte dokázat její platnost. Jeden malý a jednoduchý konkrétní příklad si teď uvedeme.

Příklad. Zkoumejme součty po sobě jdoucích lichých čísel, počínaje číslem 1:

$$\begin{aligned} 1 &= 1 \\ 1+3 &= 4 \\ 1+3+5 &= 9 \\ 1+3+5+7 &= 16 \\ 1+3+5+7+9 &= 25 \end{aligned}$$

Myslím, že hypotéza se ihned nabízí: Součet prvních n lichých čísel je roven n^2. Hypotézu se pokusíme dokázat. Nabízí se indukce, protože se jedná o tvrzení o přirozených číslech. Nejdříve bychom však měli hypotézu pořádně napsat. Zřejmě n–té kladné sudé číslo je $2n$ a první kladné liché číslo těsně předchází první sudé číslo (1 je $2-1$), takže n–té liché číslo (bereme teď v úvahu pouze kladná celá čísla) je rovno $2n-1$. Nyní již můžeme zformulovat naši hypotézu:

Nechť $P(n)$ je tvrzení

$$1+3+\cdots+(2n-1)=n^2$$

Naše hypotéza H tedy tvrdí: $P(n)$ platí pro všechna kladná celá čísla n.

Teď se můžeme pokusit o důkaz hypotézy H indukcí. Postupujeme ve dvou krocích:

1. $P(1)$: $1=1^2$ – to zřejmě platí
2. $(\forall n\in\mathbb{Z})\ 1\le n\wedge P(n)\Longrightarrow P(n+1)$: Nechť n je kladné celé číslo, pro něž platí $P(n)$, tj. $1+3+\cdots+(2n-1)=n^2$. Chceme ukázat, že platí $P(n+1)$, tj. $1+3+\cdots+(2n-1)+(2n+1)=(n+1)^2$. Počítejme:

$$\begin{aligned} 1+3+\cdots+(2n-1)+(2n+1) &= [1+3+\cdots+(2n-1)]+(2n+1) \\ &= n^2+(2n+1) \\ &= n^2+2n+1 \\ &= (n+1)^2 \end{aligned}$$

Při výpočtu jsme jednou použili informaci, že platí $P(n)$ – najdete to místo? Výpočtem jsme tedy ukázali, že $1+3+\cdots+(2n-1)+(2n+1) = (n+1)^2$, tedy ukázali jsme, že platí $P(n+1)$.

Nyní si můžeme být jisti, že naše hypotéza H opravdu platí – dokázali jsme ji indukcí. Není to již hypotéza, ale je to dokázaný matematický poznatek (věta).

Poznámka. Struktura $\mathbb{N}$ přirozených čísel je jedna z nejdůležitějších matematických struktur. Vážným pokusem o axiomatizaci struktury $\mathbb{N}$ je Peanova aritmetika PA. Studium Peanovy aritmetiky patří především do logiky. Přístupně jsou základy teorie PA vyloženy v kapitole V Přirozená čísla v učebnici [4]. Zájemce o podrobnější a hlubší studium Peanovy aritmetiky mohu odkázat na knihu [12], především na kapitolu 4 Peanova a Robinsonova aritmetika.

Cvičení.

1. Dokažte, že pro každé přirozené číslo n platí

$$0^3 + 1^3 + \cdots + n^3 = \sum_{0 \le k \le n} k^3 = \frac{1}{4} n^2 (n+1)^2$$

2. Dokažte, že pro každé celé číslo n, $1 < n$, platí

$$\frac{1}{\sqrt{1}} + \frac{1}{\sqrt{2}} + \cdots + \frac{1}{\sqrt{n}} = \sum_{1 \le k \le n} \frac{1}{\sqrt{k}} > \sqrt{n}$$

3. Dokažte, že pro každé celé číslo n, $4 \le n$, platí

$$2^n \ge n^2$$

4. Dokažte, že pro každé přirozené číslo n platí

$$6 / n^3 + 11n$$

   Poznámka: Pro $a, b \in \mathbb{Z}$ zápis a/b znamená $(\exists q \in \mathbb{Z})\ b = qa$.

5. Fibonacciho čísla F_n, $n \in \mathbb{N}$, jsou definována následovně:

$$F_0 = 0,\ F_1 = 1,\ F_n = F_{n-1} + F_{n-2} \text{ pro } n \geq 2$$

Dokažte, že pro každé kladné celé číslo n platí

$$F_n \leq \phi^{n-1}$$

Poznámka: Zde $\phi = \frac{1+\sqrt{5}}{2}$. Pravděpodobně použijete princip silné indukce.

6. Je $F_n^2 + F_{n+1}^2$ vždy Fibonacciho číslem? Odpověď zdůvodněte. Poznámka: Toto je úloha pro výzkumníky. Nejdříve si vytvořte hypotézu.

2.3 Algoritmus dělení: GCD a LCM

2.3.1. Definice. Nechť $a, b \in \mathbb{Z}$. Říkáme, že a **dělí** b (a je dělitelem b, b je dělitelné a, b je násobkem a), a píšeme a/b, pokud existuje $q \in \mathbb{Z}$ tak, že $b = qa$.

2.3.2. Poznámka.

1. Všimněte si, že pro všechna $a \in \mathbb{Z}$, $a/0$ ($0 = 0 \cdot a$). Avšak pouze 0 je násobkem čísla 0.

2. Varování: Nezaměňujte symboly a/b a $\frac{a}{b}$.

2.3.3. Tvrzení. *Nechť $a, b, c, \beta, \gamma \in \mathbb{Z}$. Jestliže a/b a a/c, pak $a/\beta b + \gamma c$.*

Důkaz. Sami jako cvičení.

2.3.4. Věta (algoritmus dělení). *Nechť $a, b \in \mathbb{Z}, a > 0$. Pak existují celá čísla q, r taková, že*

$$b = aq + r,\ 0 \leq r < a$$

Čísla q, r jsou určena jednoznačně.

Důkaz.

1. Existence: Využijeme rekurzi vzhledem k b.

- $0 \leq b$:
  $b < a$: $q = 0$, $r = b$
  $a \leq b$: Je $0 \leq b - a < b$ a existují $q', r' \in \mathbb{Z}$, $b - a = aq' + r'$, $0 \leq r' < a$. Položíme $q = q' + 1$, $r = r'$.
- $b < 0$:
  a/b: $q = \frac{b}{a}$, $r = 0$
  $\neg(a/b)$: Je $0 < -b$ a existují $q', r' \in \mathbb{Z}$, $-b = aq' + r'$, $0 \leq r' < a$. Položíme $q = -q' - 1$, $r = a - r'$.

2. Jednoznačnost:

   Nechť q, r, q', r' jsou celá čísla, $b = aq + r$, $0 \leq r < a$, $b = aq' + r'$, $0 \leq r' < a$. Chceme: $q = q'$, $r = r'$.

$$\begin{aligned} aq + r &= aq' + r' \\ r - r' &= aq' - aq \\ r - r' &= a(q' - q) \end{aligned}$$

   Vidíme, že $a/r - r'$. Dále,

$$\begin{array}{rcccl} 0 & \leq & r & < & a \\ 0 & \leq & r' & < & a \\ \hline 0 & \leq & r & < & a \\ -a & < & -r' & \leq & 0 \\ \hline -a & < & r - r' & < & a \end{array}$$

   Teď víme, že $a/r - r'$ a $-a < r - r' < a$. Z toho plyne, že $r - r' = 0$, $r = r'$. Dále, $aq = aq'$, $q = q'$.

2.3.5. Příklad. Pro $a = 3$, $b = 17$, je $q = 5$, $r = 2$ ($17 = 3 \cdot 5 + 2$, $0 \leq 2 < 3$).

Číslo q se nazývá **(neúplný) podíl** a číslo r se nazývá **zbytek** při dělení čísla b číslem a (označení: $r = b \bmod a$).

2.3.6. Poznámka. Nechť $a, b \in \mathbb{Z}, a > 0$. Pak

1. $a/b - b \bmod a$

2. $0 \leq b < a \iff b \bmod a = b$

2.3.7. Věta. *Nechť $a, b \in \mathbb{Z}$, $a \neq 0$ nebo $b \neq 0$. Pak existuje jediné $d \in \mathbb{Z}$, $d > 0$, takové, že*

1. $d/a \wedge d/b$
2. $(\forall c \in \mathbb{Z})\ c/a \wedge c/b \Longrightarrow c/d$

Důkaz.

1. Existence:

   Položme
   $$M = \{sa + tb |\ s \in \mathbb{Z} \wedge t \in \mathbb{Z} \wedge sa + tb > 0\}$$
   Ukážeme, že $M \neq \emptyset$. Rozlišíme tři případy:

   - $a > 0$: Je $1 \cdot a + 0 \cdot b = a > 0$, $a \in M$.
   - $a = 0$: Je $b \neq 0$. Jestliže $b > 0$, je $0 \cdot a + 1 \cdot b = b > 0$, $b \in M$. Jestliže $b < 0$, je $0 \cdot a + (-1) \cdot b = -b > 0$, $-b \in M$.
   - $a < 0$: Je $(-1) \cdot a + 0 \cdot b = -a > 0$, $-a \in M$.

   Je tedy $M \subseteq \mathbb{N}$, $M \neq 0$. Takže existuje $d = \min M$. Je $d \in M$, tudíž $d \in \mathbb{Z}$, $d > 0$. Dále $d = ua + vb$ pro nějaká $u, v \in \mathbb{Z}$.

   - d/a: Existují $q, r \in \mathbb{Z}$, $a = dq + r$, $0 \leq r < d$. Předpokládejme, že $r \neq 0$. Pak $r > 0$, $r = a - dq = a - (ua + vb)q = a - uaq - vbq = (1 - uq)a + (-vq)b$, $(1 - uq)a + (-vq)b = r > 0$. Jelikož $1 - uq, -vq \in \mathbb{Z}$, je $r \in M$. Pak $d \leq r$, spor. Nutně tedy $r = 0$, $a = dq$, d/a.
   - d/b: Postupujeme obdobně jako v důkazu tvrzení d/a.
   - Buď $c \in \mathbb{Z}$, c/a, c/b. Chceme: c/d. Stačí si uvědomit, že $d = ua + vb$.

2. Jednoznačnost:

   Nechť $d, e \in \mathbb{Z}$, $d > 0$, $e > 0$. Nechť

   (I) $d/a \wedge d/b$

(II) $(\forall c \in \mathbb{Z})\ c/a \wedge c/b \Longrightarrow c/d$

(III) $e/a \wedge e/b$

(IV) $(\forall c \in \mathbb{Z})\ c/a \wedge c/b \Longrightarrow c/e$

Chceme: $d = e$. Z (I) a (IV) plyne, že d/e. Z (III) a (II) plyne, že e/d. Celkem: $d > 0$, $e > 0$, d/e, e/d. Pak $d = e$ (viz cvičení).

2.3.8. Definice. Číslo d z věty 2.3.7 se nazývá **největší společný dělitel** čísel a a b; píšeme $d = GCD(a, b)$.

2.3.9. Důsledek (důkazu). *Pro všechna $a, b \in \mathbb{Z}$, $a \neq 0$ nebo $b \neq 0$, $GCD(a, b)$ je nejmenší kladná celočíselná lineární kombinace čísel a, b.*

2.3.10. Definice. Nechť a, b jsou celá čísla. Říkáme, že čísla a a b jsou **nesoudělná** (píšeme: $a \perp b$), pokud pro všechna kladná celá čísla d platí:

$$d/a \wedge d/b \Longrightarrow d = 1$$

2.3.11. Poznámka. Nechť a, b jsou celá čísla, $a \neq 0$ nebo $b \neq 0$.

1. $$a \perp b \Longleftrightarrow GCD(a, b) = 1$$

2. Nechť $a' = \frac{a}{GCD(a,b)}$, $b' = \frac{b}{GCD(a,b)}$. Pak $a' \perp b'$.
   Zdůvodnění: Existují celá čísla u, v tak, že $GCD(a, b) = ua + vb$ (viz 2.3.9). Pak $1 = ua' + vb'$ a tedy $GCD(a', b') = 1$, $a' \perp b'$.

2.3.12. Poznámka.

1. Nechť a je celé číslo, $a \neq 0$. Pak $GCD(a, 0) = |a|$.

2. Nechť a, b jsou celá čísla, $a \neq 0$ nebo $b \neq 0$. Pak $GCD(a, b) = GCD(|a|, |b|)$. Stačí tedy umět počítat největšího společného dělitele pro kladná celá čísla.

Euklidův algoritmus

VSTUP: $a, b \in \mathbb{Z}^+,\ a > b$

VÝSTUP: $GCD(a, b)$

$$(a_1, a_2) \longleftarrow (a, b)$$

KONEC: $a_2 = 0$

Je-li $a_2 \neq 0$, provedeme iteraci:

$$a_1 = qa_2 + r,\ q, r \in \mathbb{Z},\ 0 \leq r < a_2$$

$$(a_1, a_2) \mapsto (\overline{a}_1, \overline{a}_2),\ \overline{a}_1 = a_2,\ \overline{a}_2 = r$$

Jelikož $\overline{a}_2 = 0$ nebo $0 < \overline{a}_2 < a_2$, výpočet skončí po konečně mnoha krocích. Během výpočtu stále je $a_1 > a_2$.

Nechť $d \in \mathbb{Z}$. Platí:

$$d/a_1 \wedge d/a_2 \Longleftrightarrow d/a_2 \wedge d/r$$

Zdůvodnění:

1. Předpokládejme, že $d/a_1 \wedge d/a_2$. Chceme: $d/a_2 \wedge d/r$. Je jasné, že d/a_2. Je $r = a_1 + (-q)a_2$. Protože d/a_1 a d/a_2, máme d/r.

2. Předpokládejme, že $d/a_2 \wedge d/r$. Chceme: $d/a_1 \wedge d/a_2$. Je jasné, že d/a_2. Je $a_1 = qa_2 + r$. Protože d/a_2 a d/r, máme d/a_1.

Tudíž:

$$GCD(a_1, a_2) = GCD(a_2, r) = GCD(\overline{a}_1, \overline{a}_2)$$

Průběh výpočtu můžeme zapsat následovně:

$$(a, b) \mapsto \cdots \mapsto (\alpha, 0)$$

Je $GCD(a, b) = GCD(\alpha, 0) = |\alpha| = \alpha$ (je $\alpha > 0$),

$$\boxed{GCD(a, b) = \alpha}$$

2.3.13. Příklad. Pomocí Euklidova algoritmu vypočteme $GCD(172, 30)$.

$$(172, 30) \mapsto (30, 22) \mapsto (22, 8) \mapsto (8, 6) \mapsto (6, 2) \mapsto (2, 0)$$

Závěr: $GCD(172, 30) = 2$.

Rozšířený Euklidův algoritmus

VSTUP: $a, b \in \mathbb{Z}^+$, $a > b$

VÝSTUP: $GCD(a, b)$, $s, t \in \mathbb{Z}$, $GCD(a, b) = sa + tb$

$$(a_1, a_2) \longleftarrow (a, b),\ (p_1, p_2) \longleftarrow (0, 1),\ (q_1, q_2) \longleftarrow (1, 0)$$

KONEC: $a_2 = 0$

Je-li $a_2 \neq 0$, provedeme iteraci:

$$a_1 = qa_2 + r,\ q, r \in \mathbb{Z},\ 0 \leq r < a_2$$

$$\begin{array}{l} (a_1, a_2) \mapsto (\overline{a}_1, \overline{a}_2), \quad \overline{a}_1 = a_2, \quad \overline{a}_2 = r \\ (p_1, p_2) \mapsto (\overline{p}_1, \overline{p}_2), \quad \overline{p}_1 = p_2, \quad \overline{p}_2 = p_1 + qp_2 \\ (q_1, q_2) \mapsto (\overline{q}_1, \overline{q}_2), \quad \overline{q}_1 = q_2, \quad \overline{q}_2 = q_1 + qq_2 \end{array}$$

Víme již, že výpočet skončí po konečně mnoha krocích (viz Euklidův algoritmus).

$$\begin{aligned} \overline{p}_2\overline{a}_1 + \overline{p}_1\overline{a}_2 &= (p_1 + qp_2)a_2 + p_2 r \\ &= p_1 a_2 + qp_2 a_2 + p_2 r \\ &= p_1 a_2 + p_2(qa_2 + r) \\ &= p_1 a_2 + p_2 a_1 \\ &= p_2 a_1 + p_1 a_2 \end{aligned}$$

Ukázali jsme, že během výpočtu algoritmu stále platí

$$\overline{p}_2\overline{a}_1 + \overline{p}_1\overline{a}_2 = p_2 a_1 + p_1 a_2$$

$$\begin{aligned}
\overline{q}_2\overline{a}_1 + \overline{q}_1\overline{a}_2 &= (q_1 + qq_2)a_2 + q_2 r \\
&= q_1 a_2 + qq_2 a_2 + q_2 r \\
&= q_1 a_2 + q_2(qa_2 + r) \\
&= q_1 a_2 + q_2 a_1 \\
&= q_2 a_1 + q_1 a_2
\end{aligned}$$

Ukázali jsme, že během výpočtu algoritmu stále platí

$$\overline{q}_2\overline{a}_1 + \overline{q}_1\overline{a}_2 = q_2 a_1 + q_1 a_2$$

$$\begin{aligned}
\overline{q}_2\overline{p}_1 - \overline{q}_1\overline{p}_2 &= (q_1 + qq_2)p_2 - q_2(p_1 + qp_2) \\
&= q_1 p_2 + qq_2 p_2 - q_2 p_1 - q_2 q p_2 \\
&= q_1 p_2 - q_2 p_1 \\
&= -(q_2 p_1 - q_1 p_2)
\end{aligned}$$

Ukázali jsme, že během výpočtu algoritmu stále platí

$$\overline{q}_2\overline{p}_1 - \overline{q}_1\overline{p}_2 = -(q_2 p_1 - q_1 p_2)$$

Během výpočtu se tedy zachovávají následující invarianty:

1. $$\overline{p}_2\overline{a}_1 + \overline{p}_1\overline{a}_2 = p_2 a_1 + p_1 a_2$$
2. $$\overline{q}_2\overline{a}_1 + \overline{q}_1\overline{a}_2 = q_2 a_1 + q_1 a_2$$
3. $$\overline{q}_2\overline{p}_1 - \overline{q}_1\overline{p}_2 = -(q_2 p_1 - q_1 p_2)$$

Průběh výpočtu můžeme zapsat následovně:

$$\begin{array}{llllll}
(a,b) & \mapsto_1 & \ldots & \mapsto_n & (\alpha, 0) \\
(0,1) & \mapsto & \ldots & \mapsto & (\beta_1, \beta_2) \\
(1,0) & \mapsto & \ldots & \mapsto & (\gamma_1, \gamma_2)
\end{array}$$

Víme již, že

$$\boxed{GCD(a,b) = \alpha}$$

(viz Euklidův algoritmus).
Invariant 1 dává

$$\begin{aligned} \beta_2 \cdot \alpha + \beta_1 \cdot 0 &= 1 \cdot a + 0 \cdot b \\ \beta_2 \alpha &= a \end{aligned}$$

Invariant 2 dává

$$\begin{aligned} \gamma_2 \cdot \alpha + \gamma_1 \cdot 0 &= 0 \cdot a + 1 \cdot b \\ \gamma_2 \alpha &= b \end{aligned}$$

Invariant 3 dává

$$\begin{aligned} \gamma_2\beta_1 - \gamma_1\beta_2 &= (0 \cdot 0 - 1 \cdot 1) \cdot (-1)^n \\ \gamma_2\beta_1 - \gamma_1\beta_2 &= (-1)^{n+1} \\ \gamma_2\beta_1\alpha - \gamma_1\beta_2\alpha &= (-1)^{n+1} \cdot \alpha \\ \beta_1(\gamma_2\alpha) - \gamma_1(\beta_2\alpha) &= (-1)^{n+1} \cdot \alpha \\ \beta_1 b - \gamma_1 a &= (-1)^{n+1} \cdot \alpha \\ (-\gamma_1)a + \beta_1 b &= (-1)^{n+1} \cdot \alpha \\ (-1)^{n+1} \cdot (-\gamma_1)a + (-1)^{n+1} \cdot \beta_1 b &= (-1)^{n+1} \cdot (-1)^{n+1} \cdot \alpha \\ (-1)^n \cdot \gamma_1 a + (-1)^{n+1} \cdot \beta_1 b &= \alpha \\ ((-1)^n \cdot \gamma_1)a + ((-1)^{n+1} \cdot \beta_1)b &= GCD(a,b) \end{aligned}$$

Je tedy

$$\boxed{s = (-1)^n \cdot \gamma_1,\ t = (-1)^{n+1} \cdot \beta_1}$$

2.3.14. Příklad. Aplikujeme rozšířený Euklidův algoritmus na vstup $a = 172$, $b = 30$.

$$\begin{array}{ccccccccccc}
(172,30) & \mapsto_1 & (30,22) & \mapsto_2 & (22,8) & \mapsto_3 & (8,6) & \mapsto_4 & (6,2) & \mapsto_5 & (2,0) \\
(0,1) & \mapsto & (1,5) & \mapsto & (5,6) & \mapsto & (6,17) & \mapsto & (17,23) & \mapsto & (23,86) \\
(1,0) & \mapsto & (0,1) & \mapsto & (1,1) & \mapsto & (1,3) & \mapsto & (3,4) & \mapsto & (4,15)
\end{array}$$

Pomocné výpočty:

$$\begin{aligned}
172 &= 5 \cdot 30 + 22 \\
30 &= 1 \cdot 22 + 8 \\
22 &= 2 \cdot 8 + 6 \\
8 &= 1 \cdot 6 + 2 \\
6 &= 3 \cdot 2 + 0
\end{aligned}$$

Závěr: $GCD(172, 30) = 2$, $s = -4$, $t = 23$, $(-4) \cdot 172 + 23 \cdot 30 = 2$.

2.3.15. Věta. *Nechť $a, b, c \in \mathbb{Z}$.*

1. $a/bc \wedge a \perp b \Longrightarrow a/c$
2. $a/c \wedge b/c \wedge a \perp b \Longrightarrow ab/c$

Důkaz.

1. Předpoklad: $a/bc \wedge a \perp b$. Chceme: a/c. Jelikož $a \perp b$, je $a \neq 0$ nebo $b \neq 0$ a $GCD(a, b) = 1$. Pak existují $u, v \in \mathbb{Z}$ tak, že $ua + vb = 1$.

$$\begin{aligned}
ua + vb &= 1 \\
(ua + vb)c &= 1 \cdot c \\
uac + vbc &= c \\
(uc)a + v(bc) &= c
\end{aligned}$$

.

   Protože a/a a a/bc, máme a/c.

2. Předpoklad: $a/c \wedge b/c \wedge a \perp b$. Chceme: ab/c. Je $c = bu$ pro nějaké celé číslo u. Takže a/bu, $a \perp b$. Pak a/u (viz již dokázanou část 1). Je tedy $u = av$ pro nějaké celé číslo v. Pak $bu = bav$, $c = abv$, ab/c.

2.3.16. Definice. Nechť $a, b \in \mathbb{Z}$, $a > 0$, $b > 0$. Celé číslo m, $m > 0$, se nazývá **nejmenší společný násobek** čísel a, b, pokud

1. $a/m \wedge b/m$
2. $(\forall l \in \mathbb{Z})\ a/l \wedge b/l \Longrightarrow m/l$

2.3.17. Věta. *Nechť $a, b \in \mathbb{Z}$, $a > 0$, $b > 0$. Nejmenší společný násobek čísel a, b existuje a je určen jednoznačně.*

Důkaz.

- Existence: Nechť $d = GCD(a, b)$. Položme $m = \frac{ab}{d}$. Jelikož d/a, je $m \in \mathbb{Z}$. Jelikož $a > 0$, $b > 0$, $d > 0$, je $m > 0$. Ukážeme, že m je nejmenší společný násobek čísel a, b. Je třeba ukázat dvě věci:
  1. $a/m \wedge b/m$: Protože d/a, existuje $u \in \mathbb{Z}$, $a = du$. Pak $m = \frac{ab}{d} = \frac{dub}{d} = ub$, $m = ub$, b/m. Obdobně lze ukázat, že a/m.
  2. $(\forall l \in \mathbb{Z})\ a/l \wedge b/l \Longrightarrow m/l$: Předpoklad: $a/l \wedge b/l$. Chceme: m/l. Existují $u, v \in \mathbb{Z}$, $l = au$, $l = bv$. Jelikož d/a, existuje $a' \in \mathbb{Z}$, $a = da'$. Jelikož d/b, existuje $b' \in \mathbb{Z}$, $b = db'$. Pak $da'u = db'v$, $a'u = b'v$. Předpokládejme, že $a' \perp b'$. Protože $a'/b'v$, máme a'/v, $v = a'w$ pro nějaké $w \in \mathbb{Z}$. Je tedy $l = bv = ba'w = b\frac{a}{d}w = \frac{ab}{d}w = mw$, $l = mw$, m/l. Zbývá ukázat, že $a' \perp b'$. Existují celá čísla x, y taková, že $d = xa + yb$, $d = xda' + ydb'$, $1 = xa' + yb'$. Odtud již plyne, že $a' \perp b'$.
- Jednoznačnost: Nechť m, n jsou celá čísla, $m > 0$, $n > 0$. Předpokládejme, že
  1. $a/m \wedge b/m$
  2. $(\forall l \in \mathbb{Z})\ a/l \wedge b/l \Longrightarrow m/l$
  3. $a/n \wedge b/n$
  4. $(\forall l \in \mathbb{Z})\ a/l \wedge b/l \Longrightarrow n/l$

  Chceme: $m = n$. Z 1 a 4 dostáváme n/m, $n \leq m$. Z 3 a 2 dostáváme m/n, $m \leq n$. Celkem tedy $n \leq m$, $m \leq n$, $n = m$.

Jednoznačně určený nejmenší společný násobek kladných celých čísel a, b budeme označovat $LCM(a,b)$.

2.3.18. Důsledek (důkazu). *Nechť $a,b \in \mathbb{Z}$, $a>0$, $b>0$. Pak*

$$LCM(a,b) = \frac{ab}{GCD(a,b)},\ LCM(a,b)\cdot GCD(a,b) = ab$$

Cvičení.

1. Nechť $a,b,c,\beta,\gamma \in \mathbb{Z}$. Jestliže a/b a a/c, pak $a/\beta b + \gamma c$. Dokažte.
2. Nechť $d,e \in \mathbb{Z}$, $d>0$, $e>0$, d/e, e/d. Pak $d=e$. Dokažte.

2.4 Prvočísla

Aditivní struktura kladných celých čísel je poměrně jednoduchá. Každé kladné celé číslo n získáme z čísla 1 (vezmeme n kopií tohoto čísla) pomocí $n-1$ sčítání:

$$n = \underbrace{1+1+\cdots+1}_{n}$$

Multiplikativní struktura kladných celých čísel je složitější. Budeme se teď zabývat multiplikativními stavebními kameny kladných celých čísel – prvočísly.

2.4.1. Definice. Celé číslo p, $p>1$, je **prvočíslo**, pokud pro všechna kladná celá čísla d platí:

$$d/p \Longrightarrow d=1 \vee d=p$$

2.4.2. Poznámka.

1. Číslo 1 má také vlastnost, že pro všechna kladná celá čísla d platí implikace $d/1 \Longrightarrow d=1 \vee d=1$. Přesto však číslo 1 není prvočíslo (neplatí $1>1$).
2. Celé číslo s, $s>1$, které není prvočíslo, se nazývá **složené číslo**.

2.4.3. Tvrzení. *Nechť s je celé číslo, $s > 1$. Pak s je složené číslo právě tehdy, když $s = de$ pro nějaká celá čísla d, e, $1 < d < s$, $1 < e < s$.*

Důkaz.

1. Předpoklad: s je složené číslo. Chceme: existují $d, e \in \mathbb{Z}$, $s = de$, $1 < d < s$, $1 < e < s$. Protože s není prvočíslo, existuje $d \in \mathbb{Z}$, $d > 0$, d/s, $d \neq 1$, $d \neq s$. Jelikož $d > 0$, d/s, je $1 \leq d \leq s$. Celkem tedy $1 < d < s$. Existuje $e \in \mathbb{Z}$, $s = de$. Protože $d > 0$, $s > 0$, je $e > 0$. Jelikož e/s, je $1 \leq e \leq s$. Stačí ještě vyloučit případy $e = 1$ a $e = s$. Předpokládejme, že $e = 1$. Pak $d = s$, spor. Předpokládejme, že $e = s$. Pak $d = 1$, spor.

2. Předpoklad: existují $d, e \in \mathbb{Z}$, $s = de$, $1 < d < s$, $1 < e < s$. Chceme: s je složené číslo. Stačí si uvědomit, že $d > 0$, d/s, $d \neq 1$, $d \neq s$. Takže s není prvočíslo, s je složené číslo.

Nyní ukážeme, že prvočísla opravdu jsou multiplikativními stavebními kameny množiny všech kladných celých čísel.

2.4.4. Věta. *Nechť n je celé číslo, $n > 1$. Pak n je součinem jednoho nebo více prvočísel.*

Důkaz. Nechť M je množina všech celých čísel větších než jedna, která nelze zapsat ve tvaru součinu prvočísel. Chceme: $M = \emptyset$. Provedeme důkaz sporem. Předpokládejme tedy, že $M \neq \emptyset$. Je M neprázdná podmnožina množiny $\mathbb{N}$. Podle principu dobrého uspořádání má množina M nejmenší prvek; označme jej m. Je $m \in M$, takže m je celé číslo, $m > 1$, m nelze zapsat ve tvaru součinu prvočísel. Specielně tedy m není prvočíslo. Pak m je složené číslo, $m = de$, kde $d \in \mathbb{Z}$, $e \in \mathbb{Z}$, $1 < d < m$, $1 < e < m$ (viz tvrzení 2.4.3.). Protože $d < m$, $e < m$, máme $d \notin M$, $e \notin M$. Tedy čísla d, e jsou součinem několika prvočísel, $d = p_1 \cdots p_k$, $e = q_1 \cdots q_l$, $p_1, \ldots, p_k, q_1, \ldots, q_l$ jsou prvočísla. Pak

$$m = de = p_1 \cdots p_k q_1 \cdots q_l$$

Vidíme, že m lze zapsat ve tvaru součinu prvočísel. To je ve sporu s tím, že $m \in M$.

2.4.5. Poznámka. Připomeňme, že prázdný součin je roven číslu 1. Takže také číslo 1 je součinem několika prvočísel, a to 0 prvočísel.

Nyní dokážeme klasickou větu, kterou poprvé dokázal řecký matematik Eukleidés (325 př.n.l. – 260 př.n.l.).

2.4.6. Věta. (Eukleidés) *Existuje nekonečně mnoho prvočísel.*

Důkaz. Sporem. Předpokládejme, že existuje pouze k prvočísel, kde $k \in \mathbb{Z}$, $k \geq 1$ (aspoň jedno prvočíslo existuje, je jím například číslo 2). Tato prvočísla označme $p_1, \ldots, p_k$. Položme

$$n = p_1 p_2 \cdots p_k + 1$$

Číslo n je celé, $n > 1$. Je tedy n součinem několika prvočísel, jedno z nich označme q. Takže q/n. Protože q je prvočíslo a $p_1, \ldots, p_k$ jsou všechna prvočísla, je $q = p_i$ pro nějaké $i \in \{1, \ldots, k\}$. Pak $q/p_1 p_2 \ldots p_k$. Platí:

$$1 = n + (-1) \cdot p_1 \cdots p_k,\ q/n,\ q/p_1 p_2 \cdots p_k$$

Pak $q/1$, $q = 1$, spor.

K dispozici máme poměrně efektivní metodu k nalezení všech prvočísel až do jistého předem daného n. Autorem metody je Eratosthenés, který se narodil mezi lety 276 – 272 př.n.l. v Kyréně a zemřel roku 194 př.n.l. v Alexandrii.

Eratosthenovo síto

Vstup: $n \in \mathbb{Z}$, $n \geq 4$

Výstup: množina M všech prvočísel p takových, že $p \leq n$

$$M \longleftarrow \{2, 3, 4, \ldots, n\}$$

Pro $k = 2, 3, \ldots, \lfloor \sqrt{n} \rfloor$ dělej:

Jestliže $k \in M$, pak $M \leftarrow M - \{kl |\ l \in \mathbb{Z}, k \leq l\}$, jinak $M \leftarrow M$.

Konec.

Nechť P je množina všech prvočísel p takových, že $p \leq n$. Množinu M po skončení výpočtu označme M'. Chceme: $M' = P$.

Během výpočtu se množina M sice mění, avšak stále je $P \subseteq M$. Proč? Je tomu tak na počátku i po každé změně množiny M, protože číslo kl, kde $k \in \mathbb{Z}$, $l \in \mathbb{Z}$, $2 \leq k \leq l$, není prvočíslo.

Je tedy také $P \subseteq M'$. Chceme: $M' \subseteq P$.

Postupujme sporem. Nechť $m \in M'$, $m \notin P$. Je $m \in \{2, 3, 4, \ldots, n\}$, $m \notin P$, takže m není prvočíslo. Položme

$$D = \{d \in \mathbb{Z} |\ d > 0, d \neq 1, d \neq m, d/m\}$$

Jelikož m není prvočíslo, je $D \neq \emptyset$. Buď $k = \min D$. Existuje $l \in \mathbb{Z}$, $kl = m$. Protože $m > 0$, $k > 0$, je $l > 0$. Jistě l/m, $1 \leq l \leq m$. Předpokládejme, že $l = 1$. Pak $k = m$, spor. Nutně tedy $l \neq 1$. Předpokládejme, že $l = m$. Pak $k = 1$, spor. Nutně tedy $l \neq m$. Celkem: $l \in \mathbb{Z}$, $l > 0$, $l \neq 1$, $l \neq m$, l/m. Tudíž $l \in D$. Pak $k \leq l$. Máme:

$$2 \leq k \leq l \leq m \leq n$$

Pak $k^2 \leq kl$, $k^2 \leq m$, $k^2 \leq n$, $k \leq \sqrt{n}$, $k \leq \lfloor \sqrt{n} \rfloor$.

Ukážeme teď, že číslo k je prvočíslo. Předpokládejme naopak, že číslo k je složené. Pak existují celá čísla e, f, $1 < e < k$, $1 < f < k$, $k = ef$. Je $m = kl = efl$, takže f/m. Protože $f < k \leq m$, je $f \neq m$. Celkem: $f \in \mathbb{Z}$, $f > 0$, $f \neq 1$, $f \neq m$, f/m. Pak $f \in D$, $k \leq f$. To je spor. Opravdu tedy k je prvočíslo.

Protože $2 \leq k \leq n$, k je prvočíslo, je $k \in P$ a tedy stále během výpočtu je $k \in M$. V k-tém kroku výpočtu bylo z množiny M odstraněno číslo $kl = m$ (pokud nebylo již odstraněno dříve). Pak ovšem $m \notin M'$, spor.

2.4.7. Příklad. Buď $n = 80$. Pomocí Eratosthenova síta určíme všechna prvočísla p splňující $p \leq 80$.

Je $64 < 80 < 81$, takže $8 < \sqrt{80} < 9$, $\lfloor \sqrt{80} \rfloor = 8$.

$$\begin{aligned} M \ \leftarrow\ & \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, \\ & 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, \\ & 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, \\ & 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80\} \end{aligned}$$

$k = 2$:

$2 \in M$, takže $M \leftarrow M - \{2l |\ l \in \mathbb{Z}, 2 \leq l\}$, tj.

$$\begin{aligned} M \ \leftarrow\ & \{2, 3, 5, 7, 9, 11, 13, 15, 17, 19, \\ & 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, \\ & 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, \\ & 61, 63, 65, 67, 69, 71, 73, 75, 77, 79\} \end{aligned}$$

$k = 3$:
$3 \in M$, takže $M \leftarrow M - \{3l |\ l \in \mathbb{Z}, 3 \leq l\}$, tj.

$$\begin{aligned} M \quad \leftarrow \quad & \{2, 3, 5, 7, 11, 13, 17, 19, \\ & 23, 25, 29, 31, 35, 37, \\ & 41, 43, 47, 49, 53, 55, 59, \\ & 61, 65, 67, 71, 73, 77, 79\} \end{aligned}$$

$k = 4$:
$4 \notin M$, takže $M \leftarrow M$
$k = 5$:
$5 \in M$, takže $M \leftarrow M - \{5l |\ l \in \mathbb{Z}, 5 \leq l\}$, tj.

$$\begin{aligned} M \quad \leftarrow \quad & \{2, 3, 5, 7, 11, 13, 17, 19, \\ & 23, 29, 31, 37, \\ & 41, 43, 47, 49, 53, 59, \\ & 61, 67, 71, 73, 77, 79\} \end{aligned}$$

$k = 6$:
$6 \notin M$, takže $M \leftarrow M$
$k = 7$:
$7 \in M$, takže $M \leftarrow M - \{7l |\ l \in \mathbb{Z}, 7 \leq l\}$, tj.

$$\begin{aligned} M \quad \leftarrow \quad & \{2, 3, 5, 7, 11, 13, 17, 19, \\ & 23, 29, 31, 37, \\ & 41, 43, 47, 53, 59, \\ & 61, 67, 71, 73, 79\} \end{aligned}$$

$k = 8$:
$8 \notin M$, takže $M \leftarrow M$.
Závěr: Množina všech prvočísel p takových, že $p \leq 80$:

$$\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79\}$$

2.4.8. Věta. *Nechť p je prvočíslo, a, b jsou celá čísla. Pak*

$$p/ab \Longrightarrow p/a \vee p/b$$

Důkaz. Předpoklad: p/ab. Chceme: $p/a \vee p/b$. Rozlišíme dva případy:

1. p/a: Jsme hotovi.

2. $\neg(p/a)$: Ukážeme, že $p \perp a$. Nechť $d \in \mathbb{Z}$, $d > 0$, d/a, d/p. Chceme: $d = 1$. Protože p je prvočíslo, máme $d = 1$ nebo $d = p$. Stačí tedy vyloučit případ $d = p$. Předpokládejme naopak, že $d = p$. Víme, že d/a, takže p/a, spor. Nutně tedy $d \neq p$. Celkem: p/ab, $p \perp a$. Dle věty 2.3.15. p/b.

Víme, že existuje nekonečně mnoho prvočísel. První prvočíslo označíme p_1 (takže $p_1 = 2$), druhé prvočíslo označíme p_2 (takže $p_2 = 3$), třetí prvočíslo označíme p_3 (takže $p_3 = 5$), atd.

Již jsme viděli (viz větu 2.4.4.), že každé kladné celé číslo lze získat z prvočísel pomocí násobení. Dokážeme teď dokonce, že každé kladné celé číslo lze setavit z prvočísel pouze jediným způsobem.

2.4.9. Věta (Základní věta aritmetiky). *Nechť $p_1 < p_2 < p_3 < \ldots$ je soupis všech prvočísel. Každé kladné celé číslo n lze vyjádřit ve tvaru*

$$n = \prod_{1 \leq i} p_i^{a_i} = p_1^{a_1} p_2^{a_2} p_3^{a_3} \cdots$$

kde a_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0. Přitom čísla a_i jsou určena jednoznačně.

Důkaz. Existence vyjádření čísla n v uvedeném tvaru vyplývá z věty 2.4.4. Nyní dokážeme jednoznačnost. Nechť

$$n = \prod_{1 \leq i} p_i^{a_i} = \prod_{1 \leq i} p_i^{b_i}$$

kde a_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0, a také b_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0. Chceme: pro každé $i \in \mathbb{Z}$, $1 \leq i$, je $a_i = b_i$.

Existuje celé číslo k, $k > 1$, takové, že pro všechna celá čísla i, $k < i$, je $a_i = b_i = 0$. Máme tedy

$$n = \prod_{1 \leq i \leq k} p_i^{a_i} = \prod_{1 \leq i \leq k} p_i^{b_i}$$

Zbývá ukázat, že pro všechna celá čísla i, $1 \leq i \leq k$, je $a_i = b_i$. Postupujme sporem. Předpokládejme tedy, že existuje celé číslo s, $1 \leq s \leq k$, $a_s \neq b_s$. Nechť například $a_s < b_s$.

$$\begin{aligned} \prod_{1\leq i\leq k} p_i^{a_i} &= \prod_{1\leq i\leq k} p_i^{b_i} \\ p_s^{a_s} \cdot \prod_{1\leq i\leq k, i\neq s} p_i^{a_i} &= p_s^{b_s} \cdot \prod_{1\leq i\leq k, i\neq s} p_i^{b_i} \\ \prod_{1\leq i\leq k, i\neq s} p_i^{a_i} &= p_s^{b_s-a_s} \cdot \prod_{1\leq i\leq k, i\neq s} p_i^{b_i} \end{aligned}$$

Jelikož $b_s - a_s > 0$, máme

$$p_s / \prod_{1\leq i\leq k, i\neq s} p_i^{a_i}$$

Z věty 2.4.8. vyplývá, že $p_s / p_j^{a_j}$ pro nějaké $j \in \mathbb{Z}$, $1 \leq j \leq k$, $j \neq s$. Jsou dvě možnosti, obě dají spor. Tím bude věta dokázána.

- $a_j = 0$: Máme $p_s/1$, $p_s = 1$, spor (p_s je totiž prvočíslo).
- $a_j > 0$: Z věty 2.4.8. vyplývá, že p_s / p_j. Protože p_j je prvočíslo, je $p_s = 1$ nebo $p_s = p_j$. Případ $p_s = 1$ dává spor, protože p_s je prvočíslo. Případ $p_s = p_j$ dává spor, protože $s \neq j$.

2.4.10. Příklad. Je $2022 = 2 \cdot 3 \cdot 337$, $2 = p_1$, $3 = p_2$, $337 = p_{68}$, takže pro $n = 2022$ máme $a_1 = 1$, $a_2 = 1$, $a_{68} = 1$, $a_i = 0$ pro $i \in \mathbb{Z}$, $1 \leq i$, $i \notin \{1, 2, 68\}$.

Na závěr této části dokážeme známé vzorce pro výpočet $GCD(m, n)$ a $LCM(m, n)$, které jsme se kdysi naučili ve škole. Jejich praktické používání je však zásadně omezeno tím, že musíme znát prvočíselné rozklady čísel m a n.

2.4.11. Věta. *Nechť $p_1 < p_2 < p_3 < \ldots$ je soupis všech prvočísel. Nechť m, n jsou kladná celá čísla,*

$$m = \prod_{1\leq i} p_i^{a_i}, \; n = \prod_{1\leq i} p_i^{b_i}$$

kde a_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0, a také b_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0. Potom

1. $$m/n \iff a_i \leq b_i \text{ pro } i = 1, 2, 3, \ldots$$

2. $$GCD(m, n) = \prod_{1 \leq i} p_i^{\min\{a_i, b_i\}}$$

3. $$LCM(m, n) = \prod_{1 \leq i} p_i^{\max\{a_i, b_i\}}$$

Důkaz.

1. - Předpoklad: m/n. Chceme: $a_i \leq b_i$ pro $i = 1, 2, 3, \ldots$. Existuje kladné celé číslo s, $n = ms$. Je $s = \prod_{1 \leq i} p_i^{t_i}$, kde t_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0. Máme
$$\begin{aligned} \prod_{1 \leq i} p_i^{b_i} &= \prod_{1 \leq i} p_i^{a_i} \cdot \prod_{1 \leq i} p_i^{t_i} \\ \prod_{1 \leq i} p_i^{b_i} &= \prod_{1 \leq i} p_i^{a_i + t_i} \end{aligned}$$
Zvolme libovolné celé číslo i, $1 \leq i$. Je $a_i + t_i = b_i$. Ovšem $0 \leq t_i$, $a_i \leq a_i + t_i$, $a_i \leq b_i$.
   - Předpoklad: $a_i \leq b_i$ pro $i = 1, 2, 3, \ldots$. Chceme: m/n. Uvědomme si, že $b_i - a_i$, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0. Položme $s = \prod_{1 \leq i} p_i^{b_i - a_i}$. Zřejmě s je kladné celé číslo. Máme
$$\begin{aligned} \prod_{1 \leq i} p_i^{b_i} &= \prod_{1 \leq i} p_i^{a_i + (b_i - a_i)} \\ \prod_{1 \leq i} p_i^{b_i} &= \prod_{1 \leq i} p_i^{a_i} \cdot \prod_{1 \leq i} p_i^{b_i - a_i} \\ n &= m \cdot s \end{aligned}$$

Vidíme, že m/n.

2. Položme
$$d = \prod_{1 \leq i} p_i^{\min\{a_i, b_i\}}$$
Ukážeme, že $d = GCD(m, n)$. Jistě d je kladné celé číslo. Zbývá ukázat dvě věci:

   - $d/m \wedge d/n$: Pro každé kladné celé číslo i je $\min\{a_i, b_i\} \leq a_i$ a také $\min\{a_i, b_i\} \leq b_i$, takže dle již dokázané části 1 máme d/m a d/n.
   - $(\forall c \in \mathbb{Z})\ c/m \wedge c/n \implies c/d$: Předpokládejme, že c je celé číslo, c/m, c/n. Chceme: c/d. Jelikož $m \neq 0$, je $c \neq 0$. Pak $|c| > 0$,
$$|c| = \prod_{1 \leq i} p_i^{t_i}$$
pro nějaká nezáporná celá čísla t_i, $i = 1, 2, 3, \ldots$, z nichž pouze konečně mnoho je větších než 0. Protože $|c|/c$, máme $|c|/m$ a $|c|/n$. Dle již dokázané části 1 platí: $t_i \leq a_i$ a $t_i \leq b_i$ pro $i = 1, 2, 3, \ldots$. Pak ovšem $t_i \leq \min\{a_i, b_i\}$ pro $i = 1, 2, 3, \ldots$ a tedy $|c|/d$. Jelikož $c/|c|$, dostáváme c/d, což jsme chtěli.

3. Položme
$$e = \prod_{1 \leq i} p_i^{\max\{a_i, b_i\}}$$
Ukážeme, že $e = LCM(m, n)$. Jistě e je kladné celé číslo. Zbývá ukázat dvě věci:

   - $m/e \wedge n/e$: Pro každé kladné celé číslo i je $a_i \leq \max\{a_i, b_i\}$ a také $b_i \leq \max\{a_i, b_i\}$, takže dle již dokázané části 1 máme m/e a n/e.
   - $(\forall l \in \mathbb{Z})\ m/l \wedge n/l \implies e/l$: Předpokládejme, že l je celé číslo, m/l, n/l. Chceme: e/l. Pokud $l = 0$, jsme hotovi (každé celé číslo dělí nulu). Nechť tedy $l \neq 0$. Pak $|l| > 0$,
$$|l| = \prod_{1 \leq i} p_i^{t_i}$$
pro nějaká nezáporná celá čísla t_i, $i = 1, 2, 3, \ldots$, z nichž pouze konečně mnoho je větších než 0. Protože $l/|l|$, máme $m/|l|$ a $n/|l|$.

Dle již dokázané části 1 platí: $a_i \leq t_i$ a $b_i \leq t_i$ pro $i = 1, 2, 3, \ldots$. Pak ovšem $\max\{a_i, b_i\} \leq t_i$ pro $i = 1, 2, 3, \ldots$ a tedy $e/|l|$. Jelikož $|l|/l$, dostáváme e/l, což jsme chtěli.

2.4.12. Poznámka. Nechť $p_1 < p_2 < p_3 < \ldots$ je soupis všech prvočísel. Nechť m, n jsou kladná celá čísla,

$$m = \prod_{1 \leq i} p_i^{a_i}, \ n = \prod_{1 \leq i} p_i^{b_i}$$

kde a_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0, a také b_i, $i = 1, 2, 3, \ldots$, jsou nezáporná celá čísla, z nichž pouze konečně mnoho je větších než 0. Víme již dlouho, že $LCM(m, n) \cdot GCD(m, n) = mn$ (viz 2.3.18). Tuto rovnost lze s využitím věty 2.4.11 dokázat velmi snadno:

$$\begin{aligned} LCM(m,n) \cdot GCD(m,n) &= \prod_{1 \leq i} p_i^{\max\{a_i,b_i\}} \cdot \prod_{1 \leq i} p_i^{\min\{a_i,b_i\}} \\ &= \prod_{1 \leq i} p_i^{\max\{a_i,b_i\}+\min\{a_i,b_i\}} \\ &= \prod_{1 \leq i} p_i^{a_i+b_i} \\ &= \prod_{1 \leq i} p_i^{a_i} \cdot \prod_{1 \leq i} p_i^{b_i} \\ &= mn \end{aligned}$$

Při výpočtu jsme použili fakt, že $\max\{k, l\} + \min\{k, l\} = k + l$ pro libovolná dvě celá čísla k, l.

2.4.13. Příklad. Nechť $m = 113366$, $n = 224455$. Pak

$$m = 2 \cdot 11 \cdot 5153, \ n = 5 \cdot 7 \cdot 11^2 \cdot 53$$

$$m = 2^1 \cdot 5^0 \cdot 7^0 \cdot 11^1 \cdot 53^0 \cdot 5153^1, \ n = 2^0 \cdot 5^1 \cdot 7^1 \cdot 11^2 \cdot 53^1 \cdot 5153^0$$

a tedy

$$GCD(m, n) = 2^0 \cdot 5^0 \cdot 7^0 \cdot 11^1 \cdot 53^0 \cdot 5153^0 = 11$$

$$LCM(m, n) = 2^1 \cdot 5^1 \cdot 7^1 \cdot 11^2 \cdot 53^1 \cdot 5153^1 = 2313233230$$

Cvičení.

1. Pomocí Eratosthenova síta najděte všechna prvočísla p taková, že $p \leq 500$. Nalezená prvočísla zapište dle velikosti, od nejmenšího po největší.

2. Číslo 1111111111111111111 je prvočíslo. Dokažte, že při libovolném základu b poziční číselné soustavy číslo $(11\ldots1)_b$ může být prvočíslem pouze tehdy, pokud obsahuje prvočíselný počet jedniček.

2.5 Modulární aritmetika

V této části se budeme zabývat "ciferníkovou aritmetikou". Co to je? Ciferníková aritmetika se od klasické liší v tom, že používá jenom omezený rozsah celých čísel, která můžeme uspořádat do kruhu podobně jako na ciferníku hodin od 1 do 12. Platí zde jiný způsob počítání. Například $7 + 8 = 3$. Výsledek získáme tak, že začneme u sedmičky, na hodinách se posuneme o osm míst a skončíme u trojky. Podobně můžeme odčítat i násobit. Jedná se o aritmetiku na konečných množinách. Stane se například základem pro naše (krátké) studium kryptografie v části 2.8.

2.5.1. Definice. Nechť $m \in \mathbb{Z}$, $m > 0$ a $a, b \in \mathbb{Z}$. Říkáme, že a je **kongruentní s** b **modulo** m, pokud $m/a - b$, a píšeme $a \equiv b \pmod{m}$ či pouze $a \equiv b$, je-li m známo z kontextu.

2.5.2. Tvrzení. *Nechť $m \in \mathbb{Z}$, $m > 0$ a $a, b \in \mathbb{Z}$. Pak*

$$a \equiv b \pmod{m} \iff a \bmod m = b \bmod m$$

Důkaz.

1. Předpoklad: $a \equiv b \pmod{m}$. Chceme: $a \bmod m = b \bmod m$. Nechť $a = mq_1 + r_1$, kde $q_1, r_1 \in \mathbb{Z}$, $0 \leq r_1 < m$. Dále, nechť $b = mq_2 + r_2$, kde $q_2, r_2 \in \mathbb{Z}$, $0 \leq r_2 < m$. Pak $a - b = (mq_1 + r_1) - (mq_2 + r_2) = m(q_1 - q_2) + (r_1 - r_2)$. Víme, že $m/a - b$. Takže $a - b = mk$ pro nějaké celé číslo k. Je tedy $m(q_1 - q_2) + (r_1 - r_2) = mk$, $r_1 - r_2 = mk - m(q_1 - q_2)$. Vidíme, že $m/r_1 - r_2$. Ovšem

$$\begin{array}{ccccc} 0 & \leq & r_1 & < & m \\ 0 & \leq & r_2 & < & m \\ \hline 0 & \leq & r_1 & < & m \\ -m & < & -r_2 & \leq & 0 \\ \hline -m & < & r_1 - r_2 & < & m \end{array}$$

Ukázali jsme, že $m/r_1 - r_2$ a $-m < r_1 - r_2 < m$. Z toho plyne, že $r_1 - r_2 = 0$, $r_1 = r_2$. Nyní si stačí uvědomit, že $a \bmod m = r_1$ a $b \bmod m = r_2$.

2. Předpoklad: $a \bmod m = b \bmod m$. Chceme: $a \equiv b \pmod m$. Nechť $a = mq_1 + r_1$, kde $q_1, r_1 \in \mathbb{Z}$, $0 \leq r_1 < m$. Dále, nechť $b = mq_2 + r_2$, kde $q_2, r_2 \in \mathbb{Z}$, $0 \leq r_2 < m$. Pak $a - b = (mq_1 + r_1) - (mq_2 + r_2) = m(q_1 - q_2) + (r_1 - r_2)$. Protože $a \bmod m = r_1$ a $b \bmod m = r_2$, je $r_1 = r_2$, $r_1 - r_2 = 0$, $a - b = m(q_1 - q_2)$, $m/a - b$, $a \equiv b \pmod m$.

Relace $\equiv$ se chová podobně, jako relace $=$.

2.5.3. Věta. *Relace $\equiv$ je ekvivalence na množině $\mathbb{Z}$.*

Důkaz.

1. Relace $\equiv$ je reflexivní: Nechť $a \in \mathbb{Z}$. Chceme: $a \equiv a \pmod m$. Chceme tedy: $m/a - a$, $m/0$. To jistě platí.

2. Relace $\equiv$ je symetrická: Nechť $a, b \in \mathbb{Z}$, $a \equiv b \pmod m$. Chceme: $b \equiv a \pmod m$. Víme, že $m/a - b$. Proto $a - b = km$ pro nějaké celé číslo k. Pak $-(a - b) = -km$, $b - a = (-k)m$, $m/b - a$, $b \equiv a \pmod m$.

3. Relace $\equiv$ je tranzitivní. Nechť $a, b, c \in \mathbb{Z}$, $a \equiv b \pmod m$, $b \equiv c \pmod m$. Chceme: $a \equiv c \pmod m$. Víme, že $m/a - b$, $m/b - c$. Pak $m/(a - b) + (b - c)$, $m/a - c$, $a \equiv c \pmod m$.

Podobnost relace $\equiv$ s relací $=$ jde ještě dále – rovnosti můžeme sčítat a násobit a také kongruence můžeme sčítat a násobit. Dále, v rovnosti můžeme krátit (nikoli však každým číslem – číslo, kterým krátíme, musí být nenulové) a také v kongruenci můžeme krátit (nikoli však každým číslem – číslo, kterým krátíme, musí být nesoudělné s modulem). To vše vyslovíme a dokážeme v následující větě.

2.5.4. Věta. *Nechť $m \in \mathbb{Z}$, $m > 0$, $a, b, c, d \in \mathbb{Z}$. Platí:*

1. $a \equiv b \pmod m \wedge c \equiv d \pmod m \Longrightarrow a + c \equiv b + d \pmod m$

2. $a \equiv b \pmod m \wedge c \equiv d \pmod m \Longrightarrow ac \equiv bd \pmod m$

3. $ca \equiv cb \pmod{m} \wedge c \perp m \Longrightarrow a \equiv b \pmod{m}$

Důkaz.

1. Předpoklad: $a \equiv b \pmod{m}$, $c \equiv d \pmod{m}$. Chceme: $a + c \equiv b + d \pmod{m}$. Víme: $m/a - b$, $m/c - d$. Pak $m/(a - b) + (c - d)$, $m/(a + c) - (b + d)$, $a + c \equiv b + d \pmod{m}$.

2. Předpoklad: $a \equiv b \pmod{m}$, $c \equiv d \pmod{m}$. Chceme: $ac \equiv bd \pmod{m}$. Víme: $m/a-b$, $m/c-d$. Je $ac-bd = ac-bc+bc-bd = (a-b)c+b(c-d)$. Jelikož $m/a - b$ a $m/c - d$, dostáváme $m/ac - bd$, $ac \equiv bd \pmod{m}$.

3. Předpoklad: $ca \equiv cb \pmod{m}$ a $c \perp m$. Chceme: $a \equiv b \pmod{m}$. Víme: $m/ca - cb$, $m/c(a - b)$. Jelikož $m \perp c$, dostáváme $m/a - b$, $a \equiv b \pmod{m}$.

2.5.5. Věta. *Nechť* $m, n \in \mathbb{Z}$, $m > 0$, $n > 0$, $m \perp n$, $a, b \in \mathbb{Z}$. *Pak*

$$a \equiv b \pmod{mn} \Longleftrightarrow a \equiv b \pmod{m} \wedge a \equiv b \pmod{n}$$

Důkaz.

1. Předpoklad: $a \equiv b \pmod{mn}$. Chceme: $a \equiv b \pmod{m} \wedge a \equiv b \pmod{n}$. Víme: $mn/a-b$. Pak $m/a-b$ a $n/a-b$, takže $a \equiv b \pmod{m}$ a $a \equiv b \pmod{n}$.

2. Předpoklad: $a \equiv b \pmod{m} \wedge a \equiv b \pmod{n}$. Chceme: $a \equiv b \pmod{mn}$. Víme: $m/a - b$, takže $a - b = mk$ pro nějaké celé číslo k. Víme také, že $n/a - b$. Dostáváme: n/mk. Jelikož $m \perp n$, máme n/k, $k = nl$ pro nějaké celé číslo l. Pak $a - b = mk = mnl$, $mn/a - b$, $a \equiv b \pmod{mn}$.

Následující věta má velmi jednoduchý důkaz – udělejte jej jako cvičení.

2.5.6. Věta. *Nechť* $m, n \in \mathbb{Z}$, $m > 0$, $n > 0$, $a, b \in \mathbb{Z}$. *Pak*

$$an \equiv bn \pmod{mn} \Longleftrightarrow a \equiv b \pmod{m}$$

Na závěr této části ještě zformulujeme jedno jednoduché a často používané tvrzení.

2.5.7. Tvrzení. *Nechť $m, a \in \mathbb{Z}$, $m > 0$. Pak*

$$a \equiv a \bmod m \pmod m$$

Důkaz. Stačí se odvolat na první část poznámky 2.3.6.

Cvičení.

1. Dokažte větu 2.5.6.

2.6 Řešení lineárních kongruencí

V této části se budeme zabývat řešením lineárních kongruencí $ax \equiv b \pmod m$. Vyslovíme nejprve kritérium řešitelnosti lineárních kongruencí.

2.6.1. Věta. *Nechť $m \in \mathbb{Z}$, $m > 0$, $a, b \in \mathbb{Z}$. Pak lineární kongruence $ax \equiv b \pmod m$ má řešení právě tehdy, když $GCD(a, m)/b$.*

Důkaz. Pro stručnost položme $d = GCD(a, m)$.

1. Předpoklad: Kongruence $ax \equiv b \pmod m$ má řešení. Chceme: d/b. Existuje $u \in \mathbb{Z}$, $au \equiv b \pmod m$. Pak $m/au - b$, $au - b = mk$ pro nějaké celé číslo k. Je $b = au - mk$. Protože d/a a d/m, máme d/b.

2. Předpoklad: d/b. Chceme: Kongruence $ax \equiv b \pmod m$ má řešení. Jelikož d/b, je $b = kd$ pro nějaké celé číslo k. Existují celá čísla s, t taková, že $sa+tm = d$ (viz 2.3.9). Pak $k(sa+tm) = kd$, $ksa+ktm = b$, $a(ks) - b = (-kt)m$, $m/a(ks) - b$, $a(ks) \equiv b \pmod m$, Vidíme, že celé číslo ks je řešením kongruence $ax \equiv b \pmod m$.

2.6.2. Poznámka. Druhá část důkazu věty 2.6.1 dává návod, jak najít jedno řešeni x_0 kongruence $ax \equiv b \pmod m$ v případě, kdy d/b (je $d =$

$GCD(a,m)$) – najdeme celá čísla s,t taková, že $sa+tm=d$ (například rozšířeným Euklidovým algoritmem), a položíme $x_0=\frac{b}{d}s$.

2.6.3. Věta. *Nechť $m\in\mathbb{Z}$, $m>0$, $a,b\in\mathbb{Z}$.Nechť $d=GCD(a,m)$. Předpokládejme, že d/b. Pak lineární kongruence $ax\equiv b \pmod m$ má přesně d řešení modulo m, a to*

$$x_0+k\frac{m}{d},\ k=0,1,\ldots,d-1$$

kde $x_0\in\mathbb{Z}$ je libovolné řešení kongruence $ax\equiv b \pmod m$.

Důkaz. Je třeba dokázat tři věci.

1. $a(x_0+k\frac{m}{d})\equiv b \pmod m$ pro $k=0,1,\ldots,d-1$:
   Je $a(x_0+k\frac{m}{d})=ax_0+\frac{a}{d}km\equiv ax_0\equiv b \pmod m$, $a(x_0+k\frac{m}{d})\equiv b \pmod m$.

2. $\neg(x_0+k\frac{m}{d}\equiv x_0+l\frac{m}{d} \pmod m)$ pro $k,l\in\mathbb{Z}$, $0\le k<d$, $0\le l<d$, $k\neq l$:
   Postupujme sporem. Předpokládejme, že $k,l\in\mathbb{Z}, 0\le k<d, 0\le l<d$, $k\neq l$, $x_0+k\frac{m}{d}\equiv x_0+l\frac{m}{d} \pmod m$. Pak $m/(x_0+k\frac{m}{d})-(x_0+l\frac{m}{d})$, $m/k\frac{m}{d}-l\frac{m}{d}$, $m/(k-l)\frac{m}{d}$, $(k-l)\frac{m}{d}=qm$ pro nějaké celé číslo q. Pak $(k-l)m=qmd$, $k-l=qd$, $d/k-l$. Dále

   $$\begin{array}{ccccc} 0 & \le & k & < & d \\ 0 & \le & l & < & d \\ \hline 0 & \le & k & < & d \\ -d & < & -l & \le & 0 \\ \hline -d & < & k-l & < & d \end{array}$$

   Celkem $d/k-l$, $-d<k-l<d$ a tedy $k-l=0$, $k=l$, spor.

3. $au\equiv b \pmod m$ $(u\in\mathbb{Z})$ dává $u\equiv x_0+k\frac{m}{d} \pmod m$ pro nějaké $k\in\mathbb{Z}$, $0\le k<d$:
   Předpokládejme, že $u\in\mathbb{Z}$, $au\equiv b \pmod m$. Je $ax_0\equiv b \pmod m$. Tudíž $au\equiv ax_0 \pmod m$, $m/au-ax_0$, $m/a(u-x_0)$, $a(u-x_0)=lm$ pro nějaké celé číslo l. Pak $\frac{a}{d}(u-x_0)=l\frac{m}{d}$, $\frac{m}{d}/\frac{a}{d}(u-x_0)$. Je $\frac{a}{d}\perp\frac{m}{d}$ (viz 2.3.11). Nyní víme toto: $\frac{a}{d}\perp\frac{m}{d}$, $\frac{m}{d}/\frac{a}{d}(u-x_0)$. Pak ovšem $\frac{m}{d}/u-x_0$, $u-x_0=s\frac{m}{d}$ pro nějaké celé číslo s. Provedeme dělení se zbytkem: $s=qd+k$, $q,k\in\mathbb{Z}$, $0\le k<d$. Dosazením za s dostaneme $u-x_0=(qd+k)\frac{m}{d}=qm+k\frac{m}{d}$, $u-x_0-k\frac{m}{d}=qm$, $m/u-(x_0+k\frac{m}{d})$, $u\equiv x_0+k\frac{m}{d} \pmod m$; připomeňme ještě, že $k\in\mathbb{Z}$, $0\le k<d$.

Řešení lineární kongruence $ax \equiv b \pmod{m}$ nyní ukážeme na dvou konkrétních příkladech.

2.6.4. Příklad. Vyřešíme kongruenci $27x \equiv 1 \pmod{47}$. Je $a = 27$, $b = 1$, $m = 47$. Rozšířeným Euklidovým algoritmem vypočteme $d = GCD(27, 47)$.

$$\begin{array}{ccccccccccc} (47,27) & \mapsto_1 & (27,20) & \mapsto_2 & (20,7) & \mapsto_3 & (7,6) & \mapsto_4 & (6,1) & \mapsto_5 & (1,0) \\ (0,1) & \mapsto & (1,1) & \mapsto & (1,2) & \mapsto & (2,5) & \mapsto & (5,7) & \mapsto & (7,47) \\ (1,0) & \mapsto & (0,1) & \mapsto & (1,1) & \mapsto & (1,3) & \mapsto & (3,4) & \mapsto & (4,27) \end{array}$$

Pomocné výpočty:

$$\begin{aligned} 47 &= 1 \cdot 27 + 20 \\ 27 &= 1 \cdot 20 + 7 \\ 20 &= 2 \cdot 7 + 6 \\ 7 &= 1 \cdot 6 + 1 \\ 6 &= 6 \cdot 1 + 0 \end{aligned}$$

Takže: $d = GCD(27, 47) = 1$, $(-4) \cdot 47 + 7 \cdot 27 = 1$. Vidíme, že $x_0 = 7$ řeší kongruenci $27x \equiv 1 \pmod{47}$. Dle věty 2.6.3 má kongrunce $27x \equiv 1 \pmod{47}$ právě jedno řešení modulo 47, a to 7.

2.6.5. Příklad. Vyřešíme kongruenci $51x \equiv 9 \pmod{69}$. Je $a = 51$, $b = 9$, $m = 69$. Rozšířeným Euklidovým algoritmem vypočteme $d = GCD(51, 69)$.

$$\begin{array}{ccccccccc} (69,51) & \mapsto_1 & (51,18) & \mapsto_2 & (18,15) & \mapsto_3 & (15,3) & \mapsto_4 & (3,0) \\ (0,1) & \mapsto & (1,1) & \mapsto & (1,3) & \mapsto & (3,4) & \mapsto & (4,23) \\ (1,0) & \mapsto & (0,1) & \mapsto & (1,2) & \mapsto & (2,3) & \mapsto & (3,17) \end{array}$$

Pomocné výpočty:

$$\begin{aligned} 69 &= 1 \cdot 51 + 18 \\ 51 &= 2 \cdot 18 + 15 \\ 18 &= 1 \cdot 15 + 3 \\ 15 &= 5 \cdot 3 + 0 \end{aligned}$$

Takže: $d = GCD(51, 69) = 3$, $3 \cdot 69 + (-4) \cdot 51 = 3$, $9 \cdot 69 + (-12) \cdot 51 = 9$. Vidíme, že $x_0 = -12$ řeší kongruenci $51x \equiv 9 \pmod{69}$. Je $\frac{m}{d} = \frac{69}{3} = 23$. Dle věty 2.6.3 kongrunce $51x \equiv 9 \pmod{69}$ má přesně tři řešení modulo 69, a to

$$-12 + 0 \cdot 23,\ -12 + 1 \cdot 23,\ -12 + 2 \cdot 23$$

tj.

$$-12, 11, 34$$

Dle věty 2.6.3 je x_0 libovolné řešení kongruence $51x \equiv 9 \pmod{69}$, můžeme tedy vzít třeba $x_0 = 11$. Pak dostaneme, že kongruence $51x \equiv 9 \pmod{69}$ má přesně tři řešení modulo 69, a to

$$11, 34, 57$$

Uvědomme si pořádně, že uvedené vyjádření "kongruence $51x \equiv 9 \pmod{69}$ má přesně tři řešení modulo 69, a to $11, 34, 57$"znamená tři věci:

- $51 \cdot 11 \equiv 9 \pmod{69} \ \wedge\ 51 \cdot 34 \equiv 9 \pmod{69} \ \wedge\ 51 \cdot 57 \equiv 9 \pmod{69}$
- $\neg(11 \equiv 34 \pmod{69}) \wedge \neg(11 \equiv 57 \pmod{69}) \wedge \neg(34 \equiv 57 \pmod{69})$
- $(\forall u \in \mathbb{Z})\ 51u \equiv 9 \pmod{69} \implies u \equiv 11 \pmod{69} \ \vee\ u \equiv 34 \pmod{69} \ \vee\ u \equiv 57 \pmod{69}$

Samozřejmě, obdobný význam má vyjádření "kongruence $51x \equiv 9 \pmod{69}$ má přesně tři řešení modulo 69, a to $-12, 11, 34$".

Na závěr této části se ještě budeme věnovat speciálním systémům lineárních kongruencí.

2.6.6. Věta (Čínská věta o zbytcích). *Nechť $m_i \in \mathbb{Z}$, $m_i > 0$ pro $i \in \{1, 2, \ldots, k\}$. Nechť $m_i \perp m_j$ pro $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$. Nechť $a_i \in \mathbb{Z}$ pro $i \in \{1, 2, \ldots, k\}$. Položme $M = m_1 m_2 \cdots m_k$. Pak systém kongruencí*

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

má právě jedno řešení modulo M.

Důkaz.

1. Existence: Pro $i \in \{1, 2, \ldots, k\}$ položme $M_i = m_1 \cdots m_{i-1} m_{i+1} \cdots m_k$. Buď d kladné celé číslo, d/m_i, d/M_i. Předpokládejme, že $d > 1$. Pak existuje prvočíslo p, p/d. Pak p/m_i a p/M_i. Jelikož p je prvočíslo, $p/m_1 \cdots m_{i-1} m_{i+1} \cdots m_k$, existuje $j \in \{1, 2, \ldots, k\}$, $j \neq i$, p/m_j. Zjistili jsme, že p/m_i a p/m_j, kde $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$. To je spor, neboť $m_i \perp m_j$ pro $i \neq j$. Nutně tedy $d = 1$ a $m_i \perp M_i$. Pak $1 = sM_i + tm_i$ pro nějaká $s, t \in \mathbb{Z}$. Položme $M_i' = s$; je $M_i' M_i \equiv 1 \pmod{m_i}$. Ukážeme, že $x_0 = a_1 M_1' M_1 + a_2 M_2' M_2 + \cdots + a_k M_k' M_k$ je řešení zadaného systému kongruencí. Zvolme libovolně $i \in \{1, 2, \ldots, k\}$. Chceme: $x_0 \equiv a_i \pmod{m_i}$. Je $M_i' M_i \equiv 1 \pmod{m_i}$ a tedy $a_i M_i' M_i \equiv a_i \pmod{m_i}$. Pro $j \in \{1, 2, \ldots, k\}$, $j \neq i$, máme m_i / M_j, $M_j \equiv 0 \pmod{m_i}$, $a_j M_j' M_j \equiv 0 \pmod{m_i}$. Celkem

$$\begin{aligned}
a_1 M_1' M_1 &\equiv 0 \pmod{m_i} \\
&\vdots \\
a_{i-1} M_{i-1}' M_{i-1} &\equiv 0 \pmod{m_i} \\
a_i M_i' M_i &\equiv a_i \pmod{m_i} \\
a_{i+1} M_{i+1}' M_{i+1} &\equiv 0 \pmod{m_i} \\
&\vdots \\
a_k M_k' M_k &\equiv 0 \pmod{m_k}
\end{aligned}$$

   Sečtením kongruencí dostaneme $x_0 \equiv a_i \pmod{m_i}$. To jsme chtěli.

2. Jednoznačnost: Nechť $x_1 \in \mathbb{Z}$, $x_1 \equiv a_i \pmod{m_i}$ pro všechna $i \in \{1, 2, \ldots, k\}$. Chceme: $x_0 \equiv x_1 \pmod{M}$. Je $x_0 \equiv x_1 \pmod{m_i}$ pro každé $i \in \{1, 2, \ldots, k\}$. Připomeňme, že $m_i \perp m_j$ pro $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$. Aplikací věty 2.5.5 dostaneme $x_0 \equiv x_1 \pmod{m_1 m_2 \cdots m_k}$, $x_0 \equiv x_1 \pmod{M}$.

2.6.7. Poznámka. Nechť $m_i \in \mathbb{Z}$, $m_i > 0$ pro $i \in \{1, 2, \ldots, k\}$. Nechť $m_i \perp m_j$ pro $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$. Nechť $a_i \in \mathbb{Z}$ pro $i \in \{1, 2, \ldots, k\}$.

Důkaz věty 2.6.6 poskytuje návod, jak najít nějaké řešení systému kongruencí

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

Pro $i \in \{1, 2, \ldots, k\}$ položíme $M_i = m_1 \cdots m_{i-1} m_{i+1} \cdots m_k$. Je $M_i \perp m_i$. Pak určíme celé číslo M_i' splňující $M_i' M_i \equiv 1 \pmod{m_i}$ – například pomocí rozšířeného Euklidova algoritmu vypočítáme $s, t \in \mathbb{Z}$ taková, že $sM_i + tm_i = 1$, a položíme $M_i' = s$. Řešením systému kongruencí pak bude číslo

$$x_0 = a_1 M_1' M_1 + a_2 M_2' M_2 + \cdots + a_k M_k' M_k$$

Místo x_0 můžeme vzít jakékoli celé číslo x_1 splňující $x_1 \equiv x_0 \pmod{M}$, kde $M = m_1 m_2 \cdots m_k$. Proč? Máme $M / x_1 - x_0$. Pro každé $i \in \{1, 2, \ldots, k\}$ pak m_i / M, takže $m_i / x_1 - x_0$, $x_1 \equiv x_0 \pmod{m_i}$, $x_1 \equiv a_i \pmod{m_i}$ (je totiž $x_0 \equiv a_i \pmod{m_i}$).

Všimněte si, že pro výpočet čísel M_i' ($i \in \{1, 2, \ldots, k\}$) vůbec nepotřebujeme znát čísla a_i. Jestliže tedy přejdeme k systému kongruencí

$$\begin{aligned} x &\equiv b_1 \pmod{m_1} \\ x &\equiv b_2 \pmod{m_2} \\ &\vdots \\ x &\equiv b_k \pmod{m_k} \end{aligned}$$

máme ihned k dispozici jeho řešení

$$b_1 M_1' M_1 + b_2 M_2' M_2 + \cdots + b_k M_k' M_k$$

2.6.8. Příklad. Vyřešíme systém kongruencí

$$\begin{aligned} x &\equiv 2 \pmod{7} \\ x &\equiv 0 \pmod{9} \\ 2x &\equiv 6 \pmod{8} \end{aligned}$$

Nejprve třetí kongruenci upravíme dle věty 2.5.6 a dostaneme ekvivalentní systém

$$\begin{aligned} x &\equiv 2 \pmod 7 \\ x &\equiv 0 \pmod 9 \\ x &\equiv 3 \pmod 4 \end{aligned}$$

Čísla $7, 9, 4$ jsou vzájemně nesoudělná, takže dle Čínské věty o zbytcích má systém právě jedno řešení modulo $7 \cdot 9 \cdot 4 = 252$. Pojďme jej určit (návod máme v poznámce 2.6.7).

Je $M_1 = 9 \cdot 4 = 36$, $M_2 = 7 \cdot 4 = 28$, $M_3 = 7 \cdot 9 = 63$.

Nyní je třeba určit čísla M_1', M_2', M_3' taková, že $M_1' \cdot 36 \equiv 1 \pmod 7$, $M_2' \cdot 28 \equiv 1 \pmod 9$, $M_3' \cdot 63 \equiv 1 \pmod 4$. Pak číslo

$$x_0 = 2 \cdot M_1' \cdot 36 + 0 \cdot M_2' \cdot 28 + 3 \cdot M_3' \cdot 63 = 2 \cdot M_1' \cdot 36 + 3 \cdot M_3' \cdot 63$$

bude řešením zadaného systému kongruencí.

Všimněte si, že číslo M_2' nemusíme znát.

K určení čísla M_1' použijeme rozšířený Euklidův algoritmus – budeme jej aplikovat na vstup $36, 7$.

$$\begin{array}{ccccc} (36,7) & \mapsto_1 & (7,1) & \mapsto_2 & (1,0) \\ (0,1) & \mapsto & (1,5) & \mapsto & (5,36) \\ (1,0) & \mapsto & (0,1) & \mapsto & (1,7) \end{array}$$

Pomocné výpočty:

$$\begin{aligned} 36 &= 5 \cdot 7 + 1 \\ 7 &= 7 \cdot 1 + 0 \end{aligned}$$

Zjistili jsme, že $1 = 1 \cdot 36 + (-5) \cdot 7$; vezmeme tedy $M_1' = 1$.

K určení čísla M_3' opět použijeme rozšířený Euklidův algoritmus – tentokrát jej budeme aplikovat na vstup $63, 4$.

$$\begin{array}{ccccccc} (63,4) & \mapsto_1 & (4,3) & \mapsto_2 & (3,1) & \mapsto_3 & (1,0) \\ (0,1) & \mapsto & (1,15) & \mapsto & (15,16) & \mapsto & (16,31) \\ (1,0) & \mapsto & (0,1) & \mapsto & (1,1) & \mapsto & (1,2) \end{array}$$

Pomocné výpočty:

$$\begin{aligned} 63 &= 15 \cdot 4 + 3 \\ 4 &= 1 \cdot 3 + 1 \\ 3 &= 1 \cdot 3 + 0 \end{aligned}$$

Zjistili jsme, že $1 = (-1) \cdot 63 + 16 \cdot 4$; vezmeme tedy $M_3' = -1$.
Nyní

$$x_0 = 2 \cdot 1 \cdot 36 + 3 \cdot (-1) \cdot 63 = -117$$

Závěr: Systém kongruencí

$$\begin{aligned} x &\equiv 2 \pmod 7 \\ x &\equiv 0 \pmod 9 \\ 2x &\equiv 6 \pmod 8 \end{aligned}$$

má řešení -117, což je jidiné řešení modulo 252. Místo čísla -117 můžeme vzít jakékoli celé číslo x_1 splňující $x_1 \equiv -117 \pmod{252}$, například 135.

Cvičení.

1. Vyřešte kongruenci
$$6x \equiv 9 \pmod{21}$$
2. Vyřešte kongruenci
$$10x \equiv 5 \pmod{21}$$
3. Vyřešte systém kongruencí
$$\begin{aligned} x &\equiv 3 \pmod{11} \\ x &\equiv 6 \pmod 8 \\ x &\equiv 14 \pmod{15} \end{aligned}$$

4. Vyřešte systém kongruencí

$$\begin{aligned} x &\equiv 0 \pmod 2 \\ x &\equiv 0 \pmod 3 \\ x &\equiv 1 \pmod 4 \end{aligned}$$

2.7 Eulerova věta

Jednou z nejdůležitějších funkcí teorie čísel je **Eulerova φ funkce**. Co je to za funkci? Definičním oborem funkce φ je množina všech kladných celých čísel a pro kladné celé číslo n je $\varphi(n)$ rovno počtu všech kladných celých čísel, která jsou menší nebo rovna n a nesoudělná s n.

2.7.1. Definice. Nechť $n \in \mathbb{Z}$, $n > 0$. Pak klademe

$$\varphi(n) = |\{k \in \mathbb{Z};\ 1 \leq k \leq n\ \wedge\ k \perp n\}| = |\{k \in \mathbb{Z};\ 0 \leq k < n\ \wedge\ k \perp n\}|$$

Hodnoty uvedené v následující tabulce si prosím samostatně zkontrolujte.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\varphi(n)$	1	1	2	2	4	2	6	4	6	4	10	4	12	6	8	8	16	6	18

2.7.2. Tvrzení. *Nechť p je prvočíslo, n je kladné celé číslo. Pak*

$$\varphi(p^n) = p^n - p^{n-1}$$

Důkaz. Celá čísla soudělná s číslem p^n jsou právě násobky prvočísla p. Zde je soupis všech celých čísel k, $1 \leq k \leq p^n$, p/k: $1p, 2p, 3p, \ldots, p^{n-1} \cdot p$. Těchto čísel je celkem p^{n-1}. Proto $\varphi(p^n) = p^n - p^{n-1}$.

2.7.3. Definice. Funkce f, jejímž definičním oborem je množina všech kladných celých čísel, se nazývá **multiplikativní**, pokud

$$(\forall m \in \mathbb{Z}^+)(\forall n \in \mathbb{Z}^+)\ m \perp n \Longrightarrow f(mn) = f(m)f(n)$$

2.7.4. Věta. *Eulerova φ funkce je multiplikativní.*

Důkaz. Nechť $m, n \in \mathbb{Z}$, $m > 0$, $n > 0$, $m \perp n$. Chceme: $\varphi(mn) = \varphi(m)\varphi(n)$.

Položme
$A = \{k \in \mathbb{Z};\ 0 \leq k < m \ \wedge\ k \perp m\}$,
$B = \{k \in \mathbb{Z};\ 0 \leq k < n \ \wedge\ k \perp n\}$,
$C = \{k \in \mathbb{Z};\ 0 \leq k < mn \ \wedge\ k \perp mn\}$.

Je $\varphi(m) = |A|$, $\varphi(n) = |B|$, $\varphi(mn) = |C|$. Sestrojíme bijekci $f : C \to A \times B$. Pak bude $|C| = |A \times B| = |A| \cdot |B|$ a tedy $\varphi(mn) = \varphi)(m)\varphi(n)$.

Pro $k \in C$ položme $f(k) = (k \bmod m, k \bmod n)$. Nyní je třeba ukázat několik věcí.

- $(\forall k \in C)\ f(k) \in A \times B$:
  Nechť $k \in C$. Jistě $k \bmod m \in \mathbb{Z}$, $0 \leq k \bmod m < m$, $k \bmod n \in \mathbb{Z}$, $0 \leq k \bmod n < n$. Zbývá ukázat, že $k \bmod m \perp m$, $k \bmod n \perp n$. Ukážeme nejprve, že $k \bmod m \perp m$. Buď $d \in \mathbb{Z}$, $d > 0$, $d/k \bmod m$, d/m. Chceme: $d = 1$. Je $k = qm + k \bmod m$ pro nějaké celé číslo q. Protože d/m a $d/k \bmod m$, máme d/k. Také d/mn. Celkem: d/k, d/mn. Jelikož $k \perp mn$, dostáváme $d = 1$, což jsme chtěli. Obdobně se dokáže, že $k \bmod n \perp n$.

- f je injekce:
  Předpoklad: $k, l \in C$, $f(k) = f(l)$. Chceme: $k = l$. Je $(k \bmod m, k \bmod n) = (l \bmod m, l \bmod n)$; $k \bmod m = l \bmod m$ dává $k \equiv l \pmod m$ (viz 2.5.2), $k \bmod n = l \bmod n$ dává $k \equiv l \pmod n$. Takže $m/k - l$, $n/k - l$, $m \perp n$. Podle věty 2.3.15 $mn/k - l$. Dále

$$\begin{array}{rcccl} 0 & \leq & k & < & mn \\ 0 & \leq & l & < & mn \\ \hline 0 & \leq & k & < & mn \\ -mn & < & -l & \leq & 0 \\ \hline -mn & < & k-l & < & mn \end{array}$$

  Celkem: $-mn < k - l < mn$, $mn/k - l$. Proto $k - l = 0$, $k = l$.

- f je surjekce:
  Nechť $i \in A$, $j \in B$. Hledáme $k \in C$, $f(k) = (i, j)$. Uvažme systém

kongruencí

$$\begin{aligned} x &\equiv i \pmod m \\ x &\equiv j \pmod n \end{aligned}$$

Jelikož $m \perp n$, má podle Čínské věty o zbytcích tento systém nějaké řešení x_0. Položme $k = x_0 \bmod mn$. Je $x_0 = qmn + k$ pro nějaké $q \in \mathbb{Z}$.

Ukážeme, že $k \in C$. Jistě $k \in \mathbb{Z}$, $0 \leq k < mn$. Musíme ještě ukázat, že $k \perp mn$. Buď $d \in \mathbb{Z}$, $d > 0$, d/k, d/mn. Chceme: $d = 1$. Předpokládejme, že $d > 1$. Pak existuje prvočíslo p, p/d. Pak p/k, p/mn. Je $x_0 = qmn + k$, takže p/x_0. Protože p je prvočíslo, p/m nebo p/n. Uvažme nejprve, že p/m. Je $x_0 \equiv i \pmod m$, takže $m/x_0 - i$ a tedy také $p/x_0 - i$. Je $i = (-1)\cdot(x_0 - i) + x_0$ a víme také, že $p/x_0 - i$, p/x_0. Z toho plyne, že p/i. Ovšem také p/m a jelikož $i \perp m$ (je $i \in A$), dostáváme spor. Obdobně dostaneme spor v případě p/n. Nutně tedy $d = 1$.

Nyní ještě ukážeme, že $f(k) = (i, j)$. Máme: $mn/x_0 - k$, $m/x_0 - k$, $x_0 \equiv k \pmod m$, takže $k \equiv i \pmod m$, $k \bmod m = i \bmod m = i$ (je $0 \leq i < m$). Dále, $mn/x_0 - k$, $n/x_0 - k$, $x_0 \equiv k \pmod n$, takže $k \equiv j \pmod n$, $k \bmod n = j \bmod n = j$ (je $0 \leq j < n$). Tudíž $k \bmod m = i$, $k \bmod n = j$, $f(k) = (k \bmod m, k \bmod n) = (i, j)$.

Důkaz je hotov.

2.7.5. Tvrzení. *Nechť f je multiplikativní funkce, $p_1, p_2, \ldots, p_k$ jsou vzájemně různá prvočísla, $n_1, n_2, \ldots, n_k$ jsou kladná celá čísla. Pak*

$$f(p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}) = f(p_1^{n_1}) f(p_2^{n_2}) \cdots f(p_k^{n_k})$$

Důkaz. Stačí si uvědomit, že pro $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$, je $p_i^{n_i} \perp p_j^{n_j}$.

2.7.6. Tvrzení. *Nechť $p_1, p_2, \ldots, p_k$ jsou vzájemně různá prvočísla, $n_1, n_2, \ldots, n_k$ jsou kladná celá čísla, $n = p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}$. Pak*

$$\begin{aligned} \varphi(n) &= (p_1^{n_1} - p_1^{n_1 - 1})(p_2^{n_2} - p_2^{n_2 - 1}) \cdots (p_k^{n_k} - p_k^{n_k - 1}) \\ &= n \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right) \end{aligned}$$

Důkaz. Použijeme 2.7.2, 2.7.4 a 2.7.5. Dále,

$$\begin{aligned}(p_1^{n_1} - p_1^{n_1-1}) \cdots (p_k^{n_k} - p_k^{n_k-1}) &= p_1^{n_1}\left(1 - \frac{1}{p_1}\right) \cdots p_k^{n_k}\left(1 - \frac{1}{p_k}\right) \\ &= p_1^{n_1} \cdots p_k^{n_k}\left(1 - \frac{1}{p_1}\right) \cdots \left(1 - \frac{1}{p_k}\right) \\ &= n\left(1 - \frac{1}{p_1}\right) \cdots \left(1 - \frac{1}{p_k}\right)\end{aligned}$$

2.7.7. Příklad.

1. $\varphi(2022) = \varphi(2 \cdot 3 \cdot 337) = (2-1) \cdot (3-1) \cdot (337-1) = 1 \cdot 2 \cdot 336 = 672$. Samozřejmě, důležité je vědět, že čísla 2, 3 a 337 jsou prvočísla.
2. Číslo 1971 má přesně dva prvočíselné dělitele, a to 3 a 73. Takže $\varphi(1971) = 1971 \cdot \left(1 - \frac{1}{3}\right) \cdot \left(1 - \frac{1}{73}\right) = 1971 \cdot \frac{2}{3} \cdot \frac{72}{73} = 1296$.

Nyní se konečně dostáváme k větě, podle které se jmenuje tato kapitola, tedy k větě Eulerově. Je to základní věta teorie čísel a má řadu aplikací, například v kryptografii – jak brzy uvidíme.

2.7.8. Věta (Eulerova věta). *Nechť $n, a \in \mathbb{Z}$, $n > 0$, $a \perp n$. Pak*

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

Důkaz. Buď $A = \{k \in \mathbb{Z};\ 0 \leq k < n \ \wedge\ k \perp n\}$. Je $\varphi(n) = |A|$. Položme $\varphi(n) = l$, $A = \{a_1, a_2, \ldots, a_l\}$.

Pro $i = 1, 2, \ldots, l$ je $a_i a = nq_i + r_i$, $q_i, r_i \in \mathbb{Z}$, $0 \leq r_i < n$.

- Pro $i = 1, 2, \ldots l$ je $r_i \perp n$:

  Buď $d \in \mathbb{Z}$, $d > 0$, d/r_i, d/n. Chceme: $d = 1$. Předpokládejme, že $d > 1$. Pak existuje prvočíslo p, p/d. Pak p/r_i, p/n. Z toho plyne, že $p/a_i a$. Protože p je prvočíslo, máme $p/a_i \vee p/a$. Obě možnosti dají spor. To bude znamenat, že nutně $d = 1$. Pokud p/a_i, máme p/a_i, p/n, spor (připomeňme, že $a_i \perp n$, jelikož $a_i \in A$). Pokud p/a, máme p/a, p/n, spor (připomeňme, že $a \perp n$).

- Pro $i, j \in \{1, 2, \ldots, l\}$, $i \neq j$, je $r_i \neq r_j$:

  Předpokládejme, že $r_i = r_j$. Pak

$$\begin{aligned} a_i a - nq_i &= a_j a - nq_j \\ a_i a - a_j a &= nq_i - nq_j \\ (a_i - a_j)a &= n(q_i - q_j) \\ (a_i - a_j)a &\equiv 0 \pmod{n} \end{aligned}$$

  Protože $a \perp n$, dostáváme $a_i - a_j \equiv 0 \pmod{n}$, $n/a_i - a_j$.

$$\begin{array}{rcccl} 0 & \leq & a_i & < & n \\ 0 & \leq & a_j & < & n \\ \hline 0 & \leq & a_i & < & n \\ -n & < & -a_j & \leq & 0 \\ \hline -n & < & a_i - a_j & < & n \end{array}$$

  Celkem: $n/a_i - a_j$, $-n < a_i - a_j < n$. Takže $a_i - a_j = 0$, $a_i = a_j$, spor, protože $i, j \in \{1, 2, \ldots, l\}$, $i \neq j$. Nutně tedy $r_i \neq r_j$.

Máme: $0 \leq r_i < n$ pro $i = 1, 2, \ldots, l$, $r_i \perp n$ pro $i = 1, 2, \ldots, l$, $r_i \neq r_j$ pro $i, j \in \{1, 2, \ldots, l\}$, $i \neq j$. Z toho plyne, že $\{r_1, r_2, \ldots, r_l\} = A$.

Je $a_i a \equiv r_i \pmod{n}$ pro $i = 1, 2, \ldots, l$ (protože $a_i a - r_i = nq_i$). Pak

$$a_1 a_2 \cdots a_l a^l \equiv r_1 r_2 \cdots r_l \pmod{n}$$

Položme $b = a_1 a_2 \cdots a_l = r_1 r_2 \cdots r_l$. Pak

$$ba^l \equiv b \pmod{n}$$

Ukážeme, že $b \perp n$. Buď $d \in \mathbb{Z}$, $d > 0$, d/b, d/n. Chceme: $d = 1$. Předpokládejme, že $d > 1$. Pak existuje prvočíslo p, p/d. Pak p/b, p/n. Protože p/b a p je prvočíslo, existuje $i \in \{1, 2, \ldots, l\}$, p/a_i. Takže p/a_i p/n. To je spor, jelikož $a_i \perp n$. Nutně tedy $d = 1$.

Máme: $ba^l \equiv b \pmod{n}$, $b \perp n$. Z toho plyne, že $a^l \equiv 1 \pmod{n}$, $a^{\varphi(n)} \equiv 1 \pmod{n}$. Důkaz je hotov.

2.7.9. Důsledek (Malá Fermatova věta). *Nechť p je prvočíslo, $a \in \mathbb{Z}$. Pak*

$$a^p \equiv a \pmod{p}$$

Pokud navíc $\neg(p/a)$, *pak*

$$a^{p-1} \equiv 1 \pmod{p}$$

Důkaz. Rozlišíme dva případy.

1. p/a:

   Je $a^p - a = a(a^{p-1} - 1)$, $a/a^p - a$. Protože p/a, dostáváme $p/a^p - a$, $a^p \equiv a \pmod{p}$.

2. $\neg(p/a)$:

   Je $a \perp p$ a dle Eulerovy věty $a^{\varphi(p)} \equiv 1 \pmod{p}$. Je $\varphi(p) = p - 1$, takže $a^{p-1} \equiv 1 \pmod{p}$. Pak $a \cdot a^{p-1} \equiv a \cdot 1 \pmod{p}$, $a^p \equiv a \pmod{p}$.

Malou Fermatovu větu lze také dokázat indukcí – zkuste to!

Před následujícím příkladem bude vhodné zformulovat algoritmus umocňování. Využijeme jej také později, například v části 2.8.

Chceme-li spočítat například x^{100}, můžeme samozřejmě postupně počítat x, $x^2, x^3, x^4, x^5, \ldots$ a po 99 násobeních dostaneme výsledek. Můžeme ale postupovat efektivněji a postupně spočítat $x, x^2, x^3, x^6, x^{12}, x^{24}, x^{25}, x^{50}, x^{100}$. Nyní jsme potřebovali pouze 8 násobení.

Algoritmus umocňování (rekurzivní)

Vstup: x (prvek algebraické struktury s asociativním násobením), $n \in \mathbb{Z}$, $n \geq 1$

Výstup: x^n

- $$x^1 = x$$
- pro $n > 1$:

  $$x^n = \left(x^{\frac{n}{2}}\right)^2 \text{ pokud } n \text{ je sudé, } x^n = \left(x^{\frac{n-1}{2}}\right)^2 \cdot x \text{ pokud } n \text{ je liché}$$

Správnost algoritmu je zřejmá.
Označme $T(n)$ počet násobení při výpočtu x^n. Ukážeme teď, že

$$T(n) \leq 2\lfloor \lg n \rfloor$$

Postupujme indukcí. Je $T(1) = 0$, $2\lfloor \lg 1 \rfloor = 2\lfloor 0 \rfloor = 2 \cdot 0 = 0$. Předpokládejme dále, že $n > 1$. Potom máme dvě možnosti:

1. n je sudé: Je $T(n) = T(\frac{n}{2}) + 1 \leq 2\lfloor \lg \frac{n}{2} \rfloor + 1 = 2\lfloor \lg n - 1 \rfloor + 1 = 2(\lfloor \lg n \rfloor - 1) + 1 = 2\lfloor \lg n \rfloor - 2 + 1 = 2\lfloor \lg n \rfloor - 1 < 2\lfloor \lg n \rfloor$.

2. n je liché: Je $T(n) = T(\frac{n-1}{2}) + 2 \leq 2\lfloor \lg \frac{n-1}{2} \rfloor + 2 = 2\lfloor \lg(n-1) - 1 \rfloor + 2 = 2(\lfloor \lg(n-1) \rfloor - 1) + 2 = 2\lfloor \lg(n-1) \rfloor - 2 + 2 = 2\lfloor \lg(n-1) \rfloor \leq 2\lfloor \lg n \rfloor$.

2.7.10. Příklad. Určíme poslední dvojčíslí čísel 7^{2962}, $7^{2962!}$ a 6^{2962}. Nejprve si uvědomme, že poslední dvojčíslí kladného celého čísla n je rovno zbytku po dělení čísla n číslem 100, tj. je rovno číslu $n \bmod 100$.

1. Zabývejme se teď číslem 7^{2962}. Je $7 \perp 100$, takže dle Eulerovy věty je $7^{\varphi(100)} \equiv 1 \pmod{100}$. Číslo 100 má přesně dva prvočíselné dělitele, a to 2 a 5, tudíž $\varphi(100) = 100 \cdot (1 - \frac{1}{2}) \cdot (1 - \frac{1}{5}) = 100 \cdot \frac{1}{2} \cdot \frac{4}{5} = 40$ (při výpočtu jsme použili tvrzení 2.7.6). Je tedy $7^{40} \equiv 1 \pmod{100}$. Dále, $2962 = 74 \cdot 40 + 2$ a

   $$7^{2962} = 7^{74 \cdot 40 + 2} = (7^{40})^{74} \cdot 7^2 \equiv 1^{74} \cdot 7^2 = 7^2 = 49 \pmod{100}$$

   Tedy $7^{2962} \equiv 49 \pmod{100}$, $7^{2962} \bmod 100 = 49 \bmod 100$, $7^{2962} \bmod 100 = 49$.

   Závěr: poslední dvojčíslí čísla 7^{2962} je rovno 49.

2. Zřejmě $2962! = 40 \cdot k$ pro nějaké kladné celé číslo k, takže

   $$7^{2962!} = 7^{40k} = (7^{40})^k \equiv 1^k = 1 \pmod{100}$$

   Tedy $7^{2962!} \equiv 1 \pmod{100}$, $7^{2962!} \bmod 100 = 1$.

   Závěr: poslední dvojčíslí čísla $7^{2962!}$ je rovno 01.

3. Zbývá nám určit poslední dvojčíslí čísla 6^{2962}. Jelikož $\neg(6 \perp 100)$, nemůžeme postupovat obdobně jako v předchozích dvou případech.

Ovšem $6^{2962} = 2^{2962} \cdot 3^{2962}$, přičemž $3 \perp 100$, takže opět dle Eulerovy věty $3^{40} \equiv 1 \pmod{100}$ a

$$3^{2962} = 3^{74 \cdot 40 + 2} = (3^{40})^{74} \cdot 3^2 \equiv 1^{74} \cdot 3^2 = 3^2 = 9 \pmod{100}$$

Je tedy $3^{2962} \equiv 9 \pmod{100}$. Ještě se musíme vypořádat s 2^{2962}. Tady již nemůžeme použít Eulerovu větu.

$$\begin{aligned}
2^{2962} &= (2^{1481})^2 \\
2^{1481} &= (2^{740})^2 \cdot 2 \\
2^{740} &= (2^{370})^2 \\
2^{370} &= (2^{185})^2 \\
2^{185} &= (2^{92})^2 \cdot 2 \\
2^{92} &= (2^{46})^2 \\
2^{46} &= (2^{23})^2 \\
2^{23} &= (2^{11})^2 \cdot 2 \\
2^{11} &= (2^5)^2 \cdot 2 \\
2^5 &= (2^2)^2 \cdot 2 \\
2^2 &= (2^1)^2 \\
2^1 &= 2
\end{aligned}$$

Je tedy

$$\begin{aligned}
2^1 &\equiv 02 \pmod{100}\\
2^2 &\equiv 04 \pmod{100}\\
2^5 &\equiv 32 \pmod{100}\\
2^{11} &\equiv 48 \pmod{100}\\
2^{23} &\equiv 08 \pmod{100}\\
2^{46} &\equiv 64 \pmod{100}\\
2^{92} &\equiv 96 \pmod{100}\\
2^{185} &\equiv 32 \pmod{100}\\
2^{370} &\equiv 24 \pmod{100}\\
2^{740} &\equiv 76 \pmod{100}\\
2^{1481} &\equiv 52 \pmod{100}\\
2^{2962} &\equiv 04 \pmod{100}
\end{aligned}$$

Máme $2^{2962} \equiv 4 \pmod{100}$, $3^{2962} \cdot 2^{2962} \equiv 9 \cdot 4 = 36 \pmod{100}$, $6^{2962} \equiv 36 \pmod{100}$, $6^{2962} \bmod 100 = 36$.

Závěr: poslední dvojčíslí čísla 6^{2962} je rovno 36.

Cvičení.

1. Jestliže f je nekonstantní multiplikativní funkce, pak $f(1) = 1$. Dokažte.

2. Celé číslo se nazývá bezčtvercové, pokud není dělitelné druhou mocninou žádného celého čísla většího než 1. Pro $n \in \mathbb{Z}$, $n > 0$, položme $f(n) = 1$ pokud n je bezčtvercové, $f(n) = 0$ pokud n není bezčtvercové. Dokažte, že funkce f je multiplikativní.

3. Vypočtěte $\varphi(15!)$.

4. Dokažte Malou Fermatovu větu pro $p = 2$.

5. Dokažte Malou Fermatovu větu indukcí. Rada: použijte binomickou větu.

2.8 Kryptografie s veřejným klíčem

2.8.1 RSA kryptosystém

Základní idea kryptosystému s veřejným klíčem, který navrhli Rivest, Shamir a Adleman (1978), je velmi jednoduchá:

je snadné vynásobit dvě velká prvočísla p a q, ale zdá se, že je velmi obtížné najít p, q, pokud je dán pouze jejich součin $n = pq$ a n je velké.

Návrh kryptosystému RSA

Zvolíme dvě velká prvočísla p, q, $p \neq q$, položíme

$$n = pq, \ \varphi(n) = (p-1)(q-1)$$

kde φ je Eulerova funkce.

Zvolíme velké celé číslo d, $0 < d < \varphi(n)$, a celé číslo e, $0 < e < \varphi(n)$, tak, že

$$ed \equiv 1 \pmod{\varphi(n)}$$

(později se zmíníme o tom, jak to udělat).

Pak

$$n \text{ (modul)}, \ e \text{ (šifrovací exponent)}$$

tvoří **veřejný klíč** a

$$p, \ q, \ d \text{ (dešifrovací exponent)}$$

tvoří **soukromý klíč**.

Zašifrování:

Abychom získali šifrový text c, zprávu $w \in \mathbb{Z}$, $0 \leq w < n$, zašifrujeme jako

$$c = w^e \bmod n$$

Dešifrování:

$$w = c^d \bmod n$$

Detaily a správnost: Zprávu nejprve zakódujeme jako slovo nad abecedou $\{0,1,2,3,4,5,6,7,8,9\}$, pak rozdělíme do bloků délky i, kde $i \in \mathbb{Z}$, $10^i \leq n < 10^{i+1}$. Každý blok pak bereme jako celé číslo (je menší než $10^i \leq n$) a zašifrujeme ho modulárním umocňováním (viz výše).

Správnost dešifrování vyplývá z následující věty:

2.8.1. Věta. *Nechť p, q jsou prvočísla, $p \neq q$, $n = pq$, d, e jsou celá čísla, $0 < d < \varphi(n)$, $0 < e < \varphi(n)$, $ed \equiv 1 \pmod{\varphi(n)}$, $w \in \mathbb{Z}$, $0 \leq w < n$, $c = w^e \bmod n$. Pak $w = c^d \bmod n$.*

Důkaz. Víme, že $\varphi(n)/ed - 1$, takže existuje $j \in \mathbb{Z}$, $ed - 1 = j\varphi(n)$, $ed = 1 + j\varphi(n)$. Protože $e > 0$, $d > 0$, je $ed > 0$, $ed - 1 \geq 0$. Takže $j\varphi(n) \geq 0$, $j \geq 0$ (je $\varphi(n) > 0$).

Je $w^e \equiv w^e \bmod n \pmod{n}$ (viz 2.5.7), $w^e \equiv c \pmod{n}$, $(w^e)^d \equiv c^d \pmod{n}$, $w^{ed} \equiv c^d \pmod{n}$. Rozlišíme čtyři případy:

1. $\neg(p/w), \neg(q/w)$:

   Je $w \perp n$ a dle Eulerovy věty $w^{\varphi(n)} \equiv 1 \pmod{n}$. Pak

   $$w^{ed} = w^{1+j\varphi(n)} = w \cdot (w^{\varphi(n)})^j \equiv w \cdot 1^j = w \pmod{n}$$

   Je tedy $w^{ed} \equiv w \pmod{n}$, $c^d \equiv w \pmod{n}$, $c^d \bmod n = w \bmod n = w$ (je $0 \leq w < n$), $w = c^d \bmod n$.

2. $p/w, \neg(q/w)$:

   Protože p/w, máme $p/w^{ed} - w$, $w^{ed} \equiv w \pmod{p}$. Dle Malé Fermatovy věty $w^{q-1} \equiv 1 \pmod{q}$. Pak $(w^{q-1})^{p-1} \equiv 1^{p-1} \pmod{q}$, $w^{(p-1)(q-1)} \equiv 1 \pmod{q}$, $w^{\varphi(n)} \equiv 1 \pmod{q}$, takže

   $$w^{ed} = w \cdot (w^{\varphi(n)})^j \equiv w \cdot 1^j = w \pmod{q}$$

   Ukázali jsme, že $w^{ed} \equiv w \pmod{q}$. Máme: $w^{ed} \equiv w \pmod{p}$, $w^{ed} \equiv w \pmod{q}$. Protože $p \neq q$, p, q jsou prvočísla, je $p \perp q$, $w^{ed} \equiv w \pmod{pq}$ (viz 2.5.5). Je $pq = n$, tudíž $w^{ed} \equiv w \pmod{n}$, $c^d \equiv w \pmod{n}$, $c^d \bmod n = w \bmod n = w$, $w = c^d \bmod n$.

3. $\neg(p/w), q/w$:

   Postupujeme obdobně jako v případu 2.

4. $p/w, q/w$:

   Pak pq/w, n/w. Jelikož $0 \leq w < n$, je $w = 0$. Pak $c = 0^e \pmod{n}$, $c = 0 \pmod{n}$, $c = 0$. Chceme tedy ukázat, že $0 = 0^d \pmod{n}$. To jistě platí, protože $0^d = 0$.

Následující příklad je převzat z knihy [6] (Example 8.3.15, strana 485).

2.8.2. Příklad. Zvolme $p = 41$ a $q = 61$. Pak $n = 2501$ a $\varphi(n) = 2400$. Dále zvolme $d = 2087$, $e = 23$. Je $ed = 48001 = 20 \cdot 2400 + 1$, takže $ed \equiv 1 \pmod{\varphi(n)}$. Dejme tomu, že chceme zašifrovat zprávu "KARLSRUHE". Nejprve reprezentujeme písmena číslem jejich pozice v abecedě:

A	B	C	D	E	F	G	H	I	J	K	L	M
00	01	02	03	04	05	06	07	08	09	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Zprávu zakódujeme jako

$$100017111817200704$$

Jelikož $10^3 \leq n < 10^4$, zprávu rozdělíme do bloků délky 3 a dostaneme

$$100\ 017\ 111\ 817\ 200\ 704$$

Abychom zprávu zašifrovali, musíme vypočítat $100^{23} \bmod 2501$, $17^{23} \bmod 2501$, $111^{23} \bmod 2501$, $817^{23} \bmod 2501$, $200^{23} \bmod 2501$, $704^{23} \bmod 2501$.

Využijeme rekurzivní algoritmus umocňování:

$$\begin{aligned} x^{23} &= (x^{11})^2 \cdot x \\ x^{11} &= (x^5)^2 \cdot x \\ x^5 &= (x^2)^2 \cdot x \\ x^2 &= (x^1)^2 \\ x^1 &= x \end{aligned}$$

$$
\begin{aligned}
100^{1} &\equiv 100 \pmod{2501} \\
100^{2} &\equiv 2497 \pmod{2501} \\
100^{5} &\equiv 1600 \pmod{2501} \\
100^{11} &\equiv 141 \pmod{2501} \\
100^{23} &\equiv 2306 \pmod{2501}
\end{aligned}
$$

$$
\begin{aligned}
17^{1} &\equiv 17 \pmod{2501} \\
17^{2} &\equiv 289 \pmod{2501} \\
17^{5} &\equiv 1790 \pmod{2501} \\
17^{11} &\equiv 421 \pmod{2501} \\
17^{23} &\equiv 1893 \pmod{2501}
\end{aligned}
$$

$$
\begin{aligned}
111^{1} &\equiv 111 \pmod{2501} \\
111^{2} &\equiv 2317 \pmod{2501} \\
111^{5} &\equiv 1514 \pmod{2501} \\
111^{11} &\equiv 2024 \pmod{2501} \\
111^{23} &\equiv 621 \pmod{2501}
\end{aligned}
$$

$$
\begin{aligned}
817^{1} &\equiv 817 \pmod{2501} \\
817^{2} &\equiv 2223 \pmod{2501} \\
817^{5} &\equiv 782 \pmod{2501} \\
817^{11} &\equiv 342 \pmod{2501} \\
817^{23} &\equiv 1380 \pmod{2501}
\end{aligned}
$$

$$
\begin{aligned}
200^{1} &\equiv 200 \pmod{2501} \\
200^{2} &\equiv 2485 \pmod{2501} \\
200^{5} &\equiv 1180 \pmod{2501} \\
200^{11} &\equiv 1153 \pmod{2501} \\
200^{23} &\equiv 490 \pmod{2501}
\end{aligned}
$$

$$
\begin{aligned}
704^{1} &\equiv 704 \pmod{2501} \\
704^{2} &\equiv 418 \pmod{2501} \\
704^{5} &\equiv 1514 \pmod{2501} \\
704^{11} &\equiv 760 \pmod{2501} \\
704^{23} &\equiv 313 \pmod{2501}
\end{aligned}
$$

Vypočítali jsme tedy, že

$$
\begin{aligned}
100^{23} \bmod 2501 &= 2306 \\
17^{23} \bmod 2501 &= 1893 \\
111^{23} \bmod 2501 &= 621 \\
817^{23} \bmod 2501 &= 1380 \\
200^{23} \bmod 2501 &= 490 \\
704^{23} \bmod 2501 &= 313
\end{aligned}
$$

a máme šifrový text

$$2306\ 1893\ 621\ 1380\ 490\ 313$$

Abychom zprávu dešifrovali, musíme vypočítat

$$
\begin{aligned}
2306^{2087} \bmod 2501 &= 100 \\
1893^{2087} \bmod 2501 &= 17 \\
621^{2087} \bmod 2501 &= 111
\end{aligned}
$$

atd.
Opět jsme využili rekurzivní algoritmus umocňování:

$$
\begin{aligned}
x^{2087} &= (x^{1043})^2 \cdot x \\
x^{1043} &= (x^{521})^2 \cdot x \\
x^{521} &= (x^{260})^2 \cdot x \\
x^{260} &= (x^{130})^2 \\
x^{130} &= (x^{65})^2 \\
x^{65} &= (x^{32})^2 \cdot x \\
x^{32} &= (x^{16})^2 \\
x^{16} &= (x^{8})^2 \\
x^{8} &= (x^{4})^2 \\
x^{4} &= (x^{2})^2 \\
x^{2} &= (x^{1})^2 \\
x^{1} &= x
\end{aligned}
$$

$$
\begin{aligned}
2306^{1} &\equiv 2306 \pmod{2501} \\
2306^{2} &\equiv 510 \pmod{2501} \\
2306^{4} &\equiv 2497 \pmod{2501} \\
2306^{8} &\equiv 16 \pmod{2501} \\
2306^{16} &\equiv 256 \pmod{2501} \\
2306^{32} &\equiv 510 \pmod{2501} \\
2306^{65} &\equiv 780 \pmod{2501} \\
2306^{130} &\equiv 657 \pmod{2501} \\
2306^{260} &\equiv 1477 \pmod{2501} \\
2306^{521} &\equiv 1937 \pmod{2501} \\
2306^{1043} &\equiv 1082 \pmod{2501} \\
2306^{2087} &\equiv 100 \pmod{2501}
\end{aligned}
$$

$$
\begin{aligned}
1893^{1} &\equiv 1893 \pmod{2501} \\
1893^{2} &\equiv 2017 \pmod{2501} \\
1893^{4} &\equiv 1663 \pmod{2501} \\
1893^{8} &\equiv 1964 \pmod{2501} \\
1893^{16} &\equiv 754 \pmod{2501} \\
1893^{32} &\equiv 789 \pmod{2501} \\
1893^{65} &\equiv 1069 \pmod{2501} \\
1893^{130} &\equiv 2305 \pmod{2501} \\
1893^{260} &\equiv 901 \pmod{2501} \\
1893^{521} &\equiv 2344 \pmod{2501} \\
1893^{1043} &\equiv 1901 \pmod{2501} \\
1893^{2087} &\equiv 17 \pmod{2501}
\end{aligned}
$$

$$
\begin{aligned}
621^{1} &\equiv 621 \pmod{2501} \\
621^{2} &\equiv 487 \pmod{2501} \\
621^{4} &\equiv 2075 \pmod{2501} \\
621^{8} &\equiv 1404 \pmod{2501} \\
621^{16} &\equiv 428 \pmod{2501} \\
621^{32} &\equiv 611 \pmod{2501} \\
621^{65} &\equiv 2146 \pmod{2501} \\
621^{130} &\equiv 975 \pmod{2501} \\
621^{260} &\equiv 245 \pmod{2501} \\
621^{521} &\equiv 621 \pmod{2501} \\
621^{1043} &\equiv 2307 \pmod{2501} \\
621^{2087} &\equiv 111 \pmod{2501}
\end{aligned}
$$

Cvičení.

1. Vezměme $p = 17$, $q = 23$. Dostaneme $n = 391$, $\varphi(n) = 352$. Nechť $e = 29$ a $d = 85$. Zpráva 100, 017, 111, 817, 200, 704 je zašifrována jako 104, 204, 314, 154, 064, 295. Po dešifrování jsme dostali 100, 017, 111, 035, 200, 313. Kde je problém?

2. Uvažme RSA kryptosystém s $p = 47$, $q = 71$ a $e = 79$.

   - Vypočtěte d.
   - Zašifrujte zprávu "THE TRUTH IS MORE CERTAIN THAN PROBABLE". Písmena reprezentujte číslem jejich pozice v abecedě (tak jako v příkladu 2.8.2), mezery ignorujte.
   - Dešifrujte

     3301 13963 2120 1789 1701 2639 895 1150 742 1633 1572 1550
     2668 2375 1643 108

2.8.2 Analýza kryptosystému RSA

Pojďme si nyní popovídat o několika předpokladech, které jsou zásadní pro návrh RSA kryptosystému.

Prvním předpokladem je, že umíme snadno nacházet velká prvočísla.

Existuje několik rychlých pravděpodobnostních testů prvočíselnosti. Jeden si teď ukážeme. Bude to překvapivě jednoduchá procedura.

Algoritmus P (*Pravděpodobnostní test prvočíselnosti*)

Je-li dáno liché celé číslo n, $n > 1$, pokusí se tento algoritmus rozhodnout, jestli je n prvočíslem nebo ne. Nechť tedy $n = 1 + 2^k r$, kde k je kladné celé číslo a r je liché kladné celé číslo.

P1. [Vygenerování x.] Nechť x je náhodné celé číslo, $1 < x < n$.

P2. [Umocnění.] Přiřaďte $j \leftarrow 0$ a $y \leftarrow x^r \bmod n$. (K výpočtu $x^r \bmod n$ můžeme využít rekurzivní algoritmus umocňování.)

P3. [Hotovo?] (Nyní je $y = x^{2^j r} \bmod n$.) Je-li $y = n - 1$, nebo je-li $y = 1$ a $j = 0$, algoritmus končí a oznámí "n je pravděpodobně prvočíslem". Je-li $y = 1$ a $j > 0$, jděte na P5.

P4. [Zvětšení j.] Zvětšete j o 1. Je-li $j < k$, přiřaďte $y \leftarrow y^2 \bmod n$ a vraťte se na P3.

P5. [Není prvočíslo.] Výpočet ukončete a oznamte, že "n rozhodně není prvočíslem".

Řekněme si teď, na jaké myšlence je založen algoritmus P. Předpokládejme, že $x^r \bmod n \neq 1$ a $n = 1 + 2^k r$ je prvočíslo. Pak posloupnost

$$x^r \bmod n,\ x^{2r} \bmod n,\ x^{4r} \bmod n, \ldots, x^{2^k r} \bmod n$$

končí číslem 1 a hodnota těsně před prvním výskytem 1 bude $n - 1$. Proč?

- Protože $1 < x < n$, máme $\neg(n/x)$ a dle Malé Fermatovy věty (2.7.9) $x^{n-1} \equiv 1 \pmod{n}$, $x^{2^k r} \bmod n = 1$.
- Buď j celé číslo, $1 \leq j \leq k$, $x^{2^j r} \bmod n = 1$, $x^{2^i r} \bmod n \neq 1$ pro všechna celá čísla i, $0 \leq i < j$. Chceme ukázat, že $x^{2^{j-1} r} \bmod n = n - 1$. Položme $y = x^{2^{j-1} r}$. Je $y^2 = x^{2^j r}$, $y^2 \bmod n = 1$, $y^2 \equiv 1 \pmod{n}$,

n/y^2-1, $n/(y-1)(y+1)$. Protože n je prvočíslo, máme $n/y-1$ nebo $n/y+1$. Předpokládejme, že $n/y-1$. Pak $y \equiv 1 \pmod{n}$, $y \bmod n = 1$, $x^{2^{j-1}r} \bmod n = 1$, spor. Nutně tedy $n/y+1$, Pak $y \equiv -1 \pmod{n}$, $y \bmod n = -1 \bmod n = n-1$, $x^{2^{j-1}r} \bmod n = n-1$.

Lze dokázat následující základní fakt:

Nechť p_n je pravděpodobnost, s níž se Algoritmus P při svém odhadu mýlí, pokud n je liché celé číslo, $n > 1$. Pak pro všechna n je $p_n < \frac{1}{4}$.

My to zde dokazovat nebudeme, o důkazu si můžete přečíst v [8], kapitola 4.5.4, cvičení 22.

Podstatné je, že pravděpodobnost neúspěchu je ohraničená nezávisle na hodnotě n.

Uvažujme nyní, že algoritmus P spustíme opakovaně a že při každém vstupu do kroku P1 zvolíme x nezávisle a náhodně. Pokud algoritmus byť jednou oznámí, že n není prvočíslem, je tato odpověď naprosto jistá. Jestliže ale třeba desetkrát po sobě odpoví, že n je pravděpodobně prvočíslem, můžeme říci, že je téměř jistě prvočíslem. Pravděpodobnost, že algoritmus dá nesprávnou odpověď v 10 po sobě jdoucích případech, je totiž menší než $\left(\frac{1}{4}\right)^{10} = \frac{1}{1048576}$.

Nyní můžeme najít k-ciferné prvočíslo metodou postupné volby. Vytvoříme náhodné k-ciferné liché celé číslo a testem prvočíselnosti zjistíme, zda je prvočíslem. Pokud ne, vytvoříme další náhodné k-ciferné liché celé číslo, atd. Pokusme se určit průměrný počet voleb potřebný k nalezení k-ciferného prvočísla.

Počet všech k-ciferných lichých celých čísel je roven $\frac{1}{2}(10^k - 10^{k-1}) = \frac{9}{2} \cdot 10^{k-1}$. Ještě by to chtělo znát, aspoň přibližně, počet všech k-ciferných prvočísel. K tomu využijeme klasický výsledek teorie čísel, tzv. Prvočíselnou větu.

Pro reálné číslo x označme $\pi(x)$ počet všech prvočísel menších nebo rovných číslu x. Takže například $\pi(20) = 8$, protože existuje právě 8 prvočísel menších nebo rovných dvaceti, totiž 2, 3, 5, 7, 11, 13, 17, 19. Připomeňme ještě, že pro funkce $f(x)$, $g(x)$ zápis $f(x) \sim g(x)$ znamená $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$.

Prvočíselná věta

$$\pi(x) \sim \frac{x}{\ln x}$$

Počet všech k-ciferných prvočísel je tedy $\pi(10^k)-\pi(10^{k-1})$ a to je přibližně rovno

$$\begin{aligned}\frac{10^k}{\ln 10^k}-\frac{10^{k-1}}{\ln 10^{k-1}} &= \frac{10^k}{k\ln 10}-\frac{10^{k-1}}{(k-1)\ln 10}\\ &= \frac{(k-1)10^k-k10^{k-1}}{k(k-1)\ln 10}\\ &= \frac{10^{k-1}((k-1)10-k)}{k(k-1)\ln 10}\\ &= \frac{10^{k-1}(9k-10)}{k(k-1)\ln 10}\end{aligned}$$

Průměrný počet voleb potřebný k nalezení k-ciferného prvočísla je tedy roven

$$\frac{\frac{9}{2}\cdot 10^{k-1}}{\frac{10^{k-1}(9k-10)}{k(k-1)\ln 10}} = \frac{9\cdot 10^{k-1}k(k-1)\ln 10}{2\cdot 10^{k-1}(9k-10)} = \frac{9k(k-1)\ln 10}{18k-20}$$

Například průměrný počet voleb potřebný k nalezení padesáticiferného prvočísla je

$$\frac{9\cdot 50\cdot(50-1)\ln 10}{18\cdot 50-20}\doteq 58$$

Další otázkou je, jak získáme celá čísla d, e, $0<d<\varphi(n)$, $0<e<\varphi(n)$ (navíc, d by mělo být velké), taková, že

$$ed\equiv 1 \pmod{\varphi(n)}$$

Zvolíme velké celé číslo d, $0<d<\varphi(n)$. Přejeme si, aby $d\perp\varphi(n)$. Abychom prověřili, zda $d\perp\varphi(n)$, použijeme rozšířený Euklidův algoritmus. Pokud $GCD(\varphi(n),d)\neq 1$, zvolíme jiné d. Pokud $GCD(\varphi(n),d)=1$, rozšířený Euklidův algoritmus nám také dal celá čísla s, t, $s\varphi(n)+td=1$. Je $td\equiv 1 \pmod{\varphi(n)}$. Položme $e=t \bmod \varphi(n)$. Je $0\leq e<\varphi(n)$. Dále $t\equiv t \bmod \varphi(n) \pmod{\varphi(n)}$, $t\equiv e \pmod{\varphi(n)}$, $td\equiv ed \pmod{\varphi(n)}$, $1\equiv ed \pmod{\varphi(n)}$. Zbývá si uvědomit, že $e\neq 0$. Předpokládejme naopak, že $e=0$. Pak $t \bmod \varphi(n)=0$, $\varphi(n)/t$ a tedy $\varphi(n)/1$. To je však spor, protože $\varphi(n)=(p-1)(q-1)>1$ (uvědomme si, že p,q jsou velká prvočísla). Nutně tedy $e\neq 0$.

Je přirozené se ptát, jak je obtížné najít velké celé číslo d, $0<d<\varphi(n)$, $d\perp\varphi(n)$. Z následující věty vyplývá, že je to poměrně jednoduché. Pravděpodobnost jevu A označíme $Pr(A)$.

2.8.3. Věta. *Nechť u a v jsou náhodně zvolená kladná celá čísla. Pak*

$$Pr(u \perp v) = \frac{6}{\pi^2}$$

Důkaz. Poznamenejme hned na počátku, že tento důkaz nebude důkaz, ale pouze heuristické zdůvodnění. S tím se nyní spokojíme. Předpokládejme (bez důkazu) existenci pravděpodobnosti P jevu, že $u \perp v$. Máme ukázat, že $P = \frac{6}{\pi^2}$.

Nechť n je kladné celé číslo. Označme p_n počet všech uspořádaných dvojic $(k, l) \in \mathbb{Z} \times \mathbb{Z}$ takových, že $1 \leq k \leq n$, $1 \leq l \leq n$, $k \perp l$. Je

$$P = \lim_{n\to\infty} \frac{p_n}{n^2}$$

Buď d kladné celé číslo. Označme P_d pravděpodobnost toho, že pro náhodně zvolená kladná celá čísla x a y je $GCD(x, y) = d$. Položme

$$A_{nd} = \{(s, t) \in \mathbb{Z} \times \mathbb{Z} |\ 1 \leq s \leq nd,\ 1 \leq t \leq nd,\ GCD(s, t) = d\}$$

Je

$$P_d = \lim_{n\to\infty} \frac{|A_{nd}|}{(nd)^2}$$

Ovšem

$$A_{nd} = \{(kd, ld) |\ (k, l) \in \mathbb{Z} \times \mathbb{Z},\ 1 \leq k \leq n,\ 1 \leq l \leq n,\ k \perp l\}$$

Je tedy $|A_{nd}| = p_n$,

$$P_d = \lim_{n\to\infty} \frac{p_n}{n^2 d^2} = \frac{1}{d^2} \cdot \lim_{n\to\infty} \frac{p_n}{n^2} = \frac{1}{d^2} \cdot P = \frac{P}{d^2}$$

Nyní máme

$$\begin{aligned}
P_1 + P_2 + P_3 + P_4 + \ldots &= 1 \\
\frac{P}{1^2} + \frac{P}{2^2} + \frac{P}{3^2} + \frac{P}{4^2} + \ldots &= 1 \\
P \cdot \left(\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \ldots \right) &= 1 \\
P \cdot \frac{\pi^2}{6} &= 1 \\
P &= \frac{6}{\pi^6}
\end{aligned}$$

Povšimněme si, že $\frac{6}{\pi^2} = 0,6079\ldots$. Pro náhodně zvolená kladná celá čísla u a v je tedy $Pr(u \perp v) \doteq 61\%$. Velké celé číslo d, $0 < d < \varphi(n)$, $d \perp \varphi(n)$, můžeme hledat metodou postupné volby. Číslo d zvolíme náhodně a Euklidovým algoritmem zjistíme, zda je nesoudělné s $\varphi(n)$. Pokud ne, náhodně zvolíme další d, atd.

2.8.4. Poznámka. V důkazu věty 2.8.3 hrál důležitou roli fakt, že

$$\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots = \frac{\pi^2}{6}$$

Řadu převrácených hodnot čtverců kladných celých čísel poprvé sečetl Euler v roce 1734. V knize [2] v kapitole 8 najdete tři elegantní a chytré důkazy uvedené rovnosti (třetí z důkazů je zcela elementární). Knihu doporučuji všem, kteří mají rádi skvělé nápady a chytré postřehy.

Nyní se zdá, že navrhnout RSA kryptosystém je zcela jednoduché. Ve skutečnosti to však úplně jednoduché není. Aby byl navržený kryptosystém dostatečně bezpečný, čísla p, q, d a e musí být zvolena pečlivě - tak, aby byly splněny různé podmínky, například

1. Zmínili jsme již několikrát, že číslo d má být velké. Proč? Jinak by se totiž dešifrování dalo provést vyzkoušením všech malých d. Lze ukázat, že malé e by také mohlo být bezpečnostním rizikem.

2. Rozdíl $|p-q|$ by neměl být příliš malý. Proč? Předpokládejme, že číslo $|p-q|$ je malé a třeba $p > q$. Pak číslo $\frac{p+q}{2}$ je jen o málo větší než $\sqrt{n}$, jelikož $(\frac{p+q}{2})^2 - n = (\frac{p-q}{2})^2$. Uvědomme si, že čísla $\frac{p+q}{2}$ a $\frac{p-q}{2}$ jsou celá, jelikož p a q jsou lichá celá čísla (jsou to velká prvočísla). Testujeme postupně celá čísla x větší než $\sqrt{n}$ tak dlouho, až $x^2 - n$ je čtvercem nějakého kladného celého čísla y. Do testováni lze zapojit modulární aritmetiku, například musí být $(x^2 - n) \bmod 5 \in \{0, 1, 4\}$. Protože číslo $|p-q|$ je malé, najdeme čísla x a y "brzy". Pak bude $x^2 - n = y^2$, $x^2 - y^2 = n$, $(x+y)(x-y) = n$, $p = x + y$, $q = x - y$ (v případě $x - y = 1$ zvětšíme x o 1 a pokračujeme dále). V podstatě se jedná o Fermatovu metodu hledání prvočíselného rozkladu - viz kapitolu 4.5.4 v knize [8].

Může se zdát, že znalost veřejného klíče, tj. znalost čísel n a e, je postačující pro dešifrování. Víme přece, že $ed \equiv 1 \pmod{\varphi(n)}$. Takže $\varphi(n)/ed-1$, $ed - 1 = \varphi(n)r$ pro nějaké celé číslo r. Z rovnosti $ed - \varphi(n)r = 1$ plyne, že $e \perp \varphi(n)$. Rozšířeným Euklidovým algoritmem určíme celá čísla s, t, $s\varphi(n) + te = 1$ a pak $d = t \bmod \varphi(n)$. Ovšem k tomuto výpočtu čísla d potřebujeme znát číslo $\varphi(n)$. Samozřejmě, $\varphi(n)$ lze určit snadno, známe-li prvočísla p a q. Avšak prvočísla p a q nepatří do veřejného klíče, neboť kryptosystém RSA je založen právě na tom, že je velmi obtížné najít p, q, pokud je dán pouze jejich součin $n = pq$ a n je velké. Je možné určit $\varphi(n)$ nějak snadněji než tak, že najdeme prvočíselný rozklad čísla n? Následující tvrzení ukazuje, že to možné není.

2.8.5. Věta. *Nechť p a q jsou prvočísla, $n = pq$. Rozložit číslo n na součin prvočísel je stejně těžké, jako vypočítat $\varphi(n)$.*

Důkaz. Lze předpokládat, že $p \geq q$.

1. Předpokládejme, že známe p a q. Pak $\varphi(n) = (p-1)(q-1)$.

2. Předpokládejme, že známe n a $\varphi(n)$. Pak

$$\varphi(n) = (p-1)(q-1) = pq - p - q + 1 = n - (p+q) + 1$$

a tedy

$$p + q = n - \varphi(n) + 1$$

Dále

$$(p-q)^2 = p^2 - 2pq + q^2 = p^2 + 2pq + q^2 - 4pq = (p+q)^2 - 4n$$

a tedy

$$p - q = \sqrt{(p+q)^2 - 4n}$$

Ze znalosti $p+q$ a $p-q$ již snadno určíme p a q.

Cvičení.

1. Pomocí algoritmu P rozhodněte, zda 857 je prvočíslo.

2. Pomocí algoritmu P rozhodněte, zda 817 je prvočíslo.

3 Grupy

Na počátku oddílu, věnovaného grupám, bych rád zmínil knihu, do které by mohli nahlédnout případní zájemci o hlubší studium teorie grup. Zmiňuji se o ní mimo jiné proto, že kniha je psaná česky a je volně dostupná na internetu. Jedná se o starší knihu vynikajícího českého matematika:

Otakar Borůvka: *Základy teorie grupoidů a grup*. Nakladatelství Československé akademie věd, Praha, 1962.
`https://dml.cz/handle/10338.dmlcz/401378`

V této kapitole se poměrně podrobně seznámíte se základy teorie grup. Nebudou zde žádné úvody do problematiky, žádné komentáře. Vždy si přečtete název části a ihned vám bude uložen úkol – prostudovat jistou kapitolu ve volně dostupném studijním textu [10]; tam najdete motivace, vysvětlení, komentáře. Nenajdete tam však cvičení.

Samozřejmě by to chtělo mít nějakou sbírku úloh k procvičení probírané látky. Doporučení vám jistě dá váš vyučující. Nějaká cvičení jsou ve zmíněné Borůvkově knize. Já doporučím aspoň jednu sbírku, která je volně dostupná na internetu (teď a již delší dobu nazpátek), obsahuje úlohy vhodné obtížnosti, k některým úlohám dává návody k řešení a u většiny úloh řešení dokonce uvádí (na konci textu). Jde o sbírku

David Stanovský: *Příklady z algebry*
`https://www2.karlin.mff.cuni.cz/~stanovsk/vyuka/sbirka.pdf`

Mnozí jistě dobře umíte anglicky a třeba i rádi čtete anglické texty. Pro vás opakuji doporučení z úvodu. Mám na mysli následující dvě na internetu volně přístupné učebnice:

Thomas W. Judson: *Abstract Algebra: Theory and Applications*
`http://abstract.ups.edu/`
Učebnice obsahuje teorii a také širokou škálu cvičení. K vybraným cvičením jsou uvedeny návody a odpovědi.

Frederick M. Goodman: *Algebra: Abstract and Concrete*. Edition 2.6. Semi-Simple Press, Iowa City, IA, 2015.
`http://homepage.divms.uiowa.edu/~goodman/algebrabook.dir/algebrabook.html`
Opět, učebnice obsahuje teorii a také dostatek cvičení.

3.1 Základní pojmy teorie grup

3.1.1 Definice grupy

Úkol. Prostudujte kapitolu 1.1 Definice grupy v [10].

3.1.2 Mocniny

Úkol. Prostudujte kapitolu 1.2 Mocniny v [10].

3.1.3 Homomorfismy

Úkol. Prostudujte kapitolu 1.3 Homomorfismy v [10].

3.1.4 Podgrupy

Úkol. Prostudujte kapitolu 1.4 Podgrupy v [10].

3.1.5 Součiny grup

Úkol. Prostudujte kapitolu 1.5 Součiny grup v [10].

3.2 Příklady grup

3.2.1 Aditivní grupa okruhu

Úkol. Prostudujte kapitolu 2.1 Aditivní grupa okruhu v [10].

3.2.2 Grupa jednotek okruhu

Úkol. Prostudujte kapitolu 2.2 Grupa jednotek okruhu v [10].

3.2.3 Symetrická grupa

Úkol. Prostudujte kapitolu 2.3 Symetrická grupa v [10].

3.2.4 Alternující grupa

Úkol. Prostudujte kapitolu 2.4 Alternující grupa v [10].

3.2.5 Obecná lineární grupa

Úkol. Prostudujte kapitolu 2.5 Obecná lineární grupa v [10].

3.2.6 Grupa symetrií obrazce

Úkol. Prostudujte kapitolu 2.6 Grupa symetrií obrazce v [10].

3.2.7 Kvaterniony

Úkol. Prostudujte kapitolu 2.7 Kvaterniony v [10].

3.3 Lagrangeova věta a její důsledky

3.3.1 Lagrangeova věta

Úkol. Prostudujte kapitolu 3.1 Lagrangeova věta v [10].

3.3.2 Věty Fermatova a Eulerova

Úkol. Prostudujte kapitolu 3.2 Věty Fermatova a Eulerova v [10].

3.4 Cyklické grupy

3.4.1 Popis všech cyklických grup

Úkol. Prostudujte kapitolu 4.1 Popis všech cyklických grup v [10].

3.4.2 Podgrupy cyklických grup

Úkol. Prostudujte kapitolu 4.2 Podgrupy cyklických grup v [10].

4 Algebraické struktury

4.1 Definice algebraické struktury

4.1.1. Definice. Nechť A je neprázdná množina, n je kladné celé číslo. Zobrazení $* : A^n \to A$ se nazývá (n**–ární) operace** na množině A. Libovolný prvek $a \in A$ považujeme za 0–ární operaci na množině A. Místo 0–ární operace říkáme často nulární operace, místo 1–ární operace říkáme často unární

operace, místo 2–ární operace říkáme často binární operace (a, pokud nemůže dojít k nedorozumění, říkáme často pouze operace), místo 3–ární operace říkáme často ternární operace. Je – li $*$ binární operace na množině A, pak místo $*((x,y))$ píšeme většinou $x*y$ (pro libovolná $x,y \in A$). Je – li $'$ unární operace na množině A, pak místo $'(x)$ píšeme často x' či také $'x$ (pro libovolné $x \in A$).

4.1.2. Definice. Nechť $*$ a $\square$ jsou binární operace na množině A.

1. Říkáme, že operace $*$ je **asociativní**, pokud pro všechna $x,y,z \in A$ platí
$$x*(y*z) = (x*y)*z.$$

2. Říkáme, že operace $*$ je **komutativní**, pokud pro všechna $x,y \in A$ platí
$$x*y = y*x.$$

3. Říkáme, že operace $\square$ je **distributivní** vzhledem k operaci $*$, pokud pro všechna $x,y,z \in A$ platí
$$x\square(y*z) = (x\square y)*(x\square z),\ (y*z)\square x = (y\square x)*(z\square x).$$

4. Nechť $e \in A$. Říkáme, že e je **neutrální prvek** operace $*$, pokud pro všechna $x \in A$ platí
$$e*x = x,\ x*e = x.$$

5. Nechť $e,x,y \in A$, e je neutrální prvek operace $*$. Říkáme, že prvek y je **inverzní** (**inverze**) k prvku x vzhledem k operaci $*$, pokud platí
$$x*y = e,\ y*x = e.$$

4.1.3. Definice. Algebraická struktura (**algebra**) je uspořádaná $(n+1)$–tice $\mathcal{A} = (A, \circ_1, \circ_2, \ldots, \circ_n)$, kde A je neprázdná množina, zvaná **nosič** algebry $\mathcal{A}$, a $\circ_1, \circ_2, \ldots, \circ_n$ jsou operace na množině A. Často místo $\mathcal{A}$ píšeme A, tedy označením nerozlišujeme mezi algebraickou strukturou a jejím nosičem.

S jednou algebraickou strukturou jste se již poměrně podrobně seznámili v předchozí kapitole – samozřejmě mám na mysli grupy. V této souvislosti

chci upozornit, že často lze jednu strukturu formálně zapsat jako různé algebraické struktury. Například grupu můžeme chápat jako algebraickou strukturu $(G, \cdot)$, kde $\cdot$ je bimární operace na množině G, která je asociativní, má neutrální prvek a ke každému prvku existuje prvek inverzní (tento přístup byl zvolen v předchozí kapitole). Nebo lze grupu chápat jako algebraickou strukturu $(G, \cdot, ^{-1}, 1)$, kde $\cdot$ je binární operace na množině G, $^{-1}$ je unární operace na množině G, 1 je nulární operace na množině G a jsou splněny následující identity:

$$(x \cdot y) \cdot z = x \cdot (y \cdot z),\ 1 \cdot x = x,\ x \cdot 1 = x,\ x \cdot x^{-1} = 1,\ x^{-1} \cdot x = 1$$

Další možností je chápat grupu jako algebraickou strukturu $(G, \cdot, 1)$, kde $\cdot$ je binární operace na množině G, 1 je nulární operace na množině G, operace $\cdot$ je asociativní, má neutrální prvek 1 a pro každé $x \in G$ existuje $y \in G$ tak, že $x \cdot y = y \cdot x = 1$.

V dalších částech kapitoly stručně probereme některé další významné algebraické struktury.

4.2 Pologrupy a monoidy

4.2.1. Definice. Pologrupa $\mathcal{S} = (S, \cdot)$ je neprázdná množina S spolu s asociativní binární operací $\cdot$ (většinou tuto operaci nazýváme násobení). **Monoid** $\mathcal{M} = (S, \cdot, 1)$ je pologrupa $(S, \cdot)$ s neutrálním (jednotkovým) prvkem $1 \in S$ takovým, že $1 \cdot x = x \cdot 1 = x$ pro všechna $x \in S$. Pologrupa (monoid) se nazývá komutativní, pokud operace násobení je komutativní.

4.2.2. Příklady.

1. $(\mathbb{N}, +, 0)$, $(\mathbb{Z}, +, 0)$, $(\mathbb{Q}, +, 0)$, $(\mathbb{R}, +, 0)$, $(\mathbb{C}, +, 0)$, $(\mathbb{N}, \cdot, 1)$, $(\mathbb{Z}, \cdot, 1)$, $(\mathbb{Q}, \cdot, 1)$, $(\mathbb{R}, \cdot, 1)$, $(\mathbb{C}, \cdot, 1)$ jsou komutativní monoidy.

2. Nechť n je kladné celé číslo. Množinu všech čtvercových matic stupně n nad $\mathbb{Z}$ (tj. prvky matic jsou celá čísla) označme $M_n(\mathbb{Z})$. Symbolem O značíme nulovou matici, tj. matici, která má všechny prvky rovny 0. Symbolem E značíme jednotkovou matici, tj. matici splňující $e_{ii} = 1$ pro všechna $i \in \{1, \ldots, n\}$ a $e_{ij} = 0$ pro všechna $i, j \in \{1, \ldots, n\}$, $i \neq j$. Pak $(M_n(\mathbb{Z}), +, O)$ je komutativní monoid a $(M_n(\mathbb{Z}), \cdot, E)$ je monoid, který pro $n > 1$ není komutativní.

3. $(\mathbb{Z}^+, +)$ je příklad pologrupy, která není monoid, jelikož operace $+$ v $\mathbb{Z}^+$ nemá neutrální prvek.

Uvedeme teď důležirý příklad monoidu.

Abeceda je neprázdná množina.

Slovo nad abecedou Σ je konečná posloupnost prvků množiny Σ. Délku slova (posloupnosti) w značíme $|w|$, ε je prázdné slovo délky 0. Neprázdné slovo w délky n nad Σ můžeme také chápat jako zobrazení $w : \{1, \ldots, n\} \to \Sigma$, kde $w(i)$ je i–tý symbol slova w.

Σ^* označuje množinu všech slov nad Σ a Σ^+ označuje množinu všech neprázdných slov nad Σ.

Složením neprázdného slova u délky n_1 a neprázdného slova v délky n_2 je slovo w délky $n_1 + n_2$ označované $u \cdot v$ nebo pouze uv takové, že pro $1 \leq i \leq n_1$ je $w(i) = u(i)$ a pro $n_1 < i \leq n_1 + n_2$ je $w(i) = v(i - n_1)$. Dále pro každé slovo w klademe $\varepsilon w = w\varepsilon = w$.

Jestliže Σ je abeceda, pak každá podmnožina $L \subseteq \Sigma^*$ se nazývá **jazyk** nad abecedou Σ. Je tedy slovo konečná posloupnost prvků abecedy a jazyk je množina slov!

4.2.3. Příklad. Jestliže Σ je abeceda, pak $(\Sigma^*, \cdot, \varepsilon)$, kde $\cdot$ je skládání slov, je monoid, který v případě $|\Sigma| > 1$ není komutativní. Tento monoid se nazývá **volný monoid** nad abecedou Σ.

Homomorfismus monoidu $\mathcal{M}_1 = (S_1, \cdot_1, 1_1)$ do monoidu $\mathcal{M}_2 = (S_2, \cdot_2, 1_2)$ je zobrazení $\mu : S_1 \to S_2$ takové, že $\mu(1_1) = 1_2$ a $\mu(x \cdot_1 y) = \mu(x) \cdot_2 \mu(y)$ pro všechna $x, y \in S_1$.

Kongruence na monoidu $\mathcal{M} = (S, \cdot, 1)$ je relace ekvivalence (označme ji třeba $\equiv$) na množině S, která je kompatibilní s operací násobení, tedy

$$x_1 \equiv y_1 \wedge x_2 \equiv y_2 \Longrightarrow x_1 \cdot x_2 \equiv y_1 \cdot y_2$$

pro všechna $x_1, x_2, y_1, y_2 \in S$.

Cvičení.

1. Nechť n je celé číslo, $n > 1$. Dokažte, že monoid $(M_n(\mathbb{Z}), \cdot, E)$ není komutativní.

2. Nechť $\equiv$ je kongruence na monoidu $\mathcal{M} = (S, \cdot, 1)$. Třídu rozkladu $S/\equiv$, v níž leží prvek $a \in S$, označme $[a]_\equiv$. Dokažte, že množina $S/\equiv$ spolu s násobením definovaným předpisem $[a]_\equiv \cdot [b]_\equiv = [a \cdot b]_\equiv$ a s neutrálním prvkem $[1]_\equiv$ tvoří monoid, tzv. **faktorový monoid** $\mathcal{M}_\equiv$.

3. Nechť Σ je abeceda, $L \subseteq \Sigma^*$ je jazyk. Na množině Σ^* definujme relaci $\equiv_L$ takto: $u \equiv_L v$ tehdy a jen tehdy, když pro všechna $x, y \in \Sigma^*$ platí $xuy \in L \Leftrightarrow xvy \in L$. Dokažte, že $\equiv_L$ je kongruence na volném monoidu nad abecedou Σ. (Odpovídající faktorový monoid se nazývá **syntaktický monoid** jazyka L.)

4. Popište syntaktické monoidy pro následující jazyky L nad abecedou $\{0, 1\}$:

   - L je množina všech slov sudé délky
   - L je množina všech slov, která neobsahují dvě po sobě jdoucí nuly

   Co to znamená popsat syntaktický monoid jazyka L? Určíte třídy rozkladu $\{0, 1\}^*/\equiv_L$ a sestrojíte tabulku operace násobení v syntaktickém monoidu jazyka L.

4.3 Okruhy a tělesa

4.3.1. Definice. Algebraická struktura $(R, +, \cdot, 0)$ s binárními operacemi $+$ (sčítání) a $\cdot$ (násobení) a s nulární operací 0 je **okruh**, pokud $(R, +, 0)$ je komutativní grupa a násobení je distributivní vzhledem ke sčítání. Okruh se nazývá **asociativní** (**komutativní**, **s jednotkovým prvkem**), pokud operace násobení je asociativní (komutativní, má neutrální prvek).

Je – li $(R, +, \cdot, 0)$ okruh, má každý prvek $x \in R$ opačný prvek $-x$ v grupě $(R, +, 0)$, který splňuje $x + (-x) = 0$. Pro $x, y \in R$ místo $x + (-y)$ často píšeme $x - y$. V okruhu používáme úmluvy o prioritách operací, na které jsme zvyklí již ze základní a střední školy, tedy například zápis $x \cdot y + z$ znamená $(x \cdot y) + z$ a nikoli $x \cdot (y + z)$.

4.3.2. Příklad. $(\mathbb{Z}, +, \cdot, 0, 1)$ je asociativní komutativní okruh $(\mathbb{Z}, +, \cdot, 0)$ s jednotkovým prvkem 1. Nechť n je kladné celé číslo. Také $(\mathbb{Z}_n, +, \cdot, \overline{0}, \overline{1})$ je asociativní komutativní okruh s jednotkovým prvkem $\overline{1}$. Připomínám, že pro $a \in \mathbb{Z}$ je $\overline{a} = \{x \in \mathbb{Z} |\ x \equiv a\ (n)\}$; místo $x \equiv a\ (n)$ píšeme také $x \equiv a$

(mod n), místo $\overline{a}$ píšeme také $[a]$ či dokonce ještě přesněji $[a]_n$. Konstrukci okruhů $\mathbb{Z}_n$ si můžete připomenout například v části 1.2 studijního textu [9].

4.3.3. Definice. $(I, +, \cdot, 0)$ je **obor integrity**, pokud je to asociativní komutativní okruh, v němž pro všechna $x, y \in I$ platí

$$x \cdot y = 0 \Longrightarrow x = 0 \vee y = 0$$

4.3.4. Definice. $(T, +, \cdot, 0, 1)$ je **těleso**, pokud $0 \neq 1$, $(T, +, \cdot, 0)$ je asociativní okruh s jednotkovým prvkem 1 a pro každé $x \in T$, $x \neq 0$, existuje $y \in T$ tak, že $x \cdot y = y \cdot x = 1$. Prvek y se značí $\frac{1}{x}$ nebo x^{-1}. Značení je možno zavést, neboť prvek y je určen jednoznačně (nechť $x \cdot z = z \cdot x = 1$; pak $y = y \cdot 1 = y \cdot (x \cdot z) = (y \cdot x) \cdot z = 1 \cdot z = z$). Je-li v tělese násobení komutativní, pak hovoříme o komutativním tělese. Protože v tomto textu budeme pracovat výhradně s komutativními tělesy, budeme pro stručnost místo názvu komutativní těleso používat pouze slovo těleso.

4.3.5. Tvrzení. *V každém okruhu $(R, +, \cdot, 0)$ pro každý prvek $x \in R$ platí:*

$$x \cdot 0 = 0 \cdot x = 0$$

Důkaz. Buď $x \in R$. Počítejme:

$$0 + x \cdot 0 = x \cdot 0 = x \cdot (0 + 0) = x \cdot 0 + x \cdot 0$$

Je tedy $0 + x \cdot 0 = x \cdot 0 + x \cdot 0$. Jelikož v grupě $(R, +, 0)$ lze krátit (tedy $a + c = b + c \Rightarrow a = b$), dostáváme $0 = x \cdot 0$. Obdobně lze ukázat, že $0 \cdot x = 0$.

4.3.6. Tvrzení. *Každé těleso je obor integrity.*

Důkaz. Nechť $(T, +, \cdot, 0, 1)$ je těleso. Ukážeme, že $(T, +, \cdot, 0)$ je obor integrity. Z definice tělesa ihned plyne, že $(T, +, \cdot, 0)$ je asociativní komutativní okruh. Zbývá ukázat, že pro všechna $x, y \in T$ platí: $x \cdot y = 0 \Rightarrow x = 0 \vee y = 0$. Buďte tedy $x, y \in T$, $x \cdot y = 0$. Chceme: $x = 0$ nebo $y = 0$. Je – li $x = 0$, jsme hotovi. Nechť tedy $x \neq 0$. Musíme ukázat, že $y = 0$. Počítejme:

$$0 = x^{-1} \cdot 0 = x^{-1} \cdot (x \cdot y) = (x^{-1} \cdot x) \cdot y = 1 \cdot y = y$$

4.3.7. Poznámka. V definici tělesa požadujeme, aby $0 \neq 1$. Tento požadavek je ekvivalentní s požadavkem, aby těleso mělo aspoň dva prvky.

- Předpokládejme, že $0 \neq 1$. Chceme: T má aspoň dva prvky. To jistě platí, jelikož $0, 1 \in T$.
- Předpokládejme, že T má aspoň dva prvky. Chceme: $0 \neq 1$. Postupujme sporem. Předpokládejme, že $0 = 1$. Zvolme libovolné $x \in T$. Je $x = x \cdot 1 = x \cdot 0 = 0$, $x = 0$. Takže každý prvek $x \in T$ je roven 0 a T má tedy pouze jeden prvek. To je spor. Nutně tedy $0 \neq 1$.

4.3.8. Příklad. $(\mathbb{Q}, +, \cdot, 0, 1)$, $(\mathbb{R}, +, \cdot, 0, 1)$ a $(\mathbb{C}, +, \cdot, 0, 1)$ jsou tělesa. Máme také nekonečně mnoho příkladů konečných těles – pro každé prvočíslo p totiž $(\mathbb{Z}_p, +, \cdot, \overline{0}, \overline{1})$ je těleso. Dokonce pro každé celé číslo n, $n > 1$, platí: $(\mathbb{Z}_n, +, \cdot, \overline{0}, \overline{1})$ je těleso právě tehdy, když n je prvočíslo (důkaz najdete například v [9], tvrzení 1.2.21).

Konečná tělesa hrají velkou roli v aplikacích. Je tomu tak například v kódování, proto v knihách o kódování ([1], [3]) najdeme hodně informací o konečných tělesech.

Položme si nyní třeba otázku, kolik prvků může mít konečné těleso. Víme již, že pro každé prvočíslo p existuje těleso s p prvky. Může mít konečné těleso 50 prvků? Uvidíme, že nikoli.

Připomeňme značení, s nímž jste se seznámili v kapitole o grupách. Jestliže G je grupa, $a \in G$, n je celé číslo a používáme multiplikativní symboliku, tedy máme grupu $(G, \cdot, 1)$, pak n–tou mocninu prvku a značíme a^n. Jestliže používáme aditivní symboliku, tedy máme grupu $(G, +, 0)$, pak místo a^n píšeme na. Jistě také znáte základní pravidla:

Nechť G je grupa, $a, b \in G$, k, l jsou celá čísla. Pak

$$a^k a^l = a^{k+l}, \ (a^k)^l = a^{kl}$$

neboli, zapsáno v aditivní symbolice,

$$(ka) + (la) = (k + l)a, \ l(ka) = (kl)a$$

Jestliže grupa G je komutativní, pak také

$$a^k b^k = (ab)^k$$

neboli, zapsáno v aditivní symbolice,

$$(ka) + (kb) = k(a + b)$$

4.3.9. Definice. Nechť $(T, +, \cdot, 0, 1)$ je těleso. **Charakteristika** tělesa T je řád prvku 1 v grupě $(T, +, 0)$. Chakteristiku tělesa T značíme $char(T)$.

4.3.10. Tvrzení. *Charakteristika konečného tělesa je prvočíslo.*

Důkaz. Nechť $(T, +, \cdot, 0, 1)$ je konečné těleso. Grupa $(T, +, 0)$ je konečná a tedy každý její prvek má konečný řád. Takže také 1 má konečný řád a $char(T) = n$, kde n je kladné celé číslo. Specielně $n1 = 0$. Jelikož $0 \neq 1$, je $n > 1$. Chceme: n je prvočíslo. Předpokládejme naopak, že n je číslo složené. Pak $n = kl$ pro nějaká celá čísla k, l, $1 < k < n$, $1 < l < n$. Počítejme:

$$\begin{aligned}
(k1) \cdot (l1) &= (\underbrace{1 + \cdots + 1}_{k}) \cdot (\underbrace{1 + \cdots + 1}_{l}) \\
&= \underbrace{(\underbrace{1 + \cdots + 1}_{k}) \cdot 1 + \cdots + (\underbrace{1 + \cdots + 1}_{k}) \cdot 1}_{l} \\
&= \underbrace{(\underbrace{1 + \cdots + 1}_{k}) + \cdots + (\underbrace{1 + \cdots + 1}_{k})}_{l} \\
&= \underbrace{1 + \cdots + 1}_{kl} \\
&= \underbrace{1 + \cdots + 1}_{n} \\
&= n1 \\
&= 0
\end{aligned}$$

Je tedy $(k1) \cdot (l1) = 0$. Protože každé těleso je obor integrity, je $k1 = 0$ nebo $l1 = 0$. Nechť například $k1 = 0$. Protože k je kladné celé číslo a n je řád prvku 1 v grupě $(T, +, 0)$, je $n \leq k$. To je spor. Obdobně také dostaneme spor v případě $l1 = 0$. Nutně tedy n je prvočíslo.

4.3.11. Tvrzení. *Nechť $(T, +, \cdot, 0, 1)$ je konečné těleso, $char(T) = p$. Pro každé $v \in T$ je $pv = 0$.*

Důkaz. Nechť $v \in T$. Počítejme:

$$\begin{aligned} pv &= \underbrace{v + \cdots + v}_{p} \\ &= v \cdot (\underbrace{1 + \cdots + 1}_{p}) \\ &= v \cdot (p1) \\ &= v \cdot 0 \\ &= 0 \end{aligned}$$

Nyní dokážeme tvrzení o počtu prvků konečného tělesa. V důkazu využijete své znalosti základů lineární algebry.

4.3.12. Věta. *Počet prvků konečného tělesa je roven p^n, kde p je prvočíslo a n je kladné celé číslo.*

Důkaz. Nechť $(T, +, \cdot, 0, 1)$ je konečné těleso. Dle tvrzení 4.3.10 je $char(T) = p$ pro nějaké prvočíslo p. $(T, +, 0)$ je komutativní grupa. Pro každé $\overline{k} \in \mathbb{Z}_p$ (k je celé číslo) a každé $v \in T$ definujeme prvek $\overline{k} \cdot v \in T$ takto:

$$\overline{k} \cdot v = kv$$

Ukážeme, že definice je korektní. Nechť l je celé číslo, $\overline{k} = \overline{l}$. Chceme: $kv = lv$. Je $k \equiv l \pmod{p}$, $p/k - l$, $k - l = mp$ pro nějaké celé číslo m. Počítejme:

$$\begin{aligned} kv &= (l + mp)v \\ &= (lv) + ((mp)v) \\ &= (lv) + (p(mv)) \\ &= (lv) + 0 \\ &= lv \end{aligned}$$

Dokážeme teď čtyři věci:

1. pro všechna $k, l \in \mathbb{Z}$, pro všechna $u \in T$: $(\overline{k} \cdot \overline{l}) \cdot u = \overline{k} \cdot (\overline{l} \cdot u)$

   Počítejme:

   $$(\overline{k} \cdot \overline{l}) \cdot u = \overline{kl} \cdot u = (kl)u = k(lu) = k(\overline{l} \cdot u) = \overline{k} \cdot (\overline{l} \cdot u)$$

2. pro všechna $k, l \in \mathbb{Z}$, pro všechna $u \in T$: $(\overline{k} + \overline{l}) \cdot u = (\overline{k} \cdot u) + (\overline{l} \cdot u)$
   Počítejme:
   $$(\overline{k} + \overline{l}) \cdot u = \overline{k+l} \cdot u = (k+l)u = (ku) + (lu) = (\overline{k} \cdot u) + (\overline{l} \cdot u)$$

3. pro všechna $k \in \mathbb{Z}$, pro všechna $u, v \in T$: $\overline{k} \cdot (u + v) = (\overline{k} \cdot u) + (\overline{k} \cdot v)$
   Počítejme:
   $$\overline{k} \cdot (u + v) = k(u + v) = (ku) + (kv) = (\overline{k} \cdot u) + (\overline{k} \cdot v)$$

4. pro všechna $u \in T$: $\overline{1} \cdot u = u$
   Počítejme:
   $$\overline{1} \cdot u = 1u = u$$

Právě jsme ukázali, že T je vektorový prostor nad tělesem $\mathbb{Z}_p$. Protože množina T je konečná, je T vektorový prostor konečné dimenze, označme ji n. Je $n \in \mathbb{Z}$, $n \geq 0$. Ovšem $n = 0$ by znamenalo, že prázdná množina je báze vektorového prostoru T a $T = \{0\}$. To by byl spor, protože každé těleso má aspoň dva prvky. Je tedy $n > 0$. Pak vektorový prostor T je isomorfní s prostorem $\mathbb{Z}_p^n$ (viz [9], důsledek 2.8.3) a tedy

$$|T| = |\mathbb{Z}_p^n| = |\mathbb{Z}_p|^n = p^n$$

Ukázali jsme, že těleso T má p^n prvků, kde $p = char(T)$ je prvočíslo a n je kladné celé číslo.

4.3.13. Důsledek (důkazu). *Nechť T je konečné těleso, $char(T) = p$. Pak T je vektorový prostor nad tělesem $\mathbb{Z}_p$ a počet prvků tělesa T je roven p^n, kde n je dimenze vektorového prostoru T nad $\mathbb{Z}_p$.*

Teď je již jasné, že žádné těleso nemá 50 prvků. A existuje těleso, které má 49 prvků? Věta 4.3.12 existenci takového tělesa nevylučuje, ale také ji nedokazuje. Významným poznatkem je, že pro každé prvočíslo p a každé kladné celé číslo n existuje těleso mající p^n prvků. Toto tvrzení dokazovat nebudeme. Lze ho dokázat různě, velmi pěkným důkazem je konstruktivní důkaz uvedený například v knize [3] – těleso s p^n prvky se sestrojí pomocí ireducibilního polynomu stupně n nad tělesem $\mathbb{Z}_p$.

Cvičení.

1. Nechť $(R, +, \cdot, 0)$ je okruh. Dokažte, že pro všechna $x, y, z \in R$ platí:
   - $(-x) \cdot y = x \cdot (-y) = -(x \cdot y),\ (-x) \cdot (-y) = x \cdot y$
   - $x \cdot (y - z) = x \cdot y - x \cdot z,\ (x - y) \cdot z = x \cdot z - y \cdot z$

2. - Uveďte příklad nekonečného asociativního komutativního okruhu, který není obor integrity.
   - Uveďte příklad konečného asociativního komutativního okruhu, který není obor integrity.

3. - Uveďte příklad nekonečného oboru integrity s jednotkovým prvkem, který není těleso.
   - Dokažte, že každý konečný aspoň dvouprvkový obor integrity s jednotkovým prvkem je těleso.

4. Nechť n je kladné celé číslo, $(T, +, \cdot, 0, 1)$ je těleso.. Množinu všech čtvercových matic stupně n nad T (tj. prvky matic jsou prvky tělesa T) označme $M_n(T)$. Symbolem O značíme nulovou matici, tj. matici, která má všechny prvky rovny 0. Symbolem E značíme jednotkovou matici, tj. matici splňující $e_{ii} = 1$ pro všechna $i \in \{1, \ldots, n\}$ a $e_{ij} = 0$ pro všechna $i, j \in \{1, \ldots, n\}$, $i \neq j$. Dokažte, že $(M_n(T), +, \cdot, O)$ je asociativní okruh s jednotkovým prvkem E. Dokažte, že pro $n > 1$ okruh $M_n(T)$ není obor integrity a není komutativní.

5. Celé číslo d se nazývá bezčtvercové, pokud pro všechna kladná celá čísla e platí: $e^2/d \Rightarrow e = 1$. Nechť d je bezčtvercové celé číslo, $d \neq 1$. Položme $\mathbb{Q}(\sqrt{d}) = \{a + b\sqrt{d} |\ a, b \in \mathbb{Q}\}$. Dokažte, že $(\mathbb{Q}(\sqrt{d}), +, \cdot, 0, 1)$ je těleso. Dokažte aspoň pro $d = 2$. Poznámka: těleso $\mathbb{Q}(\sqrt{d})$ se nazývá **kvadratické těleso**.

6. Nechť $(T, +, \cdot, 0, 1)$ je těleso. Dokažte, že $(T - \{0\}, \cdot, 1)$ je komutativní grupa.

7. Předpokládejte, že T je čtyřprvkové těleso. Sestrojte tabulky operací sčítání a násobení v tělese T.

8. Nechť I je obor integrity, $x, y, z \in I$. Jestliže $xy = xz$ a $x \neq 0$, pak $y = z$. Dokažte.

4.4 Algebra polynomů

Jistě jste se již všichni setkali s polynomy, tedy s objekty jako $x^2 + 2x + 3$ atd. Pro aplikace není ani tak důležité, jak se formálně přesně polynomy definují (sestrojují), ale důležité je umět s nimi správně manipulovat (sčítat, násobit,...) a znát související terminologii (stupeň polynomu,...). Pojďme si udělat přehled toho nejdůležitějšího – bude to pro vás snad opakování (připomenutí) obvyklých věcí jen s tím rozdílem, že zvolíme obecnější přístup - koeficienty polynomů nebudou pouze čísla (jak jste asi byli doposud zvyklí), ale budou to moci být prvky libovolného komutativního asociativního okruhu s jednotkovým prvkem, specielně tedy oboru integrity s jednotkovým prvkem nebo tělesa.

Nechť tedy R je komutativní asociativní okruh s jednotkovýnm prvkem. **Formální mocninná řada nad okruhem** R je posloupnost $a = (a_0, a_1, a_2, \dots)$ taková, že $a_i \in R$ pro všechna $i = 0, 1, 2, \dots$. Je zvykem psát

$$a = a_0 + a_1 x + a_2 x^2 + \cdots = \sum_{i=0}^{\infty} a_i x^i = \sum_{0 \le i} a_i x^i$$

Množinu všech formálních mocninných řad nad okruhem R označíme $R[[x]]$.

Jedno důležité upozornění: Formální mocninná řada nad okruhem R je posloupnost prvků okruhu R, nikoli nějaký nekonečný součet – otázky typu konvergence nás tedy vůbec nezajímají.

Prvky množiny $R[[x]]$ sčítáme a násobíme. Formální mocninné řady se sčítají po složkách:

$$(a_0, a_1, a_2, \dots) + (b_0, b_1, b_2, \dots) = (a_0 + b_0, a_1 + b_1, a_2 + b_2, \dots)$$

Je ihned vidět (rozmyslete si to!), že $(R[[x]], +)$ je komutativní grupa s neutrálním prvkem $(0, 0, 0, \dots)$, kde 0 je nulový prvek okruhu R (tj. neutrální prvek operace $+$ v R). Přitom $-(a_0, a_1, a_2, \dots) = (-a_0, -a_1, -a_2, \dots)$, kde $-a_i$ je opačný prvek k prvku a_i v grupě $(R, +)$.

Formální mocninné řady se násobí tak, jak jste zvyklí násobit polynomy. Nechť

$$(a_0, a_1, a_2, \dots) \cdot (b_0, b_1, b_2, \dots) = (c_0, c_1, c_2, \dots)$$

Pak pro každé $i \in \mathbb{Z}$, $i \ge 0$, je

$$c_i = \sum_{0 \le j, 0 \le k, j+k=i} a_j b_k = a_0 b_i + a_1 b_{i-1} + a_2 b_{i-2} + \cdots + a_{i-1} b_1 + a_i b_0$$

Násobení formálních mocninných řad je komutativní: Nechť

$$(b_0, b_1, b_2, \dots) \cdot (a_0, a_1, a_2, \dots) = (d_0, d_1, d_2, \dots)$$

Ukážeme, že pro každé $i \in \mathbb{Z}$, $0 \leq i$, je $c_i = d_i$. Počítejme:

$$\begin{aligned} c_i &= \sum_{0 \leq j, 0 \leq k, j+k=i} a_j b_k \\ &= \sum_{0 \leq j, 0 \leq k, j+k=i} b_k a_j \\ &= \sum_{0 \leq k, 0 \leq j, k+j=i} b_j a_k \\ &= \sum_{0 \leq j, 0 \leq k, j+k=i} b_j a_k \\ &= d_i \end{aligned}$$

Ukážeme teď, že operace násobení formálních mocninných řad má neutrální prvek $(1, 0, 0, 0, \dots)$, kde 1 je jednotkový prvek okruhu R (tj. neutrální prvek operace $\cdot$ v R) a 0 je nulový prvek okruhu R. Položme $e_0 = 1$, $e_i = 0$ pro $i \in \mathbb{Z}$, $i \geq 1$, $e = (e_0, e_1, e_2, \dots)$. Nechť $ea = (f_0, f_1, f_2, \dots)$. Uvidíme, že pro každé $i \in \mathbb{Z}$, $i \geq 0$, je $f_i = a_i$. Tím bude dokázáno, že $ea = a$ pro libovolnou formální mocninnou řadu a. Tudíž $e = (1, 0, 0, \dots)$ je neutrální prvek operace násobení formálních mocninných řad. Počítejme:

$$\begin{aligned} f_i &= \sum_{0 \leq j, 0 \leq k, j+k=i} e_j a_k \\ &= e_0 a_i + e_1 a_{i-1} + \cdots + e_i a_0 \\ &= 1 \cdot a_i + 0 \cdot a_{i-1} + \cdots + 0 \cdot a_0 \\ &= a_i + 0 + \cdots + 0 \\ &= a_i \end{aligned}$$

Samostatně dokažte, že násobení formálních mocninných řad je asociativní (viz cvičení).

Dokážeme ještě, že operace násobení formálních mocninných řad nad okruhem R je distrubutivní vzhledem k operaci sčítání formálních mocninných řad nad R. Uvažme tedy mocninné řady $a, b, c \in R[[x]]$

$$a = (a_0, a_1, a_2, \dots), b = (b_0, b_1, b_2, \dots), c = (c_0, c_1, c_2, \dots)$$

Nechť

$$ab = u = (u_0, u_1, u_2, \dots), ac = v = (v_0, v_1, v_2 \dots), u+v = w = (w_0, w_1, w_2, \dots)$$

Nechť

$$b + c = f = (f_0, f_1, f_2, \dots), af = g = (g_0, g_1, g_2 \dots)$$

Ukážeme, že pro všechna $i \in \mathbb{Z}$, $i \geq 0$, je $g_i = w_i$. To bude znamenat, že $g = w$, $af = u + v$, $a(b + c) = ab + ac$.

Počítejme:

$$\begin{aligned} g_i &= \sum_{0\leq j, 0\leq k, j+k=i} a_j f_k \\ &= \sum_{0\leq j, 0\leq k, j+k=i} a_j (b_k + c_k) \\ &= \sum_{0\leq j, 0\leq k, j+k=i} a_j b_k + a_j c_k \\ &= \sum_{0\leq j, 0\leq k, j+k=i} a_j b_k + \sum_{0\leq j, 0\leq k, j+k=i} a_j c_k \\ &= u_i + v_i \\ &= w_i \end{aligned}$$

Shrňme to, co jsme již o formálních mocninných řadách nad R dokázali: $(R[[x]], +)$ je komutativní grupa, operace násobení v $R[[x]]$ je komutativní, asociativní, má jednotkový prvek a je distributivní vzhledem ke sčítání. Dokázali jsme tedy následující větu:

4.4.1. Věta. *Jestliže R je komutativní asociativní okruh s jednotkovým prvkem , pak $R[[x]]$ je komutativní asociativní okruh s jednotkovým prvkem.*

Nyní konečně řekneme, co je to polynom nad R. Nechť $a \in R[[x]]$, $a = (a_0, a_1, a_2, \dots)$. Formální mocninná řada a se nazývá **polynom nad okruhem** R, pokud posloupnost a má pouze konečně mnoho nenulových členů, tedy pokud množina

$$\{i \in \mathbb{N} |\ a_i \neq 0\}$$

je konečná.

Množinu všech polynomů nad okruhem R budeme značit $R[x]$.

Polynom $a = (a_0, a_1, a_2, \dots, a_n, 0, 0, 0, \dots)$ (tedy $a_i = 0$ pro $i > n$) je zvykem zapisovat ve tvaru

$$a = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = \sum_{i=0}^{n} a_i x^i = \sum_{0 \le i \le n} a_i x^i$$

I zde je dobré uvést důležité upozornění: Polynom nad R je posloupnost prvků okruhu R, nikoli nějaká funkce (i když za chvíli si řekneme, že polynom určuje funkci (zobrazení) R do R, nelze ho s tím zobrazením ztotožňovat – uvidíme, že různé polynomy mohou určovat tutéž funkci).

4.4.2. Tvrzení. *Nechť R je komutativní asociativní okruh s jednotkovým prvkem, f a g jsou polynomy nad R. Pak $-f$, $f+g$ a fg jsou polynomy nad R.*

Důkaz. Jako cvičení zkuste samostatně dokázat, že $-f$ a $f + g$ jsou polynomy nad R. Důkaz je opravdu snadný a jistě jej zvládnete udělat. My zde dokážeme, že fg je polynom nad R. Nechť $f = (f_0, f_1, f_2, \dots)$, $g = (g_0, g_1, g_2, \dots)$, $fg = h = (h_0, h_1, h_2, \dots)$. Protože f, g jsou polynomy, existuje $n \in \mathbb{Z}$, $0 \le n$, tak, že pro všechna $i \in \mathbb{Z}$, $i > n$ je $f_i = g_i = 0$. Ukážeme, že pro každé $i \in \mathbb{Z}$, $i > 2n$, je $h_i = 0$. Z toho pak již vyplývá, že $h = fg$ je polynom. Buď tedy $i > 2n$. Je

$$h_i = \sum_{0 \le j, 0 \le k, j+k=i} f_j g_k$$

Nechť $j, k \in \mathbb{Z}$, $0 \le j$, $0 \le k$, $j + k = i$ (přičemž $i > 2n$). Předpokládejme, že $j \le n$ a $k \le n$. Pak $j + k \le 2n$, $i \le 2n$, spor. Nutně tedy $j > n$ nebo $k > n$. Pak $f_j = 0$ nebo $g_k = 0$, $f_j g_k = 0$, a tudíž

$$h_i = \sum_{0 \le j, 0 \le k, j+k=i} f_j g_k = \sum_{0 \le j, 0 \le k, j+k=i} 0 = 0$$

4.4.3. Věta. *Jestliže R je komutativní asociativní okruh s jednotkovým prvkem, pak $R[x]$ je komutativní asociativní okruh s jednotkovým prvkem.*

Důkaz. Tvrzení věty plyne z 4.4.1 a 4.4.2. Důležité také je, že $(0, 0, 0, \dots)$ a $(1, 0, 0, \dots)$ jsou polynomy.

Striktně vzato, prvky okruhu R nejsou polynomy nad R, jelikož jeden prvek okruhu R není (nekonečnou) posloupností prvků okruhu R. Uvidíme však, že je rozumné považovat prvky okruhu R také za polynomy nad R.

Pro každé $r \in R$ je $(r, 0, 0, \dots)$ polynom nad R.

Pro všechna $r, s \in R$ máme

$$\begin{aligned}(r,0,0,\dots) + (s,0,0,\dots) &= (r+s,0,0,\dots)\\ (r,0,0,\dots) \cdot (s,0,0,\dots) &= (r \cdot s,0,0,\dots)\end{aligned}$$

Nechť $M = \{(r, 0, 0, \dots) |\ r \in R\} \subseteq R[x]$. Struktura $(M, +, \cdot)$ se chová stejně jako okruh $(R, +, \cdot)$ – stačí zapomenout na závorky, čárky, nuly (na levé straně je součet (součin) v M, na pravé straně je součet (součin) v R). Proto je rozumné polynom $(r, 0, 0, \dots)$ ztotožnit s prvkem r. Po tomto ztotožnění se prvky okruhu R stávají polynomy nad R; tyto polynomy se nazývají **konstantní polynomy**. Přitom 0 je neutrální prvek operace $+$ v R i v $R[x]$, 1 je neutrální prvek operace $\cdot$ v R i v $R[x]$.

Udělejme si teď přehled základních pojmů vztahujících se k jednomu polynomu.

Buď $a = (a_0, a_1, a_2, \dots)$ polynom nad okruhem R.

- Prvky a_i $(i = 0, 1, 2, \dots)$ se nazývají **koeficienty** polynomu a; specielně a_i je koeficient u x^i.
- a_0 se nazývá **absolutní člen** polynomu a.
- Předpokládejme, že $a \neq 0$. Množina $\{i \in \mathbb{N} |\ a_i \neq 0\}$ je neprázdná (protože $a \neq 0$) a shora omezená (protože a je polynom a má tedy pouze konečný počet nenulových koeficientů). Číslo

  $$\max\{i \in \mathbb{N} |\ a_i \neq 0\}$$

  se nazývá **stupeň** polynomu a; označení: $deg(a)$. Poznámka: Pro nulový polynom není stupeň definován.
- Nechť a je nenulový polynom, $deg(a) = d$. Koeficient a_d se nazývá **vedoucí koeficient** polynomu a; označení: $lc(a)$.
- Nechť a je nenulový polynom, $d = deg(a)$. Polynom a nejčastěji zapisujeme ve tvaru

  $$a = a_0 + a_1 x + a_2 x^2 + \dots + a_d x^d$$

nebo také

$$a(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d$$

- Říkáme, že nenulový polynom a je **monický polynom**, pokud $lc(a) = 1$.

4.4.4. Věta. *Nechť I je obor integrity s jednotkovým prvkem. Nechť a, b jsou nenulové polynomy nad I. Pak platí:*

1. *$a + b = 0$ nebo $a + b \neq 0$, $deg(a + b) \leq \max\{deg(a), deg(b)\}$*
2. *$ab \neq 0$, $deg(ab) = deg(a) + deg(b)$, $lc(ab) = lc(a)lc(b)$*

Důkaz.

1. Důkaz udělejte jako cvičení.
2. Nechť $a = (a_0, a_1, a_2 \dots)$, $b = (b_0, b_1, b_2, \dots)$, $ab = c = (c_0, c_1, c_2, \dots)$, $deg(a) = s$, $deg(b) = t$.

   - Buď $i > s + t$. Ukážeme, že $c_i = 0$.

     $$c_i = \sum_{0 \leq j, 0 \leq k, j+k=i} a_j b_k$$

     Nechť $j, k \in \mathbb{Z}$, $0 \leq j$, $0 \leq k$, $j + k = i$. Předpokládejme, že $j \leq s$ a $k \leq t$. Pak $j + k \leq s + t$, $i \leq s + t$, spor. Nutně tedy $j > s$ nebo $k > t$. Pak $a_j = 0$ nebo $b_k = 0$, $a_j b_k = 0$, $c_i = \sum_{0 \leq j, 0 \leq k, j+k=i} 0 = 0$, $c_i = 0$.
   - Ukážeme, že $c_{s+t} = a_s b_t$.

     $$c_{s+t} = \sum_{0 \leq j, 0 \leq k, j+k=s+t} a_j b_k$$

     Nechť $j, k \in \mathbb{Z}$, $0 \leq j$, $0 \leq k$, $j + k = s + t$. Pro $j > s$ nebo $k > t$ je $a_j b_k = 0$. Nechť $j \leq s$ a $k \leq t$. Jelikož $j + k = s + t$, je $j = s$, $k = t$. Je tedy

     $$c_{s+t} = a_s b_t$$

Jelikož $a_s \in I$, $b_t \in I$, $a_s \neq 0$, $b_t \neq 0$, I je obor integrity, je $a_s b_t \neq 0$, $c_{s+t} \neq 0$. Dále již víme, že $c_i = 0$ pro $i > s+t$, takže $deg(c) = s+t = deg(a) + deg(b)$, $deg(ab) = deg(a) + deg(b)$. Také $lc(c) = c_{s+t} = a_s b_t = lc(a)lc(b)$, $lc(ab) = lc(a)lc(b)$.

4.4.5. Důsledek. *Jestliže I je obor integrity s jednotkovým prvkem, pak $I[x]$ je obor integrity s jednotkovým prvkem.*

Důkaz. Tvrzení plyne z 4.4.3 a 4.4.4 (rozmyslete si jak!).

V závěru kapitoly se podívejme na chápání polynomů jakožto funkcí. Na to jste asi zvyklí ze školy (střední). Třeba jste říkali, že $x^2 + 2x + 3$ je kvadratická funkce a kreslili jste její graf atd.

Nechť R je komutativní asociativní okruh s jednotkovým prvkem, f je polynom nad R a $c \in R$. Dále, nechť

$$f = f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_n x^n$$

Klademe

$$f(c) = f_0 + f_1 c + f_2 c^2 + \cdots + f_n c^n$$

Výpočty se provádějí v okruhu R. Zřejmě $f(c) \in R$; $f(c)$ nazýváme **hodnota polynomu** f v (bodě) c. Nyní lze tedy f chápat také jako zobrazení R do R, tj. $f : R \to R$.

Zde je potřeba dát velký pozor! Různé polynomy nad R mohou určovat tutéž funkci! Například polynomy x^3 a x nad $\mathbb{Z}_3$ určují tutéž funkci $\mathbb{Z}_3$ na $\mathbb{Z}_3$ – prověřte to. Přitom se evidentně jedná o různé polynomy – první je kubický, druhý je lineární. Zdůrazněme, že algebraické chápání polynomu nad R je primárně chápání polynomu jako posloupnosti prvků okruhu R.

Dobrovolný úkol. Prostudujte formálně přesné zavedení polynomů – například v [5], Kapitola XI, či v [7]. Výhodou druhé publikace je snadná a legální dostupnost na internetu v Czech Digital Mathematics Library (což mimochodem je výborná věc a měli byste ji využívat).

Cvičení.

1. Nechť R je komutativní asociativní okruh s jednotkovým prvkem. Dokažte, že operace násobení formálních mocninných řad nad okruhem R je asociativní, tj. dokažte, že pro všechna $a, b, c \in R[[x]]$ platí

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

2. Nechť R je komutativní asociativní okruh s jednotkovým prvkem. Nechť f a g jsou polynomy nad R. Dokažte, že $-f$ a $f+g$ jsou polynomy nad R.

3. Nechť R je komutativní asociativní okruh s jednotkovým prvkem. Nechť $r, s \in R$. Dokažte, že

$$(r, 0, 0, \dots) \cdot (s, 0, 0, \dots) = (r \cdot s, 0, 0, \dots)$$

4. Nechť I je obor integrity s jednotkovým prvkem. Nechť a, b jsou nenulové polynomy nad I. Pak platí:

$$a + b = 0 \text{ nebo } a + b \neq 0, deg(a + b) \leq \max\{deg(a), deg(b)\}$$

Dokažte.

4.5 Euklidův algoritmus pro polynomy

V kapitole o celých číslech jsme zkoumali dělitelnost – zabývali jsme se věcmi jako dělení se zbytkem, největší společný dělitel, Euklidův algoritmus. Uvidíte teď, že paralelně můžeme zkoumat dělitelnost ve struktuře polynomů nad tělesem. Domnívám se, že v této kapitole budete ve studiu postupovat rychle, protože pojmy, výpočty, důkazy budou většinou analogické pojmům, výpočtům, důkazům, které jste již dělali v oboru celých čísel (jen místo sčítání a násobení čísel budete sčítat a násobit polynomy).

4.5.1. Definice. Nechť I je obor integrity s jednotkovým prvkem, $a, b \in I[x]$. Říkáme, že a **dělí** b (a je dělitelem b, b je dělitelné a, b je násobkem a), a píšeme a/b, pokud existuje $q \in I[x]$ tak, že $b = qa$.

4.5.2. Poznámka. Všimněte si, že pro všechna $a \in I[x]$, $a/0$ ($0 = 0 \cdot a$). Avšak pouze 0 je násobkem polynomu 0.

4.5.3. Tvrzení. *Nechť I je obor integrity s jednotkovým prvkem. Nechť $a, b, c, \beta, \gamma \in I[x]$. Jestliže a/b a a/c, pak $a/\beta b + \gamma c$.*

Důkaz tvrzení je jednoduchý, jistě jej dokážete udělat sami. Zkuste to!

4.5.4. Věta (algoritmus dělení). *Nechť T je těleso. Nechť $a, b \in T[x], a \neq 0$. Pak existují polynomy q, r nad tělesem T takové, že*

$$b = aq + r,\ r = 0 \vee (r \neq 0 \wedge (deg(r) < deg(a)))$$

Polynomy q, r jsou určeny jednoznačně.

Důkaz. Nejprve dokážeme jednoznačnost.

Nechť q_1, r_1, q_2, r_2 jsou polynomy nad tělesem T takové, že

$$b = aq_i + r_i,\ r_i = 0 \vee (r_i \neq 0 \wedge (deg(r_i) < deg(a)))$$

pro $i = 1, 2$. Pak

$$\begin{aligned} aq_1 + r_1 &= aq_2 + r_2 \\ aq_1 - aq_2 &= r_2 - r_1 \\ a(q_1 - q_2) &= r_2 - r_1 \end{aligned}$$

Předpokládejme, že $r_1 \neq r_2$. Pak $r_2 - r_1 \neq 0$, $q_1 - q_2 \neq 0$, $deg(r_2 - r_1) = deg(a(q_1 - q_2)) = deg(a) + deg(q_1 - q_2) \geq deg(a)$, $deg(a) \leq deg(r_2 - r_1)$. Mohou nastat tři případy, všechny však dají spor. Z toho pak vyplyne, že $r_1 = r_2$.

- $r_1 \neq 0$, $r_2 \neq 0$: $deg(r_2 - r_1) \leq \max\{deg(r_1), deg(r_2)\} < deg(a)$, $deg(r_2 - r_1) < deg(a)$, spor
- $r_1 \neq 0$, $r_2 = 0$: $deg(r_2 - r_1) = deg(0 - r_1) = deg(-r_1) = deg(r_1) < deg(a)$, $deg(r_2 - r_1) < deg(a)$, spor
- $r_1 = 0$, $r_2 \neq 0$: $deg(r_2 - r_1) = deg(r_2 - 0) = deg(r_2) < deg(a)$, $deg(r_2 - r_1) < deg(a)$, spor

Ukázali jsme, že $r_1 = r_2$. Pak $r_2 - r_1 = 0$, $a(q_1 - q_2) = 0$; jelikož $a \neq 0$, je $q_1 - q_2 = 0$, $q_1 = q_2$.

Nyní dokážeme existenci.

Jestliže $b = 0$, položíme $q = 0$, $r = b$.

Nechť nyní $b \neq 0$. Postupujme indukcí vzhledem k $deg(b)$. Jestliže $deg(b) < deg(a)$, položíme $q = 0$, $r = b$. Nechť $deg(b) \geq deg(a)$. Položme

$$b^* = b - lc(b)lc(a)^{-1}x^{deg(b)-deg(a)}a$$

Rozlišíme dva případy:

- $b^* = 0$: Je
$$b = lc(b)lc(a)^{-1}x^{deg(b)-deg(a)}a$$
a vezmeme $q = lc(b)lc(a)^{-1}x^{deg(b)-deg(a)}$, $r = 0$.

- $b^* \neq 0$: Protože polynom $lc(b)lc(a)^{-1}x^{deg(b)-deg(a)}a$ má stupeň $deg(b)$ a vedoucí koeficient $lc(b)$, je $deg(b^*) < deg(b)$. Dle indukčního předpokladu existují polynomy q^*, r^* nad tělesem T takové, že

$$b^* = aq^* + r^*,\ r^* = 0 \vee (r^* \neq 0 \wedge (deg(r^*) < deg(a)))$$

Máme

$$\begin{aligned} b &= b^* + lc(b)lc(a)^{-1}x^{deg(b)-deg(a)}a \\ &= aq^* + r^* + lc(b)lc(a)^{-1}x^{deg(b)-deg(a)}a \\ &= a(lc(b)lc(a)^{-1}x^{deg(b)-deg(a)} + q^*) + r^* \end{aligned}$$

a vezmeme $q = lc(b)lc(a)^{-1}x^{deg(b)-deg(a)} + q^*$, $r = r^*$.

Polynom q se nazývá **(neúplný) podíl** a polynom r se nazývá **zbytek** při dělení polynomu b polynomem a (označení: $r = b \bmod a$).

4.5.5. Poznámka.

1. Věta 4.5.4 si opravdu zaslouží název "algoritmus dělení", protože její důkaz je vlastně rekurzivním algoritmem pro výpočet podílu a zbytku. Pro nenulový polynom b vypočítáme polynom b^*, který je nulový (a v tom případě algoritmus končí) nebo je nenulový stupně menšího než je stupeň polynomu b a na polynom b^* aplikujeme týž postup (rekurze). Po konečně mnoha krocích musí výpočet skončit – kdyby vždy bylo $b^* \neq 0$, dostávali bychom neustále polynomy menšího a menšího stupně.

2. Možná jste se ve škole učili dělit polynom polynomem se zbytkem. Rozmyslete si, že jste postupovali stejně jako zde uvedený algoritmus pro dělení polynomů se zbytkem. Rozdíl byl samozřejmě v tom, že ve škole jste pracovali s polynomy s racionálními, reálnými či maximálně komplexními koeficienty, kdežto my teď uvažujeme polynomy nad libovolným tělesem.

3. Proč ve větě 4.5.4 pracujeme s polynomy nad tělesem? Algoritmus pracuje s prvkem $lc(a)^{-1}$ a jeho existence v tělese je zaručena. Na druhé straně, pro žádný jiný prvek inverzi algoritmus nepotřebuje. Lze tedy

algoritmus dělení se zbytkem aplikovat i tehdy, pokud I je obor integrity s jednotkovým prvkem, $a, b \in I[x]$, $a \neq 0$, vedoucí koeficient polynomu a je invertibilním prvkem oboru integrity I (což je specielně splněno vždy, když polynom a je monický). Takže například polynomem $x^3 - 2x^2 + 3x - 4$ lze se zbytkem vydělit libovolný polynom ze $\mathbb{Z}[x]$ (a podíl a zbytek budou mít všechny koeficienty celočíselné!).

4.5.6. Poznámka. Nechť T je těleso. Nechť $a, b \in T[x]$, $a \neq 0$. Pak

1. $a/b - b \bmod a$
2. $b = 0 \vee (b \neq 0 \wedge deg(b) < deg(a)) \iff b \bmod a = b$

4.5.7. Věta. *Nechť T je těleso. Nechť $a, b \in T[x]$, $a \neq 0$ nebo $b \neq 0$. Pak existuje jediný monický polynom $d \in T[x]$ takový, že*

1. $d/a \wedge d/b$
2. $(\forall c \in T[x])\ c/a \wedge c/b \Longrightarrow c/d$

Důkaz. Nejdříve dokážeme jednoznačnost.
Nechť $d_i \in T[x]$, d_i je monický,

1. $d_i/a \wedge d_i/b$
2. $(\forall c \in T[x])\ c/a \wedge c/b \Longrightarrow c/d_i$

pro $i = 1, 2$. Chceme: $d_1 = d_2$.
Informace

1. $d_1/a \wedge d_1/b$
2. $(\forall c \in T[x])\ c/a \wedge c/b \Longrightarrow c/d_2$

dává d_1/d_2,

1. $d_2/a \wedge d_2/b$
2. $(\forall c \in T[x])\ c/a \wedge c/b \Longrightarrow c/d_1$

dává d_2/d_1.

Celkem tedy d_1/d_2 a d_2/d_1 a jelikož polynomy d_1, d_2 jsou monické, je $d_1 = d_2$ (viz cvičení).

Nyní dokážeme existenci.

Položme

$$D = \{sa + tb |\ s \in T[x],\ t \in T[x],\ sa + tb \neq 0\}$$

Jestliže $a \neq 0$, pak $a = 1a + 0b \in D$. Jestliže $b \neq 0$, pak $b = 0a + 1b \in D$. Jelikož $a \neq 0$ nebo $b \neq 0$, je $D \neq \emptyset$. Také $D \subseteq T[x]$, $0 \notin D$.

Buď e polynom nejmenšího stupně v D, tedy $e \in D$ a pro všechna $f \in D$ je $deg(e) \leq deg(f)$. Takových polynomů e může být víc, my si vezmeme jeden z nich.

Položme $d = lc(e)^{-1}e$. Pak d je polynom nad tělesem T, d je monický, $deg(d) = deg(e)$, pro všechna $f \in D$ je $deg(d) \leq deg(f)$. Ukážeme, že

1. $d/a \wedge d/b$
2. $(\forall c \in T[x])\ c/a \wedge c/b \Longrightarrow c/d$

Protože $e \in D$, existují polynomy $s, t \in T[x]$, $e = sa + tb$. Je

$$d = lc(e)^{-1}e = lc(e)^{-1}(sa + tb) = lc(e)^{-1}sa + lc(e)^{-1}tb$$

Položme $u = lc(e)^{-1}s$, $v = lc(e)^{-1}t$. Pak $u, v \in T[x]$, $d = ua + vb$.

- Ukážeme, že d/a.

  Existují $q, r \in T[x]$, $a = dq + r$, $r = 0$ nebo $r \neq 0$, $deg(r) < deg(d)$. Předpokládejme, že $r \neq 0$. Je tedy $deg(r) < deg(d)$. Počítejme:

$$\begin{aligned} a &= dq + r \\ a &= (ua + vb)q + r \\ a - (ua + vb)q &= r \\ a - uaq - vbq &= r \\ (1 - uq)a + (-vq)b &= r \end{aligned}$$

  Protože $1 - uq \in T[x]$, $-vq \in T[x]$, $r \neq 0$, je $r \in D$. Pak ovšem $deg(d) \leq deg(r)$, spor. Nutně tedy $r = 0$, $a = dq$, d/a.

  Obdobně lze ukázat, že d/b.

- Nechť $c \in T[x]$, c/a, c/b. Ukážeme, že c/d.

  Stačí si připomenout, že $d = ua + vb$, kde $u, v \in T[x]$.

4.5.8. Definice. Polynom d z věty 4.5.7 se nazývá **největší společný dělitel** polynomů a a b; píšeme $d = GCD(a, b)$.

4.5.9. Důsledek (důkazu). *Nechť T je těleso, $a, b \in T[x]$, $a \neq 0$ nebo $b \neq 0$. Pak existují $u, v \in T[x]$ takové, že*

$$GCD(a, b) = ua + vb$$

4.5.10. Poznámka. Nechť T je těleso. Nechť a je polynom nad tělesem T, $a \neq 0$. Pak $GCD(a, 0) = lc(a)^{-1}a$.

Euklidův algoritmus (pro polynomy)

VSTUP: $a, b \in T[x]$ (T je těleso), $a \neq 0$, $b \neq 0$

VÝSTUP: $GCD(a, b)$

$$(a_1, a_2) \longleftarrow (a, b)$$

KONEC: $a_2 = 0$

Je-li $a_2 \neq 0$, provedeme iteraci:

$$a_1 = qa_2 + r,\ q, r \in T[x],\ r = 0 \vee (r \neq 0 \wedge deg(r) < deg(a_2))$$

$$(a_1, a_2) \mapsto (\overline{a}_1, \overline{a}_2),\ \overline{a}_1 = a_2,\ \overline{a}_2 = r$$

Jelikož $\overline{a}_2 = 0$ nebo $\overline{a}_2 \neq 0$, $deg(\overline{a_2}) < deg(a_2)$, výpočet skončí po konečně mnoha krocích. Během výpočtu stále je $a_1 \neq 0$.

Nechť $d \in T[x]$. Platí:

$$d/a_1 \wedge d/a_2 \Longleftrightarrow d/a_2 \wedge d/r$$

Zdůvodnění:

1. Předpokládejme, že $d/a_1 \wedge d/a_2$. Chceme: $d/a_2 \wedge d/r$. Je jasné, že d/a_2. Je $r = a_1 + (-q)a_2$. Protože d/a_1 a d/a_2, máme d/r.

2. Předpokládejme, že $d/a_2 \wedge d/r$. Chceme: $d/a_1 \wedge d/a_2$. Je jasné, že d/a_2. Je $a_1 = qa_2 + r$. Protože d/a_2 a d/r, máme d/a_1.

Tudíž:

$$GCD(a_1, a_2) = GCD(a_2, r) = GCD(\overline{a}_1, \overline{a}_2)$$

Průběh výpočtu můžeme zapsat následovně:

$$(a, b) \mapsto \cdots \mapsto (\alpha, 0)$$

Je $GCD(a, b) = GCD(\alpha, 0) = lc(\alpha)^{-1}\alpha$ (je $\alpha \neq 0$),

$$\boxed{GCD(a, b) = lc(\alpha)^{-1}\alpha}$$

4.5.11. Příklad. Pomocí Euklidova algoritmu vypočteme $GCD(x^2+x, x^2+2)$, kde $x^2 + x$ a $x^2 + 2$ jsou polynomy nad tělesem $\mathbb{Z}_3$.

$$(x^2 + x, x^2 + 2) \mapsto (x^2 + 2, x + 1) \mapsto (x + 1, 0)$$

Závěr: $GCD(x^2 + x, x^2 + 2) = x + 1$.

Rozšířený Euklidův algoritmus (pro polynomy)

VSTUP: $a, b \in T[x]$ (T je těleso), $a \neq 0$, $b \neq 0$

VÝSTUP: $GCD(a, b)$, $s, t \in T[x]$, $GCD(a, b) = sa + tb$

$$(a_1, a_2) \longleftarrow (a, b),\ (p_1, p_2) \longleftarrow (0, 1),\ (q_1, q_2) \longleftarrow (1, 0)$$

KONEC: $a_2 = 0$

Je-li $a_2 \neq 0$, provedeme iteraci:

$$a_1 = qa_2 + r,\ q, r \in T[x],\ r = 0 \vee (r \neq 0 \wedge deg(r) < deg(a_2))$$

$$\begin{array}{l} (a_1, a_2) \mapsto (\overline{a}_1, \overline{a}_2), \quad \overline{a}_1 = a_2, \quad \overline{a}_2 = r \\ (p_1, p_2) \mapsto (\overline{p}_1, \overline{p}_2), \quad \overline{p}_1 = p_2, \quad \overline{p}_2 = p_1 + qp_2 \\ (q_1, q_2) \mapsto (\overline{q}_1, \overline{q}_2), \quad \overline{q}_1 = q_2, \quad \overline{q}_2 = q_1 + qq_2 \end{array}$$

Víme již, že výpočet skončí po konečně mnoha krocích (viz Euklidův algoritmus pro polynomy).

$$\begin{aligned}
\overline{p}_2\overline{a}_1 + \overline{p}_1\overline{a}_2 &= (p_1 + qp_2)a_2 + p_2 r \\
&= p_1 a_2 + qp_2 a_2 + p_2 r \\
&= p_1 a_2 + p_2(qa_2 + r) \\
&= p_1 a_2 + p_2 a_1 \\
&= p_2 a_1 + p_1 a_2
\end{aligned}$$

Ukázali jsme, že během výpočtu algoritmu stále platí

$$\overline{p}_2\overline{a}_1 + \overline{p}_1\overline{a}_2 = p_2 a_1 + p_1 a_2$$

$$\begin{aligned}
\overline{q}_2\overline{a}_1 + \overline{q}_1\overline{a}_2 &= (q_1 + qq_2)a_2 + q_2 r \\
&= q_1 a_2 + qq_2 a_2 + q_2 r \\
&= q_1 a_2 + q_2(qa_2 + r) \\
&= q_1 a_2 + q_2 a_1 \\
&= q_2 a_1 + q_1 a_2
\end{aligned}$$

Ukázali jsme, že během výpočtu algoritmu stále platí

$$\overline{q}_2\overline{a}_1 + \overline{q}_1\overline{a}_2 = q_2 a_1 + q_1 a_2$$

$$\begin{aligned}
\overline{q}_2\overline{p}_1 - \overline{q}_1\overline{p}_2 &= (q_1 + qq_2)p_2 - q_2(p_1 + qp_2) \\
&= q_1 p_2 + qq_2 p_2 - q_2 p_1 - q_2 qp_2 \\
&= q_1 p_2 - q_2 p_1 \\
&= -(q_2 p_1 - q_1 p_2)
\end{aligned}$$

Ukázali jsme, že během výpočtu algoritmu stále platí

$$\overline{q}_2\overline{p}_1 - \overline{q}_1\overline{p}_2 = -(q_2 p_1 - q_1 p_2)$$

Během výpočtu se tedy zachovávají následující invarianty:

1.

$$\overline{p}_2\overline{a}_1 + \overline{p}_1\overline{a}_2 = p_2a_1 + p_1a_2$$

2.

$$\overline{q}_2\overline{a}_1 + \overline{q}_1\overline{a}_2 = q_2a_1 + q_1a_2$$

3.

$$\overline{q}_2\overline{p}_1 - \overline{q}_1\overline{p}_2 = -(q_2p_1 - q_1p_2)$$

Průběh výpočtu můžeme zapsat následovně:

$$\begin{array}{llll} (a,b) & \mapsto_1 & \ldots & \mapsto_n \quad (\alpha,0) \\ (0,1) & \mapsto & \ldots & \mapsto \quad (\beta_1,\beta_2) \\ (1,0) & \mapsto & \ldots & \mapsto \quad (\gamma_1,\gamma_2) \end{array}$$

Víme již, že

$$\boxed{GCD(a,b) = lc(\alpha)^{-1}\alpha}$$

(viz Euklidův algoritmus pro polynomy).
Invariant 1 dává

$$\begin{aligned} \beta_2 \cdot \alpha + \beta_1 \cdot 0 &= 1 \cdot a + 0 \cdot b \\ \beta_2\alpha &= a \end{aligned}$$

Invariant 2 dává

$$\begin{aligned} \gamma_2 \cdot \alpha + \gamma_1 \cdot 0 &= 0 \cdot a + 1 \cdot b \\ \gamma_2\alpha &= b \end{aligned}$$

Invariant 3 dává

$$
\begin{aligned}
\gamma_2\beta_1 - \gamma_1\beta_2 &= (0\cdot 0 - 1\cdot 1)\cdot(-1)^n \\
\gamma_2\beta_1 - \gamma_1\beta_2 &= (-1)^{n+1} \\
\gamma_2\beta_1\alpha - \gamma_1\beta_2\alpha &= (-1)^{n+1}\cdot\alpha \\
\beta_1(\gamma_2\alpha) - \gamma_1(\beta_2\alpha) &= (-1)^{n+1}\cdot\alpha \\
\beta_1 b - \gamma_1 a &= (-1)^{n+1}\cdot\alpha \\
(-\gamma_1)a + \beta_1 b &= (-1)^{n+1}\cdot\alpha \\
(-1)^{n+1}\cdot(-\gamma_1)a + (-1)^{n+1}\cdot\beta_1 b &= (-1)^{n+1}\cdot(-1)^{n+1}\cdot\alpha \\
(-1)^n\cdot\gamma_1 a + (-1)^{n+1}\cdot\beta_1 b &= \alpha \\
(-1)^n\cdot lc(\alpha)^{-1}\gamma_1 a + (-1)^{n+1}\cdot lc(\alpha)^{-1}\beta_1 b &= lc(\alpha)^{-1}\alpha \\
((-1)^n\cdot lc(\alpha)^{-1}\gamma_1)a + ((-1)^{n+1}\cdot lc(\alpha)^{-1}\beta_1)b &= GCD(a,b)
\end{aligned}
$$

Je tedy

$$\boxed{s = (-1)^n\cdot lc(\alpha)^{-1}\gamma_1,\ t = (-1)^{n+1}\cdot lc(\alpha)^{-1}\beta_1}$$

4.5.12. Příklad. Aplikujeme rozšířený Euklidův algoritmus na vstup $a = x^2+x$, $b = x^2+2$ (polynomy nad tělesem $\mathbb{Z}_3$).

$$
\begin{array}{ccccccc}
(x^2+x, x^2+2) & \mapsto_1 & (x^2+2, x+1) & \mapsto_2 & (x+1, 0) \\
(0,1) & \mapsto & (1,1) & \mapsto & (1,x) \\
(1,0) & \mapsto & (0,1) & \mapsto & (1,x+2)
\end{array}
$$

Pomocné výpočty:

$$
\begin{aligned}
x^2+x &= 1\cdot(x^2+2) + (x+1) \\
x^2+2 &= (x+2)\cdot(x+1) + 0
\end{aligned}
$$

Závěr: $GCD(x^2+x, x^2+2) = x+1$, $s = 1$, $t = 2$, $1\cdot(x^2+x)+2\cdot(x^2+2) = x+1$.

Cvičení.

1. Nechť I je obor integrity s jednotkovým prvkem. Jestliže f, g jsou monické polynomy nad oborem integrity I, f/g a g/f, pak $f = g$. Dokažte.

2. Najděte největší společný dělitel polynomů x^2+x+1 a $x+1$ nad tělesem $\mathbb{Z}_2$. Najděte polynomy s, t nad tělesem $\mathbb{Z}_2$ takové, že $GCD(x^2 + x + 1, x + 1) = s(x^2 + x + 1) + t(x + 1)$.

3. Najděte největší společný dělitel polynomů x^2+x+1 a $x+1$ nad tělesem $\mathbb{Z}_3$. Najděte polynomy s, t nad tělesem $\mathbb{Z}_3$ takové, že $GCD(x^2 + x + 1, x + 1) = s(x^2 + x + 1) + t(x + 1)$.

5 Samoopravné kódy

V této části se budeme zabývat pouze základními binárními kódy. Náš výklad je založen především na kapitole 1 knihy [3], jejímž autorem je Elwyn R. Berlekamp, který byl profesorem matematiky a informatiky na Kalifornské universitě v Berkeley. Berlekamp byl široce známý pro svou práci v oblasti informatiky, teorie kódování a kombinatorické teorie her. Z české literatury doporučuji knihu [1] od Jiřího Adámka, v níž také najdete hodně úloh k procvičení.

5.1 Opakovací kódy a kódy kontroly parity

Předpokládejme, že chceme posílat binární posloupnosti (tj. posloupnosti nul a jedniček) kanálem se šumem. Pokud pošleme jedničku, jednička pravděpodobně bude přijata; pokud pošleme nulu, nula pravděpodobně bude přijata. Avšak občas šum kanálu způsobí, že poslaná jednička bude chybně interpretována jako nula nebo poslaná nula bude chybně interpretována jako jednička. Přestože nejsme schopni kanálu zabránit ve způsobování takových chyb, můžeme jejich nežádoucí účinky omezit pomocí kódování. Základní idea je jednoduchá. Vezmeme posloupnost k číslic zprávy, kterou chceme poslat, připojíme k nim r kontrolních číslic a pošleme kanálem celý blok $n = k + r$ číslic. Za předpokladu, že šum kanálu změní dostatečně málo z těchto n poslaných číslic, může r kontrolních číslic poskytnout přijímači (příjemci) dostatečné informace, které mu umožní detekovat a opravovat chyby kanálu.

Vysílač musí mít nějaké pravidlo pro výběr r kontrolních číslic, je-li dána posloupnost k číslic zprávy. Tomu pravidlu (předpisu) se říká *kódování*. Jaká-

koli posloupnost n číslic, kterou může kodér vyslat, se nazývá *kódové slovo*. Ačkoli existuje celkem 2^n různých binárních posloupností délky n, pouze 2^k z těchto posloupností jsou kódová slova, protože r kontrolních číslic v jakémkoli kódovém slově je jednoznačně určeno k číslicemi zprávy. Soubor skládající se z těchto 2^k kódových slov délky n se nazývá *kód*.

Bez ohledu na to, které kódové slovo je posláno, může být přijata jakákoli z 2^n binárních posloupností délky n, pokud je v kanálu dostatečně velký šum. Podle n přijatých číslic se musí dekodér pokusit rozhodnout, které z 2^k možných kódových slov bylo vysláno.

Mezi nejjednodušší příklady binárních kódů patří *opakovací kódy*, kde $k = 1$, r je libovolné kladné celé číslo a $n = k + r = r + 1$. Kód obsahuje dvě kódová slova, posloupnost n nul a posloupnost n jedniček. První číslici můžeme nazvat *číslice zprávy* (*informační číslice*); ostatních r číslic pak *kontrolní číslice*. Hodnota každé kontrolní číslice v opakovacím kódu je totožná s hodnotou číslice zprávy. Dekodér může používat následující pravidlo: Spočítejte počet nul a počet jedniček v přijaté posloupnosti. Pokud je přijato více nul než jedniček, rozhodněte, že bylo odesláno kódové slovo složené ze samých nul; pokud je více jedniček než nul, rozhodněte, že bylo odesláno kódové slovo složené ze samých jedniček. Pokud se počet nul rovná počtu jedniček, nerozhodujte.

Je zřejmé, že toto dekódovací pravidlo bude dekódovat správně ve všech případech, kdy šum kanálu změní méně než polovinu číslic v bloku. Pokud šum kanálu změní přesně polovinu číslic v bloku, dekodér se dopustí *selhání*, tj. nebude dekódovat. Pokud kanál změní více než polovinu číslic v bloku, dekodér se dopustí *chyby*, tj. dekóduje přijaté slovo jako špatné (chybné) kódové slovo. Pokud se chyby kanálu vyskytují zřídka, pravděpodobnost selhání dekódování nebo chyby dekódování pro opakovací kód s velkým r je jistě velmi malá.

Ačkoli považujeme selhání dekódování za vhodnější než chybu dekódování, upřednostnili bychom správné dekódování před kteroukoli z obou těchto variant. Samozřejmě je často možné provést nějaké přesuny mezi chybami dekódování a selháním dekódování modifikací dekódovacího algoritmu. Uvažujme například binární opakovací kód s délkou bloků $n = 5$. Nejprve předpokládejme, že všechny přijaté posloupnosti obsahující 0, 1 nebo 2 jedničky jsou dekódovány jako kódové slovo složené ze samých nul a všechny přijaté posloupnosti obsahující 3, 4 nebo 5 jedniček jsou dekódovány jako kódové slovo složené ze samých jedniček. Tento dekódovací algoritmus dekóduje každé možné přijaté slovo jako jedno z možných kódových slov; je to *úplný* dekó-

dovací algoritmus (*úplné* dekódování). Úplné dekódovací algoritmy nemohou při dekódování selhat. Můžeme však použít alternativní, *neúplný* (*částečný*) dekódovací algoritmus pro stejný binární opakovací kód s délkou bloků $n = 5$. Můžeme například dekódovat všechny přijaté posloupnosti obsahující 0 nebo 1 jedničku jako kódové slovo složené ze samých nul a všechny posloupnosti obsahující 4 nebo 5 jedniček jako kódové slovo složené ze samých jedniček. Tento algoritmus nedokáže dekódovat posloupnosti obsahující 2 nebo 3 jedničky. Přestože má tento neúplný dekódovací algoritmus kladnou pravděpodobnost selhání dekódování, má neúplný algoritmus podstatně nižší pravděpodobnost chyby dekódování.

V některých aplikacích se chyba dekódování rovná katastrofě. Může to například vést k tomu, že raketa nebo vesmírná loď obdrží nesprávný příkaz. Na druhou stranu selhání dekódování může mít za následek pouze ignorování příkazu. To může představovat jen drobnou nepříjemnost, kterou lze překonat pouhým opakováním příkazu. V takových aplikacích se preferuje velmi neúplný dekódovací algoritmus, který záměrně odmítá dekódovat jakékoli dostatečně nejednoznačné přijaté slovo.

V jiných aplikacích však nemusí být možné opakovat nedekódované zprávy. V takových případech je selhání dekódování téměř stejně závažnou katastrofou jako chyba dekódování. V aplikacích tohoto typu je žádoucí maximalizovat pravděpodobnost správného dekódování. I když informace obsažená v přijatém slově může být dosti neprůkazná, úplný dekódovací algoritmus se nikdy nevzdá. Bez ohledu na to, jaká posloupnost je přijata, rozhodne se pro určité kódové slovo, i když jeho rozhodnutí je pouze inteligentní odhad.

Pro mnohé kódy je velmi obtížné rozšířit známé dobré neúplné dekódovací algoritmy na dobré úplné dekódovací algoritmy. Avšak obvykle je velmi jednoduché získat dobré neúplné dekódovací algoritmy z dobrého úplného dekódovacího algoritmu.

Viděli jsme, že lze bez obtíží formulovat úplný dekódovací algoritmus pro binární opakovací kódy. Pokud délka bloku je dostatečně velká, pravděpodobnost chyby dekódování je velmi malá. Avšak tyto kódy mají velmi nízkou *ryclost přenosu informace* (*informační poměr*)

$$R = \frac{k}{n}$$

V opakovacím kódu jsou totiž všechny číslice, až na jednu, kontrolní. Obvykle se více zajímáme o kódy, které mají vyšší rychlost přenosu informace.

Extrémním příkladem takových kódů s vysokou rychlostí přenosu jsou *kódy celkové kontroly parity*, které obsahují pouze jednu kontrolní číslici. Tato kontrolní číslice je rovna součtu $n-1$ informačních číslic, přičemž sčítáme modulo 2, tedy sčítáme v $\mathbb{Z}_2$. Z toho plyne, že v každém kódovém slově kódu celkové kontroly parity je sudý počet jedniček. Pokud přijaté slovo obsahuje sudý počet jedniček, dekodér ho dekóduje beze změny (tj. jako přijaté slovo), avšak pokud přijaté slovo obsahuje lichý počet jedniček, dekodér ho nebude dekódovat. Toto neúplné dekódovací pravidlo bude dekódovat správně pouze tehdy, když se v přijaté zprávě nevyskytuje žádná chyba. Jakýkoli lichý počet chyb povede k selhání dekódování. Jakýkoli nenulový sudý počet chyb způsobí chybu dekódování.

Tyto dva příklady, opakovací kódy a kódy kontroly parity, jsou extrémní, relativně triviální příklady binárních kódů. Opakovací kódy mají obrovskou schopnost opravy chyb, ale pouze jednu číslici zprávy na blok. Kódy kontroly parity mají velmi vysokou rychlost přenosu informace, ale protože obsahují pouze jednu kontrolní číslici na blok, nejsou schopny udělat více než detekovat lichý počet chyb. Abychom interpolovali mezi těmito dvěma extrémními třídami kódů a nalezli kódy, které mají průměrnou rychlost přenosu a průměrnou schopnost korekce chyb, vezmeme v úvahu obecnější třídu lineárních kódů, jejichž speciálními případy jsou opakovací kódy a kódy kontroly parity.

5.2 Lineární kódy

V kódu obsahujícím několik číslic zprávy a několik kontrolních číslic musí být každá kontrolní číslice nějakou funkcí číslic zprávy (tj. musí jimi být jednoznačně určena). V jednoduchém případě kódu celkové kontroly parity je jediná kontrolní číslice rovna binárnímu součtu všech číslic zprávy. Pokud chceme mít několik kontrol parity, je rozumné nastavit každou kontrolní číslici jako binární součet nějaké podmnožiny číslic zprávy. Například sestrojíme binární kód s bloky délky $n = 6$ mající $k = 3$ číslice zprávy a $r = 3$ kontrolní číslice. Tři číslice zprávy označíme c_1, c_2 a c_3 a tři kontrolní číslice označíme c_4, c_5 a c_6. Tyto kontrolní číslice určíme podle následujících pravidel:

$$\begin{array}{lllllll} c_4 & = & c_1 & + & c_2 & & \\ c_5 & = & c_1 & & & + & c_3 \\ c_6 & = & & & c_2 & + & c_3 \end{array}$$

nebo, v maticové notaci,

$$\begin{pmatrix} c_4 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Celé kódové slovo sestává z číslic c_1, c_2, c_3, c_4, c_5, c_6. Každé kódové slovo musí splňovat následující rovnice kontroly parity:

$$\begin{array}{ccccccccccc} c_1 & + & c_2 & & & + & c_4 & & & & = & 0 \\ c_1 & & & + & c_3 & & & + & c_5 & & = & 0 \\ & & c_2 & + & c_3 & + & & & & + \; c_6 & = & 0 \end{array}$$

nebo, v maticové notaci,

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \mathbf{c}^T = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Zde $\mathbf{c}^T$ označuje sloupcový vektor, který je transpozicí řádkového vektoru

$$\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6)$$

Ještě stručněji můžeme tuto rovnici napsat jako

$$\mathcal{H}\mathbf{c}^T = \mathbf{0}$$

kde $\mathbf{0}$ je třírozměrný sloupcový vektor, jehož složkami jsou samé nuly, a $\mathcal{H}$ je *kontrolní matice*

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

Kódových slov je $2^3 = 8$ a jsou to tato slova:

000000, 001011, 010101, 011110, 100110, 101101, 110011, 111000

Poté, co je zpráva zakódována, je kódové slovo posláno kanálem se šumem. Kanál k němu přičte *chybové slovo* (*slovo šumu*) $\mathbf{e} = (e_1, e_2, e_3, e_4, e_5, e_6)$ (zde i jinde v této kapitole "slovo"znamená totéž co "vektor"), kde

$e_i = 0$ pokud kanál nezmění i – tou číslici
$e_i = 1$ pokud kanál změní i – tou číslici

Přijaté slovo je posloupnost $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5, r_6)$, kde $r_i = c_i + e_i$, nebo, ve vektorové notaci, $\mathbf{r} = \mathbf{c} + \mathbf{e}$. Dekodér začne výpočtem *syndromu*, který je definován rovnicí

$$\mathbf{s}^T = \mathcal{H}\mathbf{r}^T$$

V našem příkladě tato rovnice vypadá takto:

$$\begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \mathbf{r}^T$$

Aby dekodér dekódoval, musí nakonec odpovědět na otázku: "Jaké kódové slovo $\mathbf{c}$ bylo vysláno?"Ukazuje se však, že je snazší nejprve odpovědět na otázku: "Jaké bylo chybové slovo $\mathbf{e}$?"Pokud je dekodér schopen najít správný chybový vektor $\mathbf{e}$, pak $\mathbf{c}$ lze získat ze vzorce $\mathbf{c} = \mathbf{r} - \mathbf{e}$. (V binárním případě, tedy v $\mathbb{Z}_2$, samozřejmě $+$ je $-$, rozdíl je součet a $\mathbf{c} = \mathbf{r} - \mathbf{e}$ znamená $\mathbf{c} = \mathbf{r} + \mathbf{e}$.)

Jestliže $\mathbf{r}$ je přijaté slovo, pak množina možných chybových slov je právě množina vektorů, které mají týž syndrom jako $\mathbf{r}$. Proč?

- Předpokládejme, že $\mathbf{r} = \mathbf{c} + \mathbf{e}$ pro nějaké kódové slovo $\mathbf{c}$. Chceme: $\mathcal{H}\mathbf{e}^T = \mathcal{H}\mathbf{r}^T$.
  Počítejme: $\mathcal{H}\mathbf{e}^T = \mathcal{H}(\mathbf{r} - \mathbf{c})^T = \mathcal{H}(\mathbf{r}^T - \mathbf{c}^T) = \mathcal{H}\mathbf{r}^T - \mathcal{H}\mathbf{c}^T = \mathcal{H}\mathbf{r}^T - \mathbf{0} = \mathcal{H}\mathbf{r}^T$, $\mathcal{H}\mathbf{e}^T = \mathcal{H}\mathbf{r}^T$.

- Předpokládejme, že $\mathcal{H}\mathbf{e}^T = \mathcal{H}\mathbf{r}^T$. Chceme: $\mathbf{r} = \mathbf{c} + \mathbf{e}$ pro nějaké kódové slovo $\mathbf{c}$.
  Je $\mathbf{r} = (\mathbf{r} - \mathbf{e}) + \mathbf{e}$. Stačí tedy ukázat, že $\mathbf{r} - \mathbf{e}$ je kódové slovo. Je $\mathcal{H}(\mathbf{r} - \mathbf{e})^T = \mathcal{H}(\mathbf{r}^T - \mathbf{e}^T) = \mathcal{H}\mathbf{r}^T - \mathcal{H}\mathbf{e}^T = \mathbf{0}$, $\mathcal{H}(\mathbf{r} - \mathbf{e})^T = \mathbf{0}$, $\mathbf{r} - \mathbf{e}$ je kódové slovo.

Předpokládejme například, že přijaté slovo je kódové slovo. Pak $\mathcal{H}\mathbf{r}^T = \mathbf{0}$ a $\mathcal{H}\mathbf{e}^T = \mathbf{0}$, takže chybové slovo je také kódové slovo.

Množina všech n – rozměrných binárních vektorů (tedy prvků vektorového prostoru $\mathbb{Z}_2^n$), které mají nulový syndrom, je přesně množina všech kódových slov (tedy kód). Nulový vektor je kódové slovo; je – li $\mathbf{x}$ kódové slovo a $k \in \mathbb{Z}_2$, pak také $k\mathbf{x}$ je kódové slovo, jelikož $\mathcal{H}(k\mathbf{x})^T = \mathcal{H}(k\mathbf{x}^T) = k(\mathcal{H}\mathbf{x}^T) = k\mathbf{0} = \mathbf{0}$; jsou – li $\mathbf{x}$ a $\mathbf{y}$ kódová slova, pak také $\mathbf{x} + \mathbf{y}$ je kódové slovo, jelikož $\mathcal{H}(\mathbf{x}+\mathbf{y})^T = \mathcal{H}\mathbf{x}^T + \mathcal{H}\mathbf{y}^T = \mathbf{0}+\mathbf{0} = \mathbf{0}$. Množina všech kódových slov (tj. kód) tvoří tedy podprostor vektorového prostoru $\mathbb{Z}_2^n$. Říkáme proto, že množina všech kódových slov tvoří *lineární kód*.

Na množině $\mathbb{Z}_2^n$ definujeme binární relaci $\sim$ takto:

$$\mathbf{x} \sim \mathbf{y} \iff \mathcal{H}\mathbf{x}^T = \mathcal{H}\mathbf{y}^T$$

Snadno lze ukázat, že relace $\sim$ je ekvivalence na množině $\mathbb{Z}_2^n$ (dokažte to!). Tato ekvivalence určuje rozklad $\mathbb{Z}_2^n/\sim$ a každou třídu tohoto rozkladu nazýváme *třída* podle kódu. Dva n – rozměrné binární vektory leží ve stejné třídě právě tehdy, když mají stejný syndrom.

Ukážeme teď, že počet vektorů v každé třídě je roven počtu kódových slov, tedy je roven 2^k. Z toho pak vyplývá, že počet všech tříd je roven $\frac{2^n}{2^k} = 2^{n-k}$.

Nechť T je třída podle kódu. Sestrojíme vzájemně jednoznačné zobrazení kódu na T. Zvolme libovolně (ale pevně) nějaké $\mathbf{v} \in T$. Uvažme zobrazení

$$\mathbf{x} \mapsto \mathbf{x} + \mathbf{v}$$

($\mathbf{x}$ je kódové slovo).

Přesvědčíme se nejprve, že $\mathbf{x} + \mathbf{v} \in T$. Je $\mathcal{H}(\mathbf{x} + \mathbf{v})^T = \mathcal{H}(\mathbf{x}^T + \mathbf{v}^T) = \mathcal{H}\mathbf{x}^T + \mathcal{H}\mathbf{v}^T = \mathbf{0} + \mathcal{H}\mathbf{v}^T = \mathcal{H}\mathbf{v}^T$. Vidíme, že $\mathbf{x} + \mathbf{v}$ má stejný syndrom jako $\mathbf{v}$, takže $\mathbf{x} + \mathbf{v} \in T$.

- Zobrazení je injekce: Nechť $\mathbf{x}$, $\mathbf{y}$ jsou kódová slova, $\mathbf{x} + \mathbf{v} = \mathbf{y} + \mathbf{v}$. Chceme: $\mathbf{x} = \mathbf{y}$. To jistě platí.

- Zobrazení je surjekce: Nechť $\mathbf{w} \in T$. Hledáme kódové slovo $\mathbf{x}$ splňující $\mathbf{x} + \mathbf{v} = \mathbf{w}$. Vezmeme $\mathbf{x} = \mathbf{w} - \mathbf{v}$. Stačí se přesvědčit, že $\mathbf{w} - \mathbf{v}$ je kódové slovo, tj. že $\mathcal{H}(\mathbf{w} - \mathbf{v})^T = \mathbf{0}$. Počítejme: $\mathcal{H}(\mathbf{w} - \mathbf{v})^T = \mathcal{H}(\mathbf{w}^T - \mathbf{v}^T) = \mathcal{H}\mathbf{w}^T - \mathcal{H}\mathbf{v}^T = \mathbf{0}$. Využili jsme fakt, že $\mathbf{v}, \mathbf{w} \in T$ a tudíž $\mathbf{v}$, $\mathbf{w}$ mají týž syndrom.

Možná chybová slova jsou právě slova z třídy, v níž leží přijaté slovo.

Dekodér tedy může okamžitě vyloučit všechna chybová slova, která neleží ve stejné třídě jako přijaté slovo, tj. chybová slova, která mají jiné syndromy. V úvahu však připadají všechna chybová slova, která jsou ve stejné třídě jako přijaté slovo. Žádné z těchto chybových slov nelze s jistotou vyloučit. Ale protože chyby kanálu jsou poměrně málo časté, některá chybová slova z dané třídy jsou mnohem méně pravděpodobná než jiná. Obecně platí, že chybová slova s malým počtem jedniček jsou pravděpodobnější než chybová slova s mnoha jedničkami. Přesněji, definujeme *váhu* libovolného n – rozměrného binárního vektoru jako počet jedniček v tomto vektoru. V libovolné třídě můžeme vybrat vektor s nejmenší váhou jako *reprezentanta* třídy. *Jedním z*

nejpravděpodobnějších chybových slov je reprezentant třídy obsahující přijaté slovo.

V našem příkladu je

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

Třídy podle kódu jsou v řádcích následující tabulky. V prvním řádku je třída mající nulový syndrom, tedy množina kódových slov. V soupisu slov třídy je na prvním místě vždy uveden reprezentant třídy. Každý prvek tabulky je součtem kódového slova, které je nahoře v jeho sloupci, a reprezentanta třídy, který je nalevo v jeho řádku. Připomeňme si, že pro každou třídu T a $\mathbf{v} \in T$ (za $\mathbf{v}$ můžeme vzít reprezentanta třídy T) je zobrazení $\mathbf{x} \mapsto \mathbf{x} + \mathbf{v}$ bijekce množiny všech kódových slov na T.

Slepianova standardní tabulka

syndromy	slova							
000	000000	001011	010101	011110	100110	101101	110011	111000
001	000001	001010	010100	011111	100111	101100	110010	111001
010	000010	001001	010111	011100	100100	101111	110001	111010
011	001000	000011	011101	010110	101110	100101	111011	110000
100	000100	001111	010001	011010	100010	101001	110111	111100
101	010000	011011	000101	001110	110110	111101	100011	101000
110	100000	101011	110101	111110	000110	001101	010011	011000
111	001100	000111	011001	010010	101010	100001	111111	110100

Budeme používat (a již jsme vlastně používali) terminologii z následující definice.

5.2.1. Definice. Kód se nazývá **lineární** právě tehdy, když jeho množina kódových slov je rovna množině všech vektorů $\mathbf{c}$, které splňují rovnici $\mathcal{H}\mathbf{c}^T = \mathbf{0}$, kde $\mathcal{H}$ je binární matice. Matice $\mathcal{H}$ se nazývá **kontrolní matice**. Délka bloků kódu, označená n, je rovna počtu sloupců matice $\mathcal{H}$. Počet kontrolních číslic kódu, označený r, je roven hodnosti matice $\mathcal{H}$. Počet informačních číslic kódu, označený k, je dán vztahem $k = n - r$. **Syndrom** slova $\mathbf{r}$ je definován vztahem $\mathbf{s}^T = \mathcal{H}\mathbf{r}^T$. **Třída** sestává ze všech slov majících daný syndrom. **Váha** slova je počet všech jedniček mezi jeho n číslicemi. V každé třídě je vybráno slovo mající nejmenší váhu jako **reprezentant** třídy.

Všechny lineární kódy mají důležité vlastnosti shrnuté v následující větě.

5.2.2. Věta. *Jestliže* $\mathbf{r}$ *je přijaté slovo, pak množina možných chybových slov je rovna třídě osahující* $\mathbf{r}$. *Nejpravděpodobnějším chybovým slovem je reprezentant třídy obsahující* $\mathbf{r}$. *Dekódovat můžeme následovně: Vypočtěte syndrom, najděte reprezentanta třídy mající tento syndrom a odečtěte reprezentanta této třídy od přijatého slova, abyste získali nejpravděpodobnější vyslané kódové slovo.*

Najednou vyvstávají dva hlavní problémy: (1) jak vybrat matici $\mathcal{H}$ a (2) jak určit reprezentanta třídy, je – li dán syndrom?

Pro malá n lze na tyto otázky odpovědět vyčerpávajícím (exhaustivním) hledáním – vyzkoušením všech možností. Pro velká n, k a r jsou však oba tyto problémy obecně nevyřešené. Je však známo velké množství "dobrých"metod výběru kontrolní matice a nalezení reprezentanta třídy, známe – li syndrom.

Triviální příklady dobrých lineárních kódů jsou opakovací kódy a kódy celkové kontroly parity, které jsme představili v předchozí části. Kontrolní matice pro kódy celkové kontroly parity má pouze jeden řádek a n sloupců

$$\mathcal{H} = (1111\ldots11)$$

Kontrolní matice pro opakovací kód má $n-1$ řádků a n sloupců. Pro $n = 5$:

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

Cvičení.

1. Kolik kódových slov mají kódy definované každou z následujících kontrolních matic?

$$\mathcal{H}_1 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\mathcal{H}_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

2. Pro kód definovaný kontrolní maticí $\mathcal{H}_2$ ze cvičení 1 najděte reprezentanta třídy obsahující následující slovo:

   - $\mathbf{r} = 111101000$
   - $\mathbf{r} = 110101011$
   - $\mathbf{r} = 100010001$
   - $\mathbf{r} = 010010010$

5.3 Hammingovy kódy

Při extrémně nízkých nebo extrémně vysokých rychlostech přenosu je poměrně snadné najít dobré lineární kódy. Abychom interpolovali mezi těmito dvěma extrémy, mohli bychom použít jeden ze dvou přístupů: (1) začít s nízkorychlostními kódy a postupně zvyšovat k přidáváním dalších a dalších kódových slov, ve snaze udržet velkou schopnost opravy chyb, nebo (2) začít s vysokorychlostními kódy a postupně zvyšovat schopnost opravy chyb a snažit se přitom přidat málo kontrolních číslic (tj. málo dalších kontrol parity).

Historicky se druhý přístup ukázal jako úspěšnější. Toto je přístup, který budeme následovat. Začínáme sestrojením jistých kódů pro opravu jednoduchých chyb, Hammingových kódů.

Syndrom chybového slova je dán rovnicí $\mathbf{s}^T = \mathcal{H}\mathbf{e}^T$. Pravá strana této rovnice může být napsána jako e_1 krát první sloupec matice $\mathcal{H}$ plus e_2 krát druhý sloupec matice $\mathcal{H}$ plus e_3 krát třetí sloupec matice $\mathcal{H}$ plus .... Například, jestliže

$$\mathbf{s}^T = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} (e_1, e_2, e_3, e_4, e_5, e_6)^T$$

pak

$$\begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = e_1 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + e_2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + e_3 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + e_4 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + e_5 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + e_6 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Syndrom lze tedy považovat za součet těch sloupců matice $\mathcal{H}$, kde došlo k chybám kanálu. Pokud je některý sloupec matice $\mathcal{H}$ nulový, pak chyba na této pozici nebude mít žádný vliv na syndrom. Kód na takové pozici nedokáže ani detekovat chybu. Pokud jsou dva sloupce matice $\mathcal{H}$ identické, pak jednoduchá chyba na jedné z těchto dvou pozic má stejný syndrom jako jednoduchá chyba na druhé z těchto pozic. Obě chybová slova leží ve stejné třídě. Vzhledem k tomu, že pouze jedno z nich může být vybráno jako reprezentant třídy, může být opravena pouze jedna z chyb. Přijaté slovo s druhou jednoduchou chybou bude dekódováno nesprávně. Lineární kód tedy není schopen opravit všechny jednoduché chyby, pokud sloupce jeho kontrolní matice $\mathcal{H}$ nejsou vzájemně různé a nenulové.

A naopak, předpokládejme, že všechny sloupce matice $\mathcal{H}$ jsou nenulové a vzájemně různé. Pak jednoduchá chyba na jakékoli pozici má za následek jiný (a nenulový) syndrom než jednoduchá chyba na jakékoli jiné pozici. Každý chybový vektor s jednou chybou je reprezentantem třídy a kód je schopen opravit všechny jednoduché chyby.

Dokázali jsme následující větu:

5.3.1. Věta. *Lineární kód je schopen opravit všechna přijatá slova s nejvýše jednou chybou právě tehdy, když všechny sloupce matice $\mathcal{H}$ jsou nenulové a vzájemně různé.*

Při dekódování dekodér vypočítá syndrom přijatého slova. Pokud je syndrom nulový, dekodér předpokládá, že nedošlo k žádné chybě. Pokud je syndrom nenulový a roven některému sloupci matice $\mathcal{H}$, pak dekodér předpokládá chybu na příslušné pozici. Pokud je syndrom nenulový a nerovná se žádnému sloupci matice $\mathcal{H}$, pak tento neúplný dekódovací algoritmus selže. Selhání a chyby dekódování mohou nastat pouze v případě, že dojde ke dvěma nebo více chybám. Předpokládejme například, že

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

Jestliže přijaté slovo je $\mathbf{r} = 101000101$, pak dekodér vypočítá $\mathbf{s} = 1100$. Protože $\mathbf{s}^T$ je rovno pátému sloupci matice $\mathcal{H}$, rozhodneme se, že

$$\mathbf{e} = 000010000,\ \mathbf{c} = 101010101$$

Jestliže však přijaté slovo je $\mathbf{r} = 111001111$, pak $\mathbf{s} = 1101$. Protože $\mathbf{s}^T$ není rovno žádnému sloupci matice $\mathcal{H}$, dekódovací algoritmus selže, zjistí, že v přijatém slovu jsou aspoň dvě chyby.

Zeptejme se nyní: "Jaká je největší možná délka bloku v lineárním kódu opravujícím jednu chybu, který má r kontrolních číslic?" Protože počet kontrolních číslic je roven maximálnímu počtu lineárně nezávislých řádků matice $\mathcal{H}$, je tato otázka prakticky totožná s otázkou: „Jaký je maximální počet vzájemně různých nenulových sloupců, které se mohou vyskytovat v binární matici s r řádky?" Odpověď je jasná: $2^r - 1$. Předpokládejme, že sloupce matice $\mathcal{H}$ jsou sestaveny ze všech $2^r - 1$ nenulových binárních r–tic, uspořádaných v libovolném pořadí. Lineární kód definovaný takovou maticí se nazývá *Hammingův kód*.

Pro každé kladné celé číslo r existuje binární Hammingův kód mající r kontrolních číslic. Délka bloků tohoto kódu je $n = 2^r - 1$ a počet informačních číslic (číslic zprávy) je roven $k = n - r = 2^r - 1 - r$. Kód je schopen opravit jednoduchou chybu na jakékoli pozici. Navíc, jelikož každý možný nenulový syndrom je roven nějakému sloupci matice $\mathcal{H}$, dekódování nikdy neselže. Dekódovací algoritmus je úplný. Každá třída má reprezentanta, jehož váha je nula nebo jedna. Žádné chybové slovo váhy aspoň dva nemůže být ani objeveno ani opraveno.

Aby byla zajištěna ochrana proti dvojnásobným chybám, je někdy k Hammingovu kódu připojena dodatečná kontrola celkové parity. Výsledný kód má bloky délky $n = 2^m$, $r = m + 1$ kontrolních číslic a $k = 2^m - 1 - m$ informačních číslic. Například, Hammingův kód s bloky délky 15 má kontrolní matici

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

Připojením celkové kontroly parity se získá *rozšířený* Hammingův kód s délkou bloků 16, jehož kontrolní matice je

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

Každý sloupec kontrolní matice rozšířeného Hammingova kódu má v horním řádku 1 a součet libovolných dvou sloupců má nahoře 0. Z tohoto důvodu se syndrom jakéhokoli chybového slova váhy 2 bude lišit od kteréhokoli sloupce matice $\mathcal{H}$. Je také nenulový, jelikož žádné dva sloupce matice $\mathcal{H}$ nejsou stejné. Jakékoli přijaté slovo s 2 chybami musí způsobit selhání dekódování. Rozšířené Hammingovy kódy jsou tedy schopny opravit všechny jednoduché chyby a detekovat všechna chybová slova o váze 2 (jsou – li v přijatém slově dvě chyby, dekódování selže).

Ačkoli sloupce kontrolní matice Hammingova kódu mohou být uspořádány v libovolném pořadí, některá uspořádání vedou k mnohem jednodušší práci než jiná. Vážný (ale opravdu vážný) zájemce si o tom může přečíst v části 5.1 v [3].

Hammingův kód mající bloky délky $n = 2^m - 1$ má rychlost přenosu informace

$$R = \frac{k}{n} = \frac{2^m - 1 - m}{2^m - 1} = 1 - \frac{m}{2^m - 1}$$

I když můžeme konstruovat Hammingovy kódy s většími a většími délkami bloků, delší kódy budou mít rychlost přenosu blížící se 1. Přestože jsou tedy velkým krokem vpřed od kódů celkové kontroly parity, Hammingovy kódy jsou stále třídou kódů s velmi vysokou rychlostí přenosu informace. Nejsou schopny opravit žádné dvojnásobné nebo vícenásobné chyby. Abychom opravili takové chyby, musíme hledat kódy s větším počtem kontrolních číslic.

Poměrně velká složitost nejjednodušších známých binárních kódů opravujících dvojnásobné chyby ostře kontrastuje s jednoduchostí Hammingových kódů. Velké množství práce v konstruktivní teorii kódování následovalo po publikování Hammingova průkopnického článku v roce 1950. Bylo zavedeno mnoho nových typů kódů, ale až na několik důležitých výjimek byla většina nových konstrukcí specializovanými kódy pro velmi specializované účely. Dokonce i další nejjednodušší problém, konstrukce binárních kódů s relativně

vysokou rychlostí přenosu informace, které by opravovaly jakékoli dvojnásobné chyby, zůstával nevyřešen. Když Bose a Chaudhuri v roce 1960 a Hocquenghem v roce 1959 konečně objevili kódy pro opravu dvojnásobných chyb, okamžitě následovalo zobecnění pro opravy chyb t – násobných, pro všechna t.

V mnoha ohledech představuje propast mezi Hammingovými kódy z roku 1950 a BCH kódy z roku 1960 dokonce více než deset let výzkumu. Ve skutečnosti většinu Hammingových výsledků předvídal v trochu jiném kontextu již Fisher v roce 1942 v článku, který Bose dobře znal.

Cvičení.

1. Uvažte tři Hammingovy kódy definované následujícími kontrolními maticemi:

$$\mathcal{H}_1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\mathcal{H}_2 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathcal{H}_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

   - Pro každý z daných tří kódů dekódujte přijatá slova $\mathbf{r}_1 = 1110000$ a $\mathbf{r}_2 = 1111000$.
   - Ukažte, že dvě z daných tří matic definují týž kód.

5.4 Úvod do BCH kódů pro opravu dvojnásobných chyb

Toto je rozšiřující látka a text bude napsán později.

6 Kořeny polynomů

Čím se budeme zabývat v poslední části tohoto textu? To je přece jasné, čekají nás kořeny polynomů (a problematika s nimi související). Nechť f je polynom nad R, kde R je komutativní asociativní okruh s jednotkovým prvkem. Co je to kořen polynomu f? Je to takový prvek $c \in R$, pro který je $f(c) = 0$. Co je to $f(c)$? Pro

$$f(x) = a_n x^n + \cdots + a_1 x + a_0$$

je

$$f(c) = a_n c^n + \cdots + a_1 c + a_0$$

Hledat kořeny polynomu f tedy znamená řešit rovnici

$$a_n x^n + \cdots + a_1 x + a_0 = 0$$

s neznámou x. (Rovnice tohoto typu se nazývají algebraické.)

6.1 Násobnost a počet kořenů polynomu

Již na střední škole jste hledali kořeny kvadratických polynomů (tj. řešili jste kvadratické rovnice). Zjistili jste tehdy, že kvadratický polynom s reálnými koeficienty nemá žádný reálný kořen, nebo má jeden reálný kořen, nebo má dva reálné kořeny, avšak nikdy nemá více než dva kořeny. V této části (mimo jiné) dokážeme obecné tvrzení: Jestliže f je polynom nad oborem integrity I s jednotkovým prvkem a f má stupeň n, pak f má nejvýše n kořenů v I.

6.1.1. Věta. *Nechť R je komutativní asociativní okruh s jednotkovým prvkem, f je polynom nad R, $c \in R$. Platí:*

$$f(c) = 0 \Longleftrightarrow x - c / f(x)$$

Důkaz.
$\Longleftarrow$: Nechť $x - c / f(x)$. Existuje $g \in R[x]$, $f(x) = (x - c)g(x)$. Pak $f(c) = (c - c)g(c) = 0 \cdot g(c) = 0$.
$\Longrightarrow$: Nechť $f(c) = 0$. Předpokládejme nejprve, že polynom f je konstantní. Pak $f(x) = a$, kde $a \in R$. Je $f(c) = a$, takže $a = 0$ a f je nulový polynom. Pak

je jasné, že $x - c/f(x)$. Předpokládejme nyní, že polynom f není konstantní. Pak $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, kde n je celé číslo, $n \geq 1$, $a_0, \ldots, a_n \in R$, $a_n \neq 0$. Počítejme:

$$\begin{aligned} f(x) &= f(x) - 0 \\ &= f(x) - f(c) \\ &= (a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0) \\ &\quad -(a_n c^n + a_{n-1}c^{n-1} + \cdots + a_1 c + a_0) \\ &= a_n x^n - a_n c^n + a_{n-1}x^{n-1} - a_{n-1}c^{n-1} + \cdots + a_1 x - a_1 c + a_0 - a_0 \\ &= a_n(x^n - c^n) + a_{n-1}(x^{n-1} - c^{n-1}) + \cdots + a_1(x - c) \end{aligned}$$

Nyní si stačí uvědomit, že pro každé kladné celé číslo k platí $x-c/x^k-c^k$. Pak totiž $x-c$ dělí každý sčítanec v $a_n(x^n-c^n)+a_{n-1}(x^{n-1}-c^{n-1})+\cdots+a_1(x-c)$ a tedy $x - c$ dělí součet, čili $x - c$ dělí $f(x)$.

Proč $x - c/x^k - c^k$? Pro $k = 1$ je to jasné, pro $k > 1$ uvažte rovnost

$$x^k - c^k = (x - c)(x^{k-1} + x^{k-2}c + \cdots + xc^{k-2} + c^{k-1})$$

Nyní již dokážeme větu o počtu kořenů polynomu.

6.1.2. Věta. *Nechť I je obor integrity s jednotkovým prvkem. Nechť f je nenulový polynom nad oborem integrity I. Jestliže f má stupeň n, pak f má nejvýše n kořenů v I.*

Důkaz. Postupujme indukcí vzhledem k n.
$n = 0$: Nechť f má stupeň 0. Je $f(x) = a$, kde $a \in I$, $a \neq 0$. Pro každé $c \in I$ je $f(c) = a \neq 0$, takže f nemá v I žádný kořen a tedy má v I nejvýše n kořenů.
$n > 0$: Nechť f má stupeň n. Množinu všech kořenů polynomu f v oboru integrity I označme A. Chceme: $|A| \leq n$. Jsou dvě možnosti:

- $A = \emptyset$: Je $|A| = 0$, takže $|A| < n$.
- $A \neq \emptyset$: Zvolme $c \in A$. Je $f(c) = 0$ a dle věty 6.1.1. pak $x - c/f(x)$. Potom $f(x) = (x-c)g(x)$ pro nějaký polynom g nad I. Množinu všech kořenů polynomu g v oboru integrity I označme B. Protože polynom

f je nenulový, je také polynom g nenulový. Platí: $n = deg(f(x)) = deg((x-c)g(x)) = deg(x-c) + deg(g(x)) = 1 + deg(g(x))$. Vidíme, že polynom g má stupeň $n-1$. Dle indukčního předpokladu $|B| \leq n-1$. Ukážeme teď, že $A \subseteq \{c\} \cup B$. Zvolme libovolně $d \in A$. Musíme ukázat, že $d \in \{c\} \cup B$. Protože $d \in A$, je $f(d) = 0$. Ovšem $f(x) = (x-c)g(x)$, takže $f(d) = (d-c)g(d)$, $(d-c)g(d) = 0$. Jelikož g je polynom nad I a $d \in I$, je $g(d) \in I$. Tudíž $d-c \in I$, $g(d) \in I$, $(d-c)g(d) = 0$. Protože I je obor integrity, dostáváme $d-c = 0$ nebo $g(d) = 0$. Pokud $d-c = 0$, je $d = c$ a tedy $d \in \{c\} \cup B$. Pokud $g(d) = 0$, je $d \in B$ a tedy také $d \in \{c\} \cup B$. Nyní víme: $|B| \leq n-1$ a $A \subseteq \{c\} \cup B$. Proto

$$|A| \leq |\{c\} \cup B| \leq |\{c\}| + |B| = 1 + |B| \leq 1 + (n-1) = n,\ |A| \leq n$$

6.1.3. Poznámka.

1. Samozřejmě se může stát, že polynom má méně kořenů, než je jeho stupeň. Například kvadratický polynom x^2+1 je polynom nad tělesem $\mathbb{R}$, který nemá žádný kořen v $\mathbb{R}$.

2. V důkazu věty 6.1.2. bylo podstatné, že I byl obor integrity. Z $(d-c)g(d) = 0$ jsme mohli odvodit, že $d-c = 0$ nebo $g(d) = 0$. Pokud bychom uvažovali polynomy nad komutativními asociativními okruhy s jednotkovým prvkem, nebude věta 6.1.2. platit – například kvadratický polynom $x^2 + x$ nad $\mathbb{Z}_6$ má 4 kořeny v $\mathbb{Z}_6$ (najděte je!).

Nakonec ještě budeme definovat násobnost kořene (neboť je slíbeno v názvu, že se v této části o násobnosti kořene aspoň něco řekne).

6.1.4. Definice. Nechť R je komutativní asociativní okruh s jednotkovým prvkem. Nechť f je polynom nad R, $c \in R$, k je nezáporné celé číslo. Říkáme, že prvek c je k-násobným kořenem polynomu f, pokud

$$(x-c)^k / f(x)\ \wedge\ \neg((x-c)^{k+1} / f(x))$$

6.1.5. Poznámka. O násobnosti kořene polynomu si můžete přečíst v paragrafu 1 Násobnost kořene polynomu. Hornerovo schéma v kapitole XV

Vlastnosti kořenů polynomů v učebnici [5]. Přečtete si tam například to, jak můžete násobnost kořene zjistit pomocí Hornerova schématu.

Cvičení.

1. Zjistěte násobnost kořenů $c_1 = \frac{1}{2}$, $c_2 = -2$ polynomu

$$f(x) = 8x^7 + 4x^6 - 2x^5 + 3x^4 - 6x^3 - 2x^2 + 4x - 1 \in \mathbb{Q}[x]$$

2. Nechť f je polynom nad oborem integrity I s jednotkovým prvkem, $c \in I$, k je nezáporné celé číslo. Dokažte: c je k-násobným kořenem polynomu f právě tehdy, když existuje $g \in I[x]$ takový, že

$$f(x) = (x - c)^k g(x) \ \wedge \ g(c) \neq 0$$

6.2 Základní věta algebry

Pod názvem Základní věta algebry se skrývá následující tvrzení:

6.2.1. Věta. (**Základní věta algebry**) *Každý nekonstantní polynom s komplexními koeficienty má aspoň jeden komplexní kořen.*

Základní větu algebry dokazovat nebudeme. Větu poprvé přesvědčivě dokázal C. F. Gauss v roce 1799. Přestože tvrzení Základní věty algebry je snadno pochopitelné i pro středoškolského studenta, pro její důkazy to v žádném případě neplatí.

Základní věta algebry má zajímavé a významné důsledky. Jeden z nich uvedeme a také dokážeme.

6.2.2. Důsledek. *Nechť f je nekonstantní monický polynom stupně n s komplexními koeficienty. Pak f lze v $\mathbb{C}[x]$ rozložit na součin n monických lineárních polynomů, přičemž tento rozklad je jednoznačný až na pořadí.*

Důkaz.

1. Existence: Postupujme indukcí vzhledem k $deg(f) = n$. Pro $deg(f) = 1$ není co řešit, hledaným rozkladem je "rozklad" $f = f$. Nechť tedy $deg(f) = n > 1$. Dle Základní věty algebry existuje komplexní číslo c, $f(c) = 0$. Pak $x - c/f$ (viz 6.1.1) a tedy $f = (x - c)g$ pro nějaký polynom g nad $\mathbb{C}$. Je $f \neq 0$, takže $g \neq 0$, $1 = lc(f) = lc((x - c)g) =$

$lc(x-c)lc(g) = 1 \cdot lc(g) = lc(g)$. Vidíme, že g je monický polynom. Dále, $n = deg(f) = deg((x-c)g) = deg(x-c)+deg(g) = 1+deg(g), deg(g) = n-1$. Dle indukčního předpokladu existují monické lineární polynomy $h_1, \ldots, h_{n-1} \in \mathbb{C}[x]$, $g = h_1 \cdots h_{n-1}$. Takže $f = (x-c)h_1 \cdots h_{n-1}$.

2. Jednoznačnost: Nechť $g_1 \cdots g_n = h_1 \cdots h_n$, kde $g_1, \ldots, g_n$ a $h_1, \ldots, h_n$ jsou monické lineární polynomy nad $\mathbb{C}$. Chceme: Existují $i_1, \ldots, i_n \in \mathbb{Z}$, $\{1, \ldots, n\} = \{i_1, \ldots, i_n\}$, $g_1 = h_{i_1}, \ldots, g_n = h_{i_n}$. Je $g_1 = x+c$ pro nějaké $c \in \mathbb{C}$. Pak $g_1(-c) = 0$ a tedy také $(g_1 \cdots g_n)(-c) = 0$, $(h_1 \cdots h_n)(-c) = 0$, $h_1(-c) \cdots h_n(-c) = 0$. Existuje $i_1 \in \{1, \ldots, n\}$, $h_{i_1}(-c) = 0$. Ovšem $h_{i_1} = x + d$ pro nějaké $d \in \mathbb{C}$. Pak $0 = h_{i_1}(-c) = -c + d$, $0 = -c + d$, $c = d$, $g_1 = h_{i_1}$. Je

$$\begin{aligned} g_1 g_2 \cdots g_n &= h_1 \cdots h_{i_1-1} h_{i_1} h_{i_1+1} \cdots h_n \\ g_1 g_2 \cdots g_n &= h_1 \cdots h_{i_1-1} g_1 h_{i_1+1} \cdots h_n \\ g_2 \cdots g_n &= h_1 \cdots h_{i_1-1} h_{i_1+1} \cdots h_n \end{aligned}$$

Nyní obdobně postupujeme dále. Najdeme $i_2 \in \{1, \ldots, n\} - \{i_1\}$, $g_2 = h_{i_2}$, atd.

6.3 Binomické rovnice

Rovnice

$$x^n - \alpha = 0$$

kde n je kladné celé číslo a α je nenulové komplexní číslo, se nazývá **binomická rovnice**. Chceme najít všechna řešení binomické rovnice v oboru komplexních čísel. Řešení rovnice $x^n - \alpha = 0$ nazýváme n-té odmocniny z α.

Úkol. Prostudujte paragraf 1 Binomické rovnice v kapitole XVII Algebraické řešení algebraických rovnic v knize [5]. Alternativně můžete prostudovat části 2 Druhá odmocnina komplexního čísla, 8 Odmocniny komplexních čísel s celými kladnými odmocniteli a 10 Binomické rovnice v kapitole 2 Řešení některých speciálních typů algebraických rovnic v knížce [11]; najdete tam též hodně příkladů i cvičení. Výhodou knížky [11] je volná dostupnost na internetu.

Cvičení.

1. Najděte všechna řešení binomické rovnice $x^4 - 1 = 0$ v oboru komplexních čísel.

2. Najděte všechna řešení binomické rovnice $x^3 - 1 = 0$ v oboru komplexních čísel.

6.4 Kvadratické a kubické rovnice

Jak víte, **kvadratická rovnice** je rovnice

$$ax^2 + bx + c = 0$$

kde a, b, c jsou komplexní čísla, $a \neq 0$.

Kubická rovnice pak je rovnice

$$ax^3 + bx^2 + cx + d = 0$$

kde a, b, c, d jsou komplexní čísla, $a \neq 0$.

Chceme najít všechny kořeny kvadratické rovnice i kubické rovnice v oboru komplexních čísel.

Již na střední škole jste se naučili, jak postupovat při řešení kvadratické rovnice. Teď si tento postup zopakujete a také se naučíte tzv. Cardanovy vzorce pro výpočet kořenů polynomů stupně třetího.

Úkol. Prostudujte části 1 Kvadratické rovnice s reálnými koeficienty, 3 Kvadratické rovnice s komplexními koeficienty a 9 Kubické rovnice a rovnice čtvrtého stupně v kapitole 2 Řešení některých speciálních typů algebraických rovnic v knížce [11]. Opět konstatuji, že knížka [11] je snadno a legálně dostupná na internetu a obsahuje hodně příkladů i cvičení. Také v ní v kapitole 2 najdete další zajímavé informace související s právě probíranou látkou – například části 4 Řešení některých rovnic převedením na kvadratickou rovnici, 5 Slovní úlohy vedoucí na kvadratickou rovnici, 13 Reciproké rovnice třetího stupně, 14 Reciproké rovnice čtvrtého stupně, 15 Reciproké rovnice pátého stupně. Můžete si také přečíst paragraf 2 Algebraická řešitelnost rovnic 2., 3. a 4. stupně v kapitole XVII Algebraické řešení algebraických rovnic v učebnici [5]; látka tohoto paragrafu je však pokročilejší (obtížnější).

Cvičení.

1. Vyřešte rovnici $x^2 + 2x + 3 = 0$ v oboru komplexních čísel.

2. Vyřešte rovnici $x^3 - 6x - 9 = 0$ v oboru komplexních čísel.

6.5 Kořeny polynomů nad celými čísly

Zde vyslovíme a dokážeme jednoduché tvrzení, které lze použít k nalezení všech racionálních kořenů polynomu s celočíselnými koeficienty.

6.5.1. Tvrzení. *Nechť n je kladné celé číslo. Nechť*

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

je polynom nad celými čísly, tj. $a_0, \ldots, a_n$ jsou celá čísla. Nechť r, s jsou celá čísla, $s > 0$, r a s jsou nesoudělná. Jestliže $f(\frac{r}{s}) = 0$, pak $r|a_0$ a $s|a_n$.

Důkaz. Předpokládejme, že $f(\frac{r}{s}) = 0$. Pak

$$a_n \left(\frac{r}{s}\right)^n + a_{n-1}\left(\frac{r}{s}\right)^{n-1} + \cdots + a_1\left(\frac{r}{s}\right) + a_0 = 0$$

Rovnost vynásobíme číslem s^n. Dostaneme

$$a_n r^n + a_{n-1}r^{n-1}s + \cdots + a_1 r s^{n-1} + a_0 s^n = 0$$

Číslo r dělí všechny sčítance $a_n r^n, a_{n-1}r^{n-1}s, \ldots, a_1 r s^{n-1}$, takže $r|a_0 s^n$. Jelikož $r \perp s$, máme $r|a_0$.

Číslo s dělí všechny sčítance $a_{n-1}r^{n-1}s, \ldots, a_1 r s^{n-1}, a_0 s^n$, takže $s|a_n r^n$. Jelikož $r \perp s$, máme $s|a_n$.

Jak nám pomůže tvrzení 6.5.1 při hledání racionálních kořenů polynomu $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$? Jistě lze předpokládat, že $a_n \neq 0$. Předpokládejme také, že $a_0 \neq 0$. Jestliže $f(\frac{r}{s}) = 0$ ($r, s \in \mathbb{Z}$, $s > 0$, $r \perp s$), pak dle 6.5.1 máme $r|a_0$ a $s|a_n$. Ovšem $r|a_0$ dává $|r| \leq |a_0|$ a $s|a_n$ dává $|s| \leq |a_n|$. Dostaneme tak konečný počet kandidátů na racionální kořen polynomu $f(x)$ a dosazením zjistíme, který z kandidátů opravdu je kořenem.

Možná namítnete, že může být $a_0 = 0$. Pak pro každé celé číslo r máme $r|a_0$ a máme nekonečně mnoho kandidátů na kořen polynomu $f(x)$. Ano, avšak i v případě $a_0 = 0$ nám tvrzení 6.5.1 pomůže získat pouze konečný počet kandidátů na kořen. Jak? Na to jistě přijdete sami.

Cvičení.

1. Najděte všechny racionální kořeny polynomu $2x^6 - 3x^4 + 2x^3 - x + 1$.

2. Nechť n je kladné celé číslo. Nechť

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

je polynom nad celými čísly, tj. $a_0, \ldots, a_n$ jsou celá čísla. Nechť r, s jsou celá čísla, $s > 0$, r a s jsou nesoudělná. Jestliže $f(\frac{r}{s}) = 0$, pak $r - ms/f(m)$ pro libovolné celé číslo m. Specielně $r - s/f(1)$ a $r + s/f(-1)$. Dokažte.

3. Najděte všechny racionální kořeny polynomu $x^3 - 6x^2 + 15x - 14$.

4. Najděte všechny komplexní kořeny polynomu $x^3 - x^2 + 3x + 5$. Poznámka: Možná se vám bude hodit algoritmus dělení pro polynomy (věta 4.5.4).

5. Najděte všechny komplexní kořeny polynomu $x^4 - 2x^3 - 4x^2 - 16x$.

6.6 Hornerovo schéma

Nechť n je kladné celé číslo, I je obor integrity s jednotkovým prvkem. Nechť

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

je polynom nad oborem integrity I, tj. $a_0, \ldots, a_n \in I$.

Hornerovo schéma je efektivní algoritmus pro určení hodnoty $f(c)$, kde $c \in I$. Kdy se nám to bude hodit? Například tehdy, když budeme chtít zjistit, zda c je kořenem polynomu $f(x)$.

Myšlenku, na níž je Hornerovo schéma založeno, ukážeme nejprve pro $n = 4$. Máme

$$f(x) = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

Při naivním výpočtu hodnoty $f(c)$ postupujeme takto:

$$f(c) = a_4 cccc + a_3 ccc + a_2 cc + a_1 c + a_0$$

Vidíme, že potřebujeme provést 10 násobení a 4 sčítání. Při chytřejším počítání postupně vypočítáme $cc = c^2$, $c^2 c = c^3$ a $c^3 c = c^4$ – k tomu potřebujeme 3 násobení. No a pak spočteme $a_4 c^4 + a_3 c^3 + a_2 c^2 + a_1 c + a_0 = f(c)$ - teď jsme potřebovali 4 násobení a 4 sčítání. Celkem jsme tedy pro výpočet $f(c)$ potřebovali 7 násobení a 4 sčítání.

Můžeme postupovat ještě chytřeji? Ano, a to s využitím vztahu

$$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = (((a_4x + a_3)x + a_2)x + a_1)x + a_0$$

Tento vztah je základem Hornerova schématu pro $n = 4$. Vidíme, že k výpočtu $f(c)$ teď potřebujeme 4 násobení a 4 sčítání.

Obecně, základem Hornerova schématu je vztah

$$f(x) = (\dots(((a_nx + a_{n-1})x + a_{n-2})x + a_{n-3})x + \cdots + a_1)x + a_0$$

Dle Hornerova schématu pro výpočet hodnoty $f(c)$ je potřeba n násobení a n sčítání.

Lze dokonce dokázat, že Hornerovo schéma je nejlepší algoritmus pro vyhodnocení polynomu – neexistuje žádný algoritmus, který by používal méně než n sčítání a také neexistuje žádný algoritmus, který by používal méně než n násobení.

Na závěr si ještě povšimněte, že při naivním výpočtu $f(c)$ (pouhým dosazením za x) je potřeba $\frac{n(n+1)}{2}$ násobení a n sčítání, a při výpočtu, v němž si nejdříve postupně určíme hodnoty $c^2, c^3, \dots, c^n$, pak použijeme $2n - 1$ násobení a n sčítání.

6.6.1. Poznámka. Můžete si přečíst poměrně podrobné pojednání o Hornerově schématu v paragrafu 1 Násobnost kořene polynomu. Hornerovo schéma v kapitole XV Vlastosti kořenů polynomů v knize [5]. Tam se také dočtete, jak lze Hornerovo schéma využít při určování násobnosti kořene.

Cvičení.

1. Užitím Hornerova schématu určete kořeny polynomu $x^4-x^3-7x^2+x+6$.
2. Která z čísel $-3, -2, 0, 1, 2, 3, 4$ jsou kořeny polynomu $p(x)$?

$$p(x) = x^5 - x^4 - 10x^3 - 5x^2 - 21x + 36$$

7 Pokyny (doporučení) k tempu studia

Mnozí si jistě sami dobře rozdělíte čas věnovaný (samo)studiu předmětu Algebra s aplikacemi. Některou látku pochopíte rychleji (poznáte to například

z toho, že snadno vyřešíte cvičení či budete schopni samostatně dokázat tvrzení nebo aspoň důkazům dobře porozumíte), některé látce budete muset věnovat více času.

Při studiu předmětu na univerzitě budete samozřejmě mít svého konkrétního vyučujícího, který vám (možná) dá pokyny či doporučení k organizaci studia předmětu nebo se s ním aspoň budete moci poradit.

Níže najdete doporučení k tempu studia. Látka, obsažená v této studijní opoře, je rozdělena do 13 týdnů, neboť právě tolik týdnů má výuková část semestru. Nedivte se, že některý týden byste měli prostudovat 16 stran a v jiném týdnu 1 stranu – souvisí to samozřejmě s tím, že některá látka je v opoře přímo vyložena, kdežto jinou látku máte celou studovat z jiného textu.

Návrh rozdělení studia předmětu Algebra s aplikacemi do 13 týdnů:

Jestliže jste ještě nečetli kapitolu 1 Úvod (strany 2 – 6), pak si ji určitě přečtěte.

1. 2.1 Úvod, 2.2 Indukce, 2.3 Algoritmus dělení: GCD a LCM
   strany 6 – 22

2. 2.4 Prvočísla, 2.5 Modulární aritmetika
   strany 22 – 35

3. 2.6 Řešení lineárních kongruencí, 2.7 Eulerova věta
   strany 35 - 51

4. 2.8 Kryptografie s veřejným klíčem
   strany 52 – 65

5. 3.1 Základní pojmy teorie grup
   strany 66 – 67

6. 3.2 Příklady grup
   strany 67 – 68

7. 3.3 Lagrangeova věta a její důsledky, 3.4 Cyklické grupy
   strana 68

8. 4.1 Definice algebraické struktury, 4.2 Pologrupy a monoidy, 4.3 Okruhy a tělesa
   strany 68 – 78

9. 4.4 Algebra polynomů, 4.5 Euklidův algoritmus pro polynomy
   strany 79 – 96

10. 5.1 Opakovací kódy a kódy kontroly parity, 5.2 Lineární kódy, 5.3 Hammingovy kódy
    strany 96 – 109

11. 6.1 Násobnost a počet kořenů polynomu, 6.2 Základní věta algebry, 6.3 Binomické rovnice
    strany 110 – 115

12. 6.4 Kvadratické a kubické rovnice, 6.5 Kořeny polynomů nad celými čísly, 6.6 Hornerovo schéma
    strany 115 – 118

13. Časová rezerva.

Reference

[1] Adámek, J.: *Kódování.* SNTL, Praha, 1989.

[2] Aigner, M., Ziegler, G. M.: *Proofs from THE BOOK.* Springer, 2010.

[3] Berlekamp, E. R.: *Algebraic Coding Theory.* Aegean Park Press, 1984.

[4] Blažek, J., Calda, E., Koman, M., Kussová, B.: *Algebra a teoretická aritmetika, I. díl.* Státní pedagogické nakladatelství, n.p., Praha, 1983.

[5] Blažek, J., Koman, M., Vojtášková, B.: *Algebra a teoretická aritmetika, II. díl.* Státní pedagogické nakladatelství, n.p., Praha, 1985.

[6] Gruska, J.: *Foundations of Computing.* International Thomson Computer Press, Boston, 1997.

[7] Hruša, K.: *Polynomy v moderní algebře.* Mladá fronta, Praha, 1970. `https://dml.cz/handle/10338.dmlcz/403708`

[8] Knuth, D. E.: *Umění programování, 2. díl - Seminumerické algoritmy.* Computer Press, a.s., Brno, 2010.

[9] Kuřil, M.: *Lineární algebra.*
https://kma.ujep.cz/administrace/uploads/144f052.pdf

[10] Kuřil, M.: *Základy algebry.*
https://kma.ujep.cz/administrace/uploads/8f878f1.pdf

[11] Šisler, M.: *O řešení algebraických rovnic.* Mladá fronta, Praha, 1966.
https://dml.cz/handle/10338.dmlcz/403551

[12] Švejdar, V.: *Logika – neúplnost, složitost a nutnost.* Academia, Praha, 2002.

Martin Kuřil
Katedra matematiky
Přírodovědecká fakulta
Univerzita Jana Evangelisty Purkyně
Ústí nad Labem
martin.kuril@ujep.cz