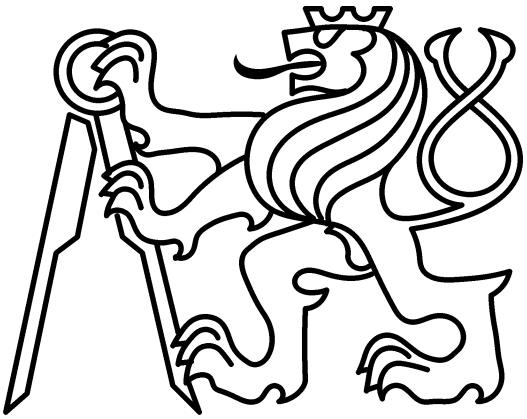
České Vysoké Učení Technické v Praze
Fakulta elektrotechnická

Diskrétní matematika

Text k přednášce

Jiří Velebil
katedra matematiky
Praha, 2007

velebil@math.feld.cvut.cz
http://math.feld.cvut.cz/velebil



Obsah

Úvod

Počítače nám dovolují dělat více chyb a rychleji, než kterýkoli jiný vynález v historii lidstva — s možnou výjimkou střelných zbraní a tequily.
Mitch Ratcliffe

Text, který začínáte číst, by měl posloužit jako skriptum k předmětu *Diskrétní matematika* pro druhý ročník. Cílem předmětu je podat solidní základy matematiky, potřebné ke studiu *computer science*.[1] Samozřejmě, že nebylo možné postihnout všechny aspekty moderní computer science, bylo nutné se řídit sylabem přednášky. V této úvodní kapitole je tudíž stručně popsáno, co lze v textu nalézt a co všechno v něm chybí. Toto je varianta textu, vzniklého přepracováním textu pro předmět *Diskrétní matematika a logika*. Proto přivítám jakékoli reakce.

Text je zveřejněn jako „klikací PDF“, tj. jako soubor s hypertextovými odkazy. Přesto nejde o „klikací knihu“, takový text bych psal a řadil úplně jinak. Hypertextové odkazy jsou vytvořeny jen pro zpříjemnění prohlížení elektronické podoby dokumentu.

Poděkování. Rád bych poděkoval studentkám a studentům *Diskrétní matematiky a logiky* zimního semestru 2006 za připomínky k některým částem textu. Za případné chyby, které v textu přesto zůstaly, nesu ovšem plnou odpovědnost já sám. K napsání byl použit systém LaTeX Leslieho Lamporta, hypertextové odkazy byly vytvořeny balíkem hyperref Sebastiana Rahtze a Heiko Oberdiek, obrázky byly nakresleny pomocí maker XY-pic Kristoffera H. Rose a Rosse Moorea.

Přeji Vám příjemnou četbu.

Jiří Velebil
V Praze, v červnu 2007

Co v textu je

Byla to kniha na zabití času pro ty, kteří ho mají raději mrtvý.
Rose MacCaulay

I když je text psaný systémem definice, věta, důkaz, je v textu řada příkladů, které teoretické pojmy ilustrují na praktických problémech. Na konci každé kapitoly potom je řada cvičení. Pokud při čtení narazíte na značku

zvolněte: tak je označeno něco, co si vyžaduje Vaši pozornost. V každé kapitole na konci je také seznam možné doplňující četby. *V žádném případě nejde o literaturu povinnou.* Všechny definice, tvrzení apod. jsou průběžně číslovány a značkou ■ je označen konec důkazu. Na konci textu je rejstřík.

[1] Jak poznamenal známý odpůrce příkazu `goto` (a vynikající matematik) Edsger Dijkstra (1930–2002), je computer science natolik vědou o počítačích, nakolik je astronomie vědou o dalekohledech. V textu toho o dalekohledech mnoho nenajdete.

V textu jsou často i historické odbočky a poznámky, které bezprostředně s probíranou látkou nesouvisí. Účelem bylo ukázat, že matematika je jedna velká stavba, kterou staví lidé z masa a kostí, není to jen sbírka pouček z přednášek.

Kapitola 1 je věnována nejrůznějším principům indukce. Možná nezvyklý pro Vás může být *dynamický* pohled na indukci jako na běh rekursivního algoritmu. Rekursivní algoritmy se vinou celým textem jako červená nit a většinou budete vyzváni dokázat *korektnost* Vámi navrženého algoritmu. K tomu je často vhodná právě indukce. Dalším tématem kapitoly 1 jsou *množiny zadané induktivně*. Zvládnutí této partie je užitečné při studiu formálních jazyků. Příklady induktivně zadaných množin potkáme znovu v kapitolách 5 a 6 při *definici termů*.

Kapitoly 2 a **3** jsou úvodem do teorie *počítání modulo*. Je zde jednak vybudována základní teorie, vedoucí k lineární algebře nad konečným tělesem a tudíž k *teorii lineárních kódů*, jednak jsou zde dokázány hlubší výsledky, vedoucí k *algoritmům pro počítání s velkými čísly* a k šifrovacímu *protokolu RSA*.

Teorie konečných těles, započatá v kapitole 2, je dovršena v **kapitole 4** a je aplikována na *cyklické kódy*.

Po prvních čtyřech kapitolách byste měli mít nepříjemný pocit, že se řada věcí v textu opakuje. Je to proto, že vlastně celou dobu studujeme nějakou *algebraickou strukturu*. V **kapitolách 5** a **6** se proto budeme algebraickým strukturám věnovat v plné obecnosti. Přesto se celou dobu budeme věnovat ryze praktické otázce jak *specifikovat datový typ*.

V **dodatku A** zmíníme klasické téma teorie čísel: *řetězové zlomky*, a použijeme je pro vysvětlení *Wienerova útoku na protokol RSA*.

Řada elementárních *vlastností prvočísel* a *testů prvočíselnosti* je uvedena v **dodatku B**.

Co v textu není

> Prodám úplný soubor *Encyclopædia Britannica*,
> nadále nepotřebný, protože manžel ví všechno.
> inzerát v *Lancashire Post*

V textu není jediná zmínka o **modelech výpočtu** (například o *Turingových strojích* nebo *λ-kalkulu*), o **teorii formálních jazyků** a s nimi spojených otázkách **algoritmické řešitelnosti** problémů. Standardní učebnicí těchto témat se stala kniha

☞ J. E. Hopcroft, R. Motwani a J. D. Ullman, *Introduction to Automata and Language Theory*, Addison-Wesley, New York, 2000

Pokud si o algoritmicky neřešitelných problémech chcete přečíst něco napsaného populární formou, pak doporučujeme útlou knížku

☞ D. Harel, *Computers Ltd., What They Really Can't Do*, Oxford University Press, 2000

nebo klasickou, vynikající a obsáhlou knihu

☞ D. R. Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, New York, 1979

která od doby svého vzniku vyšla v mnoha nových vydáních.

V textu dále není zmíněna **teorie složitosti** algoritmů. O té se je možno dočíst například v knihách

☞ J. L. Balcázar, J. Díaz a J. Gabarró, *Structural Complexity I, II*, Springer, Berlin, 1995

Otázku, zda mohou **počítače myslet** (a co by to vůbec mělo znamenat), probírají například knihy

☞ R. Penrose, *Emperor's New Mind: Concerning Computers, Minds and The Laws of Physics*, Oxford University Press, 1989

☞ R. Penrose, *Shadows of the Mind: A Search for the Missing Science of Consciousness*, Vintage Books, 1995

V těchto dvou knihách je důraz kladen na analýzu **Churchovy-Turingovy teze**[2] a **Gödelovy věty o neúplnosti**.[3] Pokud máte raději knihy více filosofické, pak kniha

☞ D. C. Dennet, *Consciousness Explained*, Penguin Books, 1993

je pro Vás tím pravým.

Řadu klasických a zásadních prací světových matematiků, opatřených komentářem od Stephena Hawkinga, si můžete vychutnat v knize

☞ S. Hawking (ed.), *God Created the Integers*, Penguin Books, 2006

V této knize naleznete (kromě řady dalších) originální práce Eukleida, Diophanta, Karla Friedricha Gausse, Georga Cantora, Kurta Gödela a Alana Mathisona Turinga, o kterých se v následujícím textu často zmiňujeme.

Chci jen udělat zkoušku. Co mám číst?

> Vracím tento kus papíru na psaní Vám, protože jej někdo celý popsal bláboly a na horní řádek napsal Vaše jméno.
>
> profesor angličtiny, Ohio University

Rada první: Zcela ignorujte dodatky A a B. Látka, zařazená do dodatků, se u zkoušky neobjeví.

Rada druhá: V každé kapitole se zaměřte na zvládnutí a pochopení následujících pojmů:

Kapitola 1: slabý a silný princip indukce, princip dobrého uspořádání, důkaz indukcí jako běh rekursivního algoritmu, variant rekursivního algoritmu, princip strukturální indukce.

Kapitola 2: binární relace a relace ekvivalence, rozdíl mezi dělením a dělitelností, rozšířený Eukleidův algoritmus, Bezoutova rovnost, kongruence modulo, pojmy okruhu a tělesa.

Kapitola 3: lineární rovnice v $\mathbb{Z}_m$, lineární algebra nad $\mathbb{Z}_m$, generující a kontrolní matice lineárního kódu, čínská věta o zbytcích, Eulerova věta, algoritmy pro zpracování velkých čísel (čínská věta a opakované čtverce), tvorba klíčů protokolu RSA, elementární útoky na RSA.

Kapitola 4: polynom jako výraz, rozšířený Eukleidův algoritmus pro polynomy, Bezoutova rovnost pro polynomy, kongruence modulo polynom, základní myšlenky teorie cyklických kódů.

Kapitola 5: myšlenka algebraické specifikace abstraktního datového typu, tabulka binární operace, grupoidy, pologrupy, monoidy, grupy, homomorfismy.

Kapitola 6: relace uspořádání, Hasseho diagramy, supremum a infimum, svazy, distributivní svazy, Booleovy algebry, vícesortová operace.

Rada třetí: Pokuste se zpracovat cvičení ke každé kapitole.

Rada čtvrtá: Zkouška nebude jen z početní techniky. Měli byste na jisté úrovni být schopni popsat i teoretické důvody, které zaručují správnost Vašeho řešení. Proto si zvykněte ke každému příkladu, který řešíte, přidat i drobný vysvětlující komentář.

[2] Tato teze zhruba tvrdí, že všechny modely pojmu algoritmus jsou navzájem ekvivalentní, viz 1.5.4. Pojmenovaná je podle matematiků Alonza Churche (1903–1995) a Alana Mathisona Turinga (1912–1954). Oba pány je právem možno považovat za jedny z otců moderní computer science. Alonzo Church vymyslel λ-*kalkulus*, který je teoretickým základem *funkcionálních programovacích jazyků*, jakým je například Lisp, Miranda nebo Haskell. Alan Turing v konceptu *Turingova stroje* popsal abstraktně, co je to *programovatelný počítač*. Obě práce vyšly v roce 1936.

[3] Věta brněnského rodáka Kurta Gödela (1906–1978), dokázaná v roce 1931, patří k nejzajímavějším výsledkům v matematice 20. století. Zhruba řečeno říká, že každý složitý systém (složitý=umožňující aritmetiku přirozených čísel) obsahuje věty, které jsou pravdivé, ale nedokazatelné. Zajímavé je, že pro důkaz své věty Gödel vlastně důvtipně formalizoval analogii *Epimenidova paradoxu* popisovaného například v Bibli (List Titovi, 1, verše 12–13): *Jeden z nich, jejich vlastní prorok, řekl: „Kréťané jsou samí lháři, zlá zvířata, lenivá břicha.“ A je to pravdivé svědectví.* Gödelův důkaz umožnuje sestavit analogické tvrzení: *Jsem nedokazatelná věta.*

Kapitola 1

Matematická indukce a rekurentní rovnice

„Co ti připomíná ta ryba?“
„Další rybu.“
„A co ti připomínají další ryby?“
„Jiné ryby.“

Joseph Heller, *Hlava XXII*

Dokazování indukcí má v computer science prominentní postavení. V této kapitole se nejprve seznámíme s přesnou formulací a použitím dvou základních indukčních principů — *slabého* (1.1.2 a 1.1.4) a *silného* (1.1.12). Později k nim přidáme ještě třetí princip: *princip dobrého uspořádání* (1.1.15) a ukážeme, že všechny tyto principy jsou navzájem ekvivalentní.

V odstavci 1.2 předvedeme použití principů indukce na problémy *designu* a *korektnosti* rekursivních algoritmů.

S principy indukce a otázkami složitosti algoritmů souvisejí *rekurentní rovnice*. Základům teorie rekurentních rovnic je věnován odstavec 1.3.

Konečně v odstavci 1.4 vysvětlíme, jak principy indukce použít k *induktivnímu zadávání* množin slov nad konečnou abecedou (tj. k popisu abstraktních jazyků).

Odstavec 1.5 *není* součástí přednášky: pojednává o aplikaci induktivního zadávání na teorii rekursivních funkcí.

1.1 Indukční principy

Indukční principy patří k základním důkazovým metodám matematiky. Jsou založeny na následující „konstruktivní“ představě množiny $\mathbb{N} = \{0, 1, \ldots\}$ přirozených čísel:

1. 0 je přirozené číslo.
2. Pokud je n přirozené číslo, je i $n + 1$ přirozené číslo.
3. Žádným jiným způsobem, než dvěma výše uvedenými, nelze přirozené číslo zkonstruovat.

Výše uvedeným třem bodům (neformálně) odpovídají tři části důkazu indukcí jistého tvrzení *podle přirozeného čísla* n:

1. Dokážeme, že tvrzení platí pro $n = 0$.
2. Dokážeme, že pokud tvrzení platí pro libovolné pevné n, potom platí i pro $n + 1$.
3. S použitím principu indukce prohlásíme, že jsme dané tvrzení dokázali pro *všechna přirozená čísla* $n \geq 0$.

Později uvidíme, že indukční principy (pro množinu $\mathbb{N}$) jsou ekvivalentní jistému tvrzení o (dobře známém) uspořádání $0 < 1 < 2 < \ldots$ množiny $\mathbb{N}$ (princip dobrého uspořádání, viz 1.1.15).

Podívejme se nejprve na typický příklad důkazu indukcí:

1.1.1 Příklad Dokažte tvrzení:

(*) Pro všechna přirozená čísla n platí rovnost $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$

Jako $T(n)$ nejprve označíme následující tvrzení o přirozeném čísle n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Toto značení nám dovolí úlohu (*) přepsat následujícím způsobem:

(**) Pro všechna přirozená čísla n platí $T(n)$.

Tvrzení (**) nyní dokážeme indukcí.

1. **Základní krok.** Nejprve ukážeme platnost $T(n)$ pro *nejmenší* možnou hodnotu n, tj. pro $n = 0$. Jinými slovy, máme dokázat, že platí rovnost $\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$. Ta skutečně platí.

2. **Indukční krok.** *Předpokládejme*, že $T(n)$ pro pevné (ale libovolné) přirozené číslo n *platí*. (Tomu říkáme *indukční předpoklad*.)

   Odvodíme, že platí i $T(n+1)$. Abychom ukázali platnost rovnosti $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$, napíšeme levou stranu následovně: $\sum_{i=0}^{n+1} i = \sum_{i=0}^{n} i + (n+1)$. (Tomu říkáme *dekomposice problému.*[1])

   Výše uvedená dekomposice nám totiž dovolí použít indukční předpoklad $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ a použitím (snadné) rovnosti $\frac{n(n+1)}{2} + (n+1) = \frac{(n+1)((n+1)+1)}{2}$, dostáváme $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$. A to je rovnost, kterou jsme chtěli dokázat.

Bod 1. nám říká, že platí $T(0)$. Z bodu 2. odvodíme platnost $T(1)$. Opětovným použitím bodu 2. dostáváme platnost $T(2)\ldots$

Pozor! Zatím nám nic nedovoluje prohlásit, že tvrzení

Pro *všechna* přirozená čísla n platí $T(n)$.

skutečně platí. Možnost učinit tento poslední krok je přesně tím, co nám princip indukce přináší (viz poznámky 1.1.5 a 1.1.20).

Nyní zformulujeme princip indukce přesně:

1.1.2 (Slabý princip indukce — varianta 1)
Ať T je vlastnost přirozených čísel a předpokládejme, že jsou splněny následující dvě podmínky:

(WIP1) Číslo 0 má vlastnost T.

(WIP2) Jestliže číslo n má vlastnost T, pak má vlastnost T i číslo $n+1$.

Pak mají všechna přirozená čísla n vlastnost T.

Výše uvedený princip je přesně tím, co jsme využili v důkazu tvrzení (*) z příkladu 1.1.1: příslušná vlastnost T (přirozeného čísla n) je platnost rovnosti $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

1.1.3 Příklad Dokažte následující tvrzení:

[1] K čemu dekomposice přesně slouží si povíme v poznámce 1.1.9.

(*) Pro všechna přirozená čísla $n \geq 4$ platí nerovnost $n! \geq 2^n$.

Nerovnost $n! \geq 2^n$ označíme jako $V(n)$. Potom lze (*) přeformulovat následovně:

(**) Pro všechna přirozená čísla $n \geq 4$ platí $V(n)$.

Tvrzení (**) dokážeme indukcí.

1. **Základní krok.** Ukážeme nejprve, že $V(n)$ platí pro *nejmenší* hodnotu n, která přichází v úvahu, tj. pro $n = 4$. To znamená dokázat nerovnost $4! \geq 2^4$, a to je triviální.

2. **Indukční krok.** *Předpokládáme*, že pro pevné (ale libovolné) $n \geq 4$ platí $V(n)$ a ukážeme, že platí i $V(n+1)$.

   Musíme tedy ukázat platnost nerovnosti $(n+1)! \geq 2^{n+1}$. *Dekomponujeme* levou stranu nerovnosti takto: $(n+1)! = n! \cdot (n+1)$. Použitím indukčního předpokladu $n! \geq 2^n$ a zřejmé nerovnosti $n+1 \geq 2$, dostaneme po vynásobení obou nerovností nerovnost $(n+1)! \geq 2^{n+1}$. A to jsme potřebovali.

Bod 1. říká, že platí $V(4)$. Podle bodu 2. víme, že platí $V(5)$. Opětovným použitím bodu 2. dostáváme platnost $V(6)\ldots$

Zatím nám opět nic nedovoluje prohlásit, že $V(n)$ platí pro všechna přirozená čísla $n \geq 4$ — tuto práci musí udělat indukční princip.

1.1.4 (Slabý princip indukce — varianta 2)

Ať n_0 je pevné přirozené číslo a ať V je vlastnost přirozených čísel $n \geq n_0$. Ať jsou dále splněny následující dvě podmínky:

(WIP1)' Číslo n_0 má vlastnost V.

(WIP2)' Jestliže číslo $n \geq n_0$ má vlastnost V, potom i číslo $n+1$ má vlastnost V.

Potom vlastnost V platí pro všechna přirozená čísla n, kde $n \geq n_0$.

1.1.5 Poznámka Mělo by být nyní zcela zřejmé, že indukční princip nám dovoluje se vyhnout nekonečné posloupnosti tvrzení a nabízí nám „zkratku“. Projděme znovu příklad 1.1.1: byli jsme schopni odvodit nekonečnou posloupnost pravdivých tvrzení

$$\begin{aligned}
\sum_{i=0}^{0} i &= \frac{0(0+1)}{2} \\
\sum_{i=0}^{1} i &= \frac{1(1+1)}{2} \\
\sum_{i=0}^{2} i &= \frac{2(2+1)}{2} \\
\sum_{i=0}^{3} i &= \frac{3(3+1)}{2} \\
&\vdots
\end{aligned}$$

Sestavili jsme (alespoň mentálně) výše uvedený nekonečný seznam (každý řádek je oindexován přirozeným číslem), kde každý řádek sám o sobě je pravdivý. Všimněme si totiž, že pro každý jednotlivý řádek máme důkaz pravdivosti. (Tyto důkazy jsou pro každý řádek různé. Napište si například důkazy řádků 5 a 8. Jsou stejné?)

Použití indukčního principu nám dovolí nekonečnou posloupnost důkazů „zabalit“ do jednoho důkazu a tvrdit tak, že *všechny* řádky jsou pravdivé.

Zajímavě o obtížnosti správného pochopení indukce píše *Gottfried Wilhelm Leibniz* v dopise pruské královně Žofii Charlottě (1702):

> ... Proto se geometři vždy domnívali, že všechno, co se v geometrii a aritmetice dokazuje pouze *indukcí* nebo příklady, není nikdy dokonale dokázáno. Tak nás například zkušenost učí, že lichá čísla, přičítáme-li je neustále k sobě, dávají po řadě čísla druhých mocnin, tj. čísla, jež vycházejí při násobení nějakého čísla jím samým. Tak $1+3=4$, tj. 2×2; $1+3+5$ dává 9, tj. 3×3; $1+3+5+7=16$, tj. 4×4; $1+3+5+7+9=25$, tj. 5×5. A stejným způsobem to jde dále. Kdybychom to stotisíckrát vyzkoušeli a pokračovali v počtu dosti daleko, mohli bychom rozumně usoudit, že výpočet musí *vždy* souhlasit. Bezpodmínečnou jistotu však přece mít nebudeme, dokud nepoznáme důkaz této zvláštnosti, který matematikové poznali již dávno.
>
> ... Ve skutečnosti jsou *experimenty*, které se nesčetně často a obyčejně zdaří, u nichž se přesto v určitých výjimečných případech shledá, že jsou *instance*, kdy se experiment nezdaří. Kdybychom například stotisíckrát vyzkoušeli, že kus železa, který je vhozen do vody, se potopí, nejsme si přece jisti, že tomu *musí* tak být vždy. Neboť aniž bychom se utíkali k zázraku proroka Eliáše, který nechal železo plavat, víme, že železnou nádobu lze vyhloubit tak, že plave po hladině, a že dokonce dokáže unést i značné břemeno, jako lodi z mědi a železného plechu.[2]

Celá tato poznámka může vyznívat triviálně. Později (poznámka 1.1.20) ale uvidíme, že indukční principy souvisejí velmi těsně s naší představou přirozených čísel.

Zformulovali jsme zatím dvě varianty indukčního principu. Nyní ukážeme, že jsou logicky ekvivalentní.

1.1.6 Tvrzení *Z druhé varianty slabého principu indukce plyne varianta první.*

Důkaz. Předpokládáme, že platí druhá varianta slabého principu indukce. Dále předpokládejme, ze je dána vlastnost T přirozených čísel, která splňuje podmínky (WIP1) a (WIP2) z 1.1.2, tj:

(WIP1) Číslo 0 má vlastnost T.

(WIP2) Jestliže n má vlastnost T, pak i číslo $n+1$ má vlastnost T.

Položme (ve značení z 1.1.4) $n_0 = 0$. Potom každé přirozené číslo $n \geq 0$ (tj. *každé* přirozené číslo n) má vlastnost T.

Proto platí první varianta slabého principu indukce. Důkaz je hotov. ■

1.1.7 Tvrzení *Z první varianty slabého principu indukce plyne varianta druhá.*

Důkaz. Předpokládáme, že platí první varianta slabého principu indukce. Zvolme libovolné přirozené číslo n_0. Ať je dána vlastnost V přirozených čísel $n \geq n_0$, která splňuje podmínky (WIP1)' a (WIP2)' z 1.1.4, tj.:

(WIP1)' Číslo n_0 má vlastnost V.

(WIP2)' Jestliže má n vlastnost V, potom i číslo $n+1$ má vlastnost V.

Chceme ukázat, že všechna přirozená čísla $n \geq n_0$ mají vlastnost V.

Za tím účelem definujeme novou vlastnost T, o které ukážeme, že splňuje podmínky (WIP1) a (WIP2) z 1.1.2:

> Přirozené číslo n má vlastnost T, právě tehdy, když přirozené číslo $n_0 + n$ má vlastnost V.

Výše uvedené T skutečně podmínky (WIP1) a (WIP2) z 1.1.2 splňuje:

(WIP1) Číslo 0 má vlastnost T, protože n_0 $(= n_0 + 0)$ má vlastnost V podle předpokladu.

(WIP2) Předpokládejme, že n má vlastnost T. Tudíž číslo $n_0 + n$ má vlastnost V, a podle podmínky (WIP2)' má i číslo $n_0 + (n+1)$ vlastnost V. To ale znamená, že $n+1$ má vlastnost V.

Podle první varianty slabého principu indukce mají všechna přirozená čísla vlastnot T. A proto mají vlastnost V všechna přirozená čísla $n \geq n_0$. To jsme chtěli dokázat. ■

[2] G. W. Leibniz, *Monadologie a jiné práce*, Svoboda, Praha, 1982, str. 140–141.

1.1.8 Důsledek *Obě varianty slabého principu indukce jsou navzájem ekvivalentní.*

Předchozí důsledek by neměl být šokující. Obě varianty slabého principu jsou si podobné jako vejce vejci: jedna varianta přechází na druhou pouhým „posunutím“ přirozených čísel o n_0 (přečtěte si znovu důkazy). Navíc mají obě varianty společný rys v tom, jakým způsobem „rozšiřujeme vědomosti“: víme, že něco platí pro přirozené číslo n a snažíme se to dokázat i pro číslo $n + 1$.

1.1.9 Poznámka Poznamenejme nyní, že i když v důkazu pomocí slabého principu indukce zdánlivě postupujeme „směrem nahoru“, tj. z n na $n + 1$, může nám tato představa způsobit komplikace. Než uvedeme příklad, připomeňme, že *strom* je souvislý graf bez kružnic, a že v každém stromu na alespoň dvou vrcholech existuje alespoň jeden vrchol stupně 1.

Chceme ukázat tvrzení

> Pro všechna $n \geq 1$ platí: každý strom na n vrcholech má $n - 1$ hran.

Základní krok indukce nečiní žádné potíže: existuje pouze jeden strom na jednom vrcholu a to je graf, který nemá žádné hrany.

Podle slabého principu indukce nyní máme ukázat, že libovolný strom T na $n + 1$ vrcholech má přesně n hran. Protože je $n + 1 \geq 2$, existuje v T alespoň jeden vrchol stupně 1. Strom T si tedy lze představit takto:

(1.1)

kde graf T' „uvnitř“ kružnice má přesně n vrcholů a jde o souvislý graf bez kružnic, neboli T' je strom na n vrcholech.

Tato představa stromu T nám dává okamžitě *dekomposici* problému a tím i představu o tom, jak zformulovat a použít indukční předpoklad:

1. Indukční přepoklad zní: každý strom na přesně n vrcholech má $n - 1$ hran.
2. K důkazu toho, že náš strom T má přesně n hran nyní využijeme dekomposici z (1.1): podle indukčního předpokladu má strom T' přesně $n - 1$ hran a strom T má o jednu hranu více než T', tedy n hran a to jsme chtěli dokázat.

Výše uvedené tvrzení o stromech nyní dokážeme ještě jednou, a to *špatně*. Nesprávnost důkazu spočívá ve špatném přístupu k indukčnímu kroku: myšlenka je *přidat* ke stromu T' na n vrcholech jeden vrchol a jednu hranu. Tím samozřejmě dostaneme *nějaký* strom na $n + 1$ vrcholech, který má pochopitelně n hran. Dokázali jsme indukční krok? Nikoli! Ještě bychom potřebovali ukázat, že *každý* strom T na $n+1$ vrcholech vznikl tímto způsobem. To nás ale vrací opět k dekomposici stromu T a tím i ke správnému důkazu indukčního kroku.

Dekomposice tedy slouží k explicitnímu popisu toho *jak* budeme indukční předpoklad *používat* (a leckdy je i návodem, *jak* indukční předpoklad *zformulovat*, viz příklad 1.1.11). V tomto smyslu si můžeme každý důkaz indukcí představit *dynamicky* jako chod rekursivního algoritmu. Více o tomto tématu řekneme v odstavci 1.2.

1.1.10 Poznámka První variantu slabého principu indukce nyní přepíšeme do tvaru, který je v moderní computer science obvyklý. Všimněte si, že tento přepis *opět* ukazuje, že indukce pracuje jako rekursivní algoritmus, tj. směrem dolů.

Především si uvědomme, že každou vlastnost T přirozených čísel můžeme ztotožnit s množinou A těch přirozených čísel, které mají vlastnost T. Obráceně, každá množina A přirozených čísel určuje vlastnost T: číslo n má vlastnost T právě tehdy, když $n \in A$.

Pro každou podmnožinu A definujeme novou podmnožinu $\mathsf{next\_time\_in}(A)$ takto:

$$\mathsf{next\_time\_in}(A) = \{n \mid \text{ pokud } n - 1 \text{ existuje, pak } n - 1 \in A\ \}$$

Vhodný pohled na množinu $\mathbb{N}$ přirozených čísel je jako na *jednoduchý automat*, jehož stavy jsou jednotlivá přirozená čísla a který v „jednom výpočetním kroku“ je schopen přejít ze stavu n do stavu $n - 1$ a ve stavu 0 je v *deadlock* stavu. Množina $\mathsf{next\_time\_in}(A)$ je potom ta množina stavů, z nichž se lze v jednom výpočetním kroku dostat do množiny A, obohacená o stav 0.

Slabý princip indukce potom říká následující:

Jestliže platí $\mathsf{next\_time\_in}(A) \subseteq A$, *potom platí* $A = \mathbb{N}$.

Přiřazení $A \mapsto \mathsf{next\_time\_in}(A)$ je příkladem *modality* na podmnožinách přirozených čísel. Více se o modalitách a jejich použití v computer science lze dočíst například v textu

☞ J. Velebil, *Logika programů*, http://math.feld.cvut.cz/velebil/

Viz také cvičení 1.6.13.

Viděli jsme, že při důkazu indukcí je podstatné najít správnou *dekomposici problému*. V dalším příkladu uvidíme, že dekomposice typu

problém velikosti $n+1$ redukuj na problém velikosti n

není vždy možná (viz také problém egyptských forem v odstavci 1.2).

1.1.11 Příklad Připomeňme, že přirozenému číslu p říkáme *prvočíslo*, pokud platí $p > 1$ a pokud p je dělitelné pouze čísly 1 a p. Přirozeným číslům $x > 1$, která nejsou prvočísly, říkáme *složená čísla*.

Prvočíselný rozklad přirozeného čísla x je zápis

$$x = p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r},$$

kde $r \geq 1$ je přirozené číslo, $p_1 < p_2 < \ldots < p_r$ jsou prvočísla a $n_1, n_2, \ldots, n_r$ jsou kladná přirozená čísla.

Dokážeme následující tvrzení:

(*) Každé přirozené číslo $x \geq 2$ má prvočíselný rozklad.

Pokusme se o důkaz (*) slabým principem indukce.

1. **Základní krok.** Pro $x = 2$ je zřejmě zápis $2 = 2^1$ hledaný prvočíselný rozklad.
2. **Indukční krok.** Indukční předpoklad zní, že číslo x má prvočíselný rozklad. Hledáme prvočíselný rozklad čísla $x+1$. Nastane jeden ze dvou případů:
   (a) $x+1$ je prvočíslo, řekněme p. Potom $x+1 = p^1$ je hledaný prvočíselný rozklad.
   (b) $x+1$ je složené číslo. Pak existují dvě přirozená čísla řekněme a a b tak, že platí $x+1 = a \cdot b$ a $1 < a < x+1$ a $1 < b < x+1$. Kdyby náš indukční předpoklad zaručoval prvočíselné rozklady čísel a a b, nalezli bychom jistě i prvočíselný rozklad čísla $x+1$. Náš indukční předpoklad však zaručuje prvočíselný rozklad pouze pro číslo x a tento rozklad nám v řešení problému nepomůže.

Námi zformulovaný indukční předpoklad nám nedovolil tvrzení (*) dokázat. Dekomposice problému však naznačila, že máme zkusit zformulovat indukční předpoklad *silněji*: prvočíselný rozklad má existovat pro všechna přirozená čísla k, pro která platí $2 \leq k < x+1$. Důkaz (*) *silným principem indukce* vypadá takto:

1. **Základní krok.** Pro $x = 2$ je zřejmě zápis $2 = 2^1$ hledaný prvočíselný rozklad.
2. **Indukční krok.** Předpokládáme, že prvočíselný rozklad existuje pro všechna přirozená čísla k, pro která platí $2 \leq k < x+1$. Nastane jeden ze dvou případů:
   (a) $x+1$ je prvočíslo, řekněme p. Potom $x+1 = p^1$ je hledaný prvočíselný rozklad.
   (b) $x+1$ je složené číslo. Pak existují dvě přirozená čísla řekněme a a b tak, že platí $x+1 = a \cdot b$ a $1 < a < x+1$ a $1 < b < x+1$.
   Podle indukčního předpokladu existuje prvočíselný rozklad
   $$a = p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r},$$
   a prvočíselný rozklad
   $$b = q_1^{m_1} \cdot q_2^{m_2} \cdot \ldots \cdot q_s^{m_s}.$$
   Vynásobte tyto dva prvočíselné rozklady. Po případném přesunutí jednotlivých faktorů dostáváme prvočíselný rozklad čísla $x+1 = a \cdot b$.

Jako vždy, závěr, že *každé* přirozené číslo $x \geq 2$ má prvočíselný rozklad, nám umožní až (silný) princip indukce.

Silný princip indukce nyní zformulujeme přesně:

1.1.12 (Silný princip indukce)
Ať n_0 je přirozené číslo a ať W je vlastnost přirozených čísel $n \geq n_0$. Dále předpokládejme, že jsou splněny následující dvě podmínky:

(SIP1) Číslo n_0 má vlastnost W.

(SIP2) Jestliže všechna přirozená čísla k, $n_0 \leq k < n+1$, mají vlastnost W, pak $n+1$ má vlastnost W.

Potom všechna přirozená čísla $n \geq n_0$ mají vlastnost W.

Znovu zdůrazněme základní rozdíl mezi silným a slabým principem indukce: v silném principu není „skok“ z n na $n+1$. Chceme-li silným principem indukce ukázat platnost vlastnosti W pro číslo $n+1$, vyžadujeme platnost vlastnosti W pro *všechna* přirozená čísla k, pro která je $n_0 \leq k < n+1$.

1.1.13 Tvrzení *Ze silného principu indukce plyne slabý.*

Důkaz. Předpokládejme, že platí silný princip indukce. Zvolme libovolné přirozené číslo n_0 a vlastnost V přirozených čísel $n \geq n_0$. Dále předpokládejme, že V splňuje podmínky (WIP1)’ a (WIP2)’ z 1.1.4, tj.:

(WIP1)’ Číslo n_0 má vlastnost V.

(WIP2)’ Jestliže $n \geq n_0$ má vlastnost V, potom i $n+1$ má vlastnost V.

Chceme ukázat, že všechna $n \geq n_0$ mají vlastnost V. Stačí ukázat, že V splňuje podmínky (SIP1) a (SIP2) z 1.1.12.

(SIP1) V evidentně splňuje (SIP1), protože n_0 má podle předpokladů vlastnost V.

(SIP2) Předpokládejme, že všechna k, kde $n_0 \leq k < n+1$, mají vlastnost V. Potom pro $k = n$ víme, že n má vlastnost V. Protože o V předpokládáme (WIP2)’, plyne, že $n+1$ má vlastnost V.

Podle silného principu indukce mají všechna přirozená čísla $n \geq n_0$ vlastnost V. Důkaz je ukončen. ■

Platnost výše uvedeného tvrzení jsme očekávali: říká, že silný princip indukce je silnější než slabý princip indukce. Překvapivé proto může být následující tvrzení:

1.1.14 Tvrzení *Ze slabého principu indukce plyne silný princip indukce.*

Důkaz. Předpokládejme, že platí slabý princip indukce. Zvolme libovolné přirozené číslo n_0 a vlastnost W přirozených čísel $n \geq n_0$. Dále předpokládejme, že W splňuje podmínky (SIP1) a (SIP2) z 1.1.12, tj.:

(SIP1) Číslo n_0 má vlastnost W.

(SIP2) Jestliže všechna k, kde $n_0 \leq k < n+1$, mají vlastnost W, pak $n+1$ má vlastnost W.

Chceme ukázat, že všechna $n \geq n_0$ mají vlastnost W.
Stačí ukázat, že vlastnost W splňuje podmínky (WIP1)’ a (WIP2)’ z 1.1.4.

(WIP1)’ W samozřejmě splňuje (WIP1)’, protože n_0 má vlastnost W.

(WIP2)’ Předpokládejme, že existuje $n \geq n_0$ tak, že n má vlastnost W, ale číslo $n+1$ vlastnost W nemá.

Mezi čísly n_0, n_0+1, …, n musí být alespoň jedno (označme jej k) tak, že k nemá vlastnost W. Vskutku, kdyby tomu bylo naopak, pak by číslo $n+1$ muselo, podle podmínky (SIP2), mít vlastnost W.

Vyberme nejmenší takové k a označme jej k_0 (takové k_0 můžeme samozřejmě najít — množina $\{n_0, n_0+1, \ldots, n\}$ je konečná a neprázdná).

Nyní víme, že všechna přirozená čísla l, kde $n_0 \leq l < k_0$, mají vlastnost W. Podle (SIP2) musí i číslo k_0 mít vlastnost W. To je ovšem ve sporu s naší volbou k_0.

Ukázali jsme, že W splňuje podmínku (WIP2)’.

Podle slabého principu indukce mají všechna přirozená čísla $n \geq n_0$ vlastnost W. ■

Povšimněte si, že v předchozím důkazu jsme použili (triviálního) faktu, že každá konečná neprázdná množina přirozených čísel má nejmenší prvek. Mají *všechny* neprázdné podmnožiny $\mathbb{N}$ nejmenší prvek? Ukazuje se, že kladná odpověď je variantou indukčního principu! (Srovnejte s axiomem výběru v poznámce 6.3.7.)

1.1.15 (Princip dobrého uspořádání)
Každá neprázdná podmnožina množiny přirozených čísel má nejmenší prvek.

Princip dobrého uspořádání lze zformulovat i jinak: tato formulace má velký význam v computer science (viz pojem *variantu* v odstavci 1.2).

1.1.16 Tvrzení Princip dobrého uspořádání je ekvivalentní tvrzení: v množině $\mathbb{N}$ neexistuje nekonečná klesající posloupnost $x_0 > x_1 > x_2 > \ldots$.
Důkaz. Předpokládejme, že taková nekonečná klesající posloupnost existuje a označme $M = \{x_0, x_1, x_2, \ldots\}$. Potom M je neprázdná množina přirozených čísel a podle principu dobrého uspořádání má nejmenší prvek, řekněme x_k. To ale není možné, protože platí nerovnost $x_k > x_{k+1}$. Takže nekonečná klesající posloupnost přirozených čísel nemůže existovat.

Obráceně: chceme ukázat, že princip dobrého uspořádání plyne z neexistence nekonečných klesajících posloupností. Zvolme jakoukoli neprázdnou množinu M přirozených čísel. Pokud je M konečná množina, má M jistě nejmenší prvek. Ať je tedy množina M nekonečná. Kdyby M neměla nejmenší prvek, dovedeme sestrojit nekonečnou klesající posloupnost $x_0 > x_1 > x_2 > \ldots$ prvků M. To ale není možné, proto má M nejmenší prvek. ■

1.1.17 Tvrzení *Ze silného principu indukce plyne princip dobrého uspořádání.*

Důkaz. Předpokládejme, že máme množinu X s následující vlastností: X je podmnožina $\mathbb{N}$, která nemá nejmenší prvek. Chceme ukázat, že $X = \emptyset$

Definujeme $Y = \mathbb{N} \setminus X$ a silným principem indukce ukážeme, že $Y = \mathbb{N}$. Tím bude důkaz hotov.

- $0 \in Y$. V opačném případě je totiž $0 \in X$ a X by měla nejmenší prvek (totiž 0).
- Předpokládejme, že $k \in Y$ pro všechna k, kde $0 \leq k < n+1$. Kdyby platilo $n+1 \notin Y$, tak by číslo $n+1$ bylo nejmenším prvkem X. Proto musí platit $n+1 \in Y$.

Podle silného principu indukce platí $Y = \mathbb{N}$. ■

1.1.18 Tvrzení *Z principu dobrého uspořádání plyne silný princip indukce.*

Důkaz. Předpokládejme, že X je podmnožina $\mathbb{N}$, splňující následující dvě podmínky:

1. 0 je prvkem X.
2. Jestliže pro každé k, kde $0 \leq k < n+1$, platí $k \in X$, pak platí $n+1 \in X$.

Chceme dokázat, že platí rovnost $X = \mathbb{N}$. Postupujeme sporem: ať $X \neq \mathbb{N}$. Potom je množina $Y = \mathbb{N} \setminus X$ neprázdná a podle principu dobrého uspořádánání má nejmenší prvek, řekněme číslo y. Protože $0 \in X$, musí platit $y > 0$. Navíc, všechna přirozená čísla menší než y musí být prvky X. Podle bodu 2. platí, že $y \in X$ — to je spor. ■

Úplný obrázek principů indukce na $\mathbb{N}$ tedy je:

1.1.19 Důsledek *Následující principy jsou ekvivalentní:*

1. *Slabý princip indukce.*

2. *Silný princip indukce.*

3. *Princip dobrého uspořádání.*

1.1.20 Poznámka Nyní můžeme podrobněji vysvětlit poznámku 1.1.5: předchozí důsledek nám říká, že pokud chceme přijmout jeden z indukčních principů, *musíme* přijmout všechny a *musíme* přijmout fakt, že pracujeme s dobře uspořádanými přirozenými čísly (každá neprázdná podmnožina má nejmenší prvek).

Obráceně, pokud jsme přijali dobré uspořádání množiny $\mathbb{N}$, musíme přijmout principy indukce jako důkazovou techniku!

Je samozřejmě myslitelné pracovat v nestandardním modelu přirozených čísel. Představte si situaci, kdy chceme axiomatizovat nekonečně veliká přirozená čísla. (Standardní model žádná nekonečně veliká přirozená čísla *neobsahuje*.) To znamená, že k již existujícím přirozeným číslům chceme přidat „nová přirozená čísla“ a každé nové číslo K by mělo mít tuto vlastnost:

$K > n$ pro všechna „stará“ přirozená čísla n.

To není šílená myšlenka, ve skutečnosti právě mluvíme o základech *nestandardní analýzy* — té části matematiky, která studuje infinitesimální kalkulus v původních intencích Isaaca Newtona a Gottfrieda Wilhelma Leibnize.[3]

V takovém nestandardním světě přirozených čísel musí být člověk na indukční principy velmi opatrný. Označme „rozšířenou“ množinu nestandardních přirozených čísel jako ${}^*\mathbb{N}$. I když jsme nepodali exaktní definici množiny ${}^*\mathbb{N}$, mělo by být intuitivně zřejmé, že množina ${}^*\mathbb{N}\setminus\mathbb{N}$ je neprázdná a přesto nemá nejmenší prvek. Je neprázdná, protože jsme postulovali existenci alespoň jednoho $K \in {}^*\mathbb{N}\setminus\mathbb{N}$. Kdyby ${}^*\mathbb{N}\setminus\mathbb{N}$ měla nejmenší prvek, řekněme K, pak by pro všechna $n \in \mathbb{N}$ musela platit nerovnost $K - 1 > n$ a tudíž by platilo $K - 1 \in {}^*\mathbb{N} \setminus \mathbb{N}$, což je spor.

Více se lze dočíst například v knize

☞ M. Davis, *Applied Nonstandard Analysis*, John Wiley, New York, 1977

Viz také poznámku 6.3.7.

1.2 Principy indukce a rekursivní algoritmy

V tomto (krátkém) odstavci připomeneme jiný pohled na principy indukce. Půjde nám o vybudování analogie mezi indukcí a chováním (terminací a korektností) rekursivních algoritmů.[4] Každému rekursivnímu algoritmu, který svoji práci ukončí, lze přiřadit parametr, který ukončení algoritmu (terminaci) zaručí. Tomuto parametru říkáme *variant*. Pro jednoduchost předpokládejme, že tento parametr je nějaké přirozené číslo n. Kdykoli to nepovede k nedorozumění, budeme číslu n také říkat *rozměr řešeného problému*.

Typické rysy, které terminující rekursivní algoritmus má, jsou:

1. Existuje *nejmenší* rozměr problému, pro který algoritmus nepracuje rekursivně. (Ve skutečnosti může být těchto „nejmenších“ rozměrů několik, ale zatím situaci nekomplikujme.)

2. Problémy větších rozměrů jsou nejprve *dekomponovány* na problémy menších rozměrů a poté je vykonána rekurse.

[3] V matematice jsou Gottfried Wilhelm von Leibniz (1646–1716) a sir Isaac Newton (1643–1727) známí především svými pracemi z diferenciálního a integrálního počtu. Newton napsal v říjnu 1666 traktách o *fluxích*, tj. o kvantitách, které „plynou“ v čase. Značení x' pro derivaci (u Newtona podle času) pochází z tohoto článku. Leibniz přemýšlel o derivacích a integrálech jinak: v jeho pracech se objevují značky dy a dx pro přírůstky hodnot. 21. listopadu 1675 použil Leibniz ve svém článku poprvé notaci $\int$ (protáhlé S, jako *suma*) pro integrál. Oba muži se na sklonku svých kariér dostali kvůli téměř totožným výsledkům do hořkých sporů o prvenství. Oba se však zapsali do historie nesmazatelně: Newton svým dílem *Philosophiae naturalis principia mathematica* (1687), kde je moderním matematickým způsobem vyložena mechanika těles (ostatně na jeho náhrobku ve Westminster Abbey stojí *Who, by vigor of mind almost divine, the motions and figures of the planets, the paths of comets, and the tides of the seas first demonstrated.*), Leibniz sestrojil první mechanický kalkulátor (který uměl násobit a dělit) a vyvinul moderní formu binárního zápisu.

[4] Slovo *algoritmus* používáme na počest perského matematika jménem Abu Ja'far Mohammed ibn Mûsa *al-Khowârizm*, který přibližně v roce 825 sepsal dílo *Kitab al-jabr wa'l-muquabala*. V této knize o aritmetice al-Khowârizm vysvětluje výhody pozičního decimálního zápisu (převzatého z indických prací — viz stranu 70), zavádí konvenci shlukování cifer do trojic a popisuje algoritmus pro násobení čísel, viz cvičení 1.6.15. Stojí za zmínku, že arabské *al-jabr* z názvu této knihy obohatilo náš jazyk o jiné důležité slovo — *algebra*. Česky slovo *al-jabr* znamená *spojení zlomených částí* — v tomto smyslu se dodnes ve španělštině používá výraz *algebrista* pro osobu léčící zlomeniny: viz například kapitolu XV *dona Quijota* od Miguela de Cervantese (1547–1616).

Co to má společného s principy indukce? Představme si následující algoritmus (zapsaný v pseudokódu):

```
procedure FAC(n: natural): natural;
begin
if n = 0 then FAC := 1; stop
         else FAC := n*FAC(n-1)
end
```

Určitě jste rozpoznali rekursivní výpočet faktoriálu.

Variantem je číslo `n` (to je čistá shoda náhod, později uvidíme, že variant nemá s hodnotou vstupních dat nic společného). Algoritmus pracuje nerekursivně pro `n=0` a pro větší hodnoty nejprve „dekomponuje“ (řešení problému `n` je redukováno na řešení problému velikosti `n-1`).

Použití variantu a principu indukce často dovolí dokázat *totální korektnost* algoritmu. Totální korektností rozumíme splnění dvou podmínek:

1. Terminace: algoritmus skončí svůj běh pro libovolná vstupní data.
2. Parciální korektnost: jestliže algoritmus skončí, potom skončí korektně. (Spočítá to, co má spočítat.)

Totální korektnost výše uvedeného algoritmu pro výpočet faktoriálu dokážeme následovně:

1. Terminaci zajistí hodnoty variantu v $\mathbb{N}$, viz tvrzení 1.1.16.
2. Parciální korektnost dokážeme takto: stačí indukcí ukázat rovnost `FAC(n)` $= n!$ pro všechna $n \geq 0$.
   (a) **Základní krok.** Zřejmě platí `FAC(0)` $= 1 = 0!$.
   (b) **Indukční krok.** Přepokládejme, že platí `FAC(n)` $= n!$. Nejprve využijeme dekomposici `FAC(n+1)` $=$ `(n+1)*FAC(n)` danou algoritmem. Použitím indukčního předpokladu dostáváme rovnost `FAC(n+1)` $= (n+1)!$.

   Rovnost `FAC(n)` $= n!$ pro všechna n dostáváme použitím slabého principu indukce.

Jiný přístup k dokazování parciální korektnosti rekursivního algoritmu, založený na pojmu *invariant*, zmíníme v kapitole 2, viz také příklad 2.1.18.

1.2.1 Poznámka Otázka terminace rekursivního algoritmu může být velmi složitá i pro zdánlivě nevinný algoritmus. O následujícím algoritmu (opět zapsaném v pseudokódu)

```
while x > 1 do
      if even(x) then x:=x/2 else x:=3*x+1 endif
```

se *neví*, zda pro každou přirozenou hodnotu `x` svou práci skončí či ne. Tomuto problému se říká *Collatzův problém*. Bylo dokázáno, že výše uvedený algoritmus se zastaví pro všechny počáteční hodnoty identifikátoru `x` menší nebo rovny číslu $3 \cdot 2^{53}$ (stav z roku 1999). Vyzkoušejte si běh tohoto algoritmu pro některé hodnoty `x`. Viz také například

☞ http://mathworld.wolfram.com/CollatzProblem.html

Pokračujme komplikovanějším příkladem, ve kterém *navrhneme* rekursivní algoritmus a použijeme princip indukce k důkazu jeho korektnosti. Začneme definicí:

1.2.2 Definice Zlomek $\frac{p}{q}$ je napsán v *egyptské formě*, pokud existuje seznam navzájem různých přirozených čísel $n_1, n_2, \ldots, n_k$ takových, že platí rovnost

$$\frac{p}{q} = \frac{1}{n_1} + \frac{1}{n_2} + \ldots + \frac{1}{n_k}$$

Například zlomek $\frac{8}{15}$ má egyptskou formu $\frac{1}{5}+\frac{1}{3}$. Pozor: zápis $\frac{1}{3}+\frac{1}{3}$ *není* egyptskou formou zlomku $\frac{2}{3}$, protože v definici 1.2.2 musí být jmenovatelé navzájem různá přirozená čísla.

Hodláme nyní vyřešit následující problém:

Dokažte, že každý zlomek $\frac{p}{q}$ mezi 0 a 1, kde p a q jsou přirozená čísla, lze napsat v egyptské formě.[5]

Problém vyřešíme návrhem (terminujícího a korektního) rekursivního algoritmu, který pro vstupní data $\frac{p}{q}$ vrátí egyptskou formu.

Protože zatím nemáme nejmenší představu, jak by algoritmus měl fungovat, vyzkoušíme si nejprve jak najít egyptskou formu zlomku $\frac{7}{19}$ hrubou silou:

1. Nejprve najdeme největší číslo tvaru $\frac{1}{n}$, pro které je $\frac{1}{n}\leq\frac{7}{19}$. Snadno zjistíme, že n je rovno 3.
2. Spočítáme $\frac{7}{19}-\frac{1}{3}=\frac{2}{57}$ a postupujeme rekursivně podle bodu 1. Jako n nyní musíme zvolit 29. Protože platí $\frac{2}{57}-\frac{1}{29}=\frac{1}{1653}$, zastavujeme rekursi a hledaná egyptská forma je

$$\frac{7}{19}=\frac{1}{3}+\frac{1}{29}+\frac{1}{1653}$$

Měli jsme štěstí a jednu instanci problému jsme vyřešili. Není ovšem jasné, zda se rekurse zastaví pro obecná vstupní data. Musíme nejprve zodpovědět tuto otázku: co je „velikostí“ problému, který máme řešit? (Neboli: co je variant?)

Pro kterákoli ze vstupních dat $\frac{1}{21}$, $\frac{1}{234}$ nebo $\frac{1}{7664}$ algoritmus zjevně nepracuje rekursivně. To naznačuje, že velikost těchto problémů by měla být totožná. Za variant zvolíme čitatel p zlomku $\frac{p}{q}$ a číslo 1 je jeho nejmenší hodnota. (To ilustruje fakt, že variant obecně s velikostí vstupních dat nesouvisí.) Ve výše uvedeném příkladu získáme dekomposici odečtením zlomku $\frac{1}{3}$ a vskutku: variant se zmenšil z hodnoty 7 („původní velikost“) na hodnotu 2 („velikost po dekomposici“). Pokud se nám povede dokázat, že se čitatel po každém odečtení zmenší, dokážeme přinejmenším terminaci našeho algoritmu. Použitím (evidentně) silného principu indukce pak dokážeme korektnost algoritmu.

Předpokládejme tedy, že je dán zlomek $\frac{p}{q}$ mezi 0 a 1. Nejprve ukážeme, že existuje kladné přirozené číslo n_1 tak, že platí

$$\frac{1}{n_1}\leq\frac{p}{q}<\frac{1}{n_1-1}.$$

Možná překvapivě to lze dokázat principem indukce (přesněji: principem dobrého uspořádání). Množina $A=\{n\geq 1 \mid \frac{1}{n}\leq\frac{p}{q}\}$ je totiž neprázdná, protože posloupnost $\left\{\frac{1}{n}\right\}_{n=1}^{\infty}$ konverguje k 0. Zde využíváme nerovnosti $\frac{p}{q}>0$ a definici limity. Podle principu dobrého uspořádání má množina A nejmenší prvek, řekněme n_1. Protože n_1 patří do A, platí nerovnost $\frac{1}{n_1}\leq\frac{p}{q}$. Dále platí nerovnost $\frac{p}{q}<\frac{1}{n_1-1}$, protože n_1 je nejmenším prvkem množiny A (všimněte si, že n_1 nemůže být rovno 1, protože platí $\frac{p}{q}<1$).

Algoritmus zastavíme v případě, že platí rovnost $\frac{1}{n_1}=\frac{p}{q}$. V opačném případě spočítáme rozdíl

$$\frac{p}{q}-\frac{1}{n_1}=\frac{n_1\cdot p-q}{q\cdot n_1}.$$

[5] Místo „zlomek“ bychom mohli říci „racionální číslo“. Připomeňme, že racionální čísla jsou ty zlomky $\frac{p}{q}$, kde p a q jsou nesoudělná. Naše formulace však vede k řešení, kdy není třeba vyšetřovat řadu speciálních případů.

Především je zřejmé, že zlomek $\frac{n_1 \cdot p - q}{q \cdot n_1}$ má opět hodnotu mezi 0 a 1 a obě hodnoty $n_1 \cdot p - q$ a $q \cdot n_1$ jsou kladné. Potřebujeme dokázat nerovnost $n_1 \cdot p - q < p$, což plyne z nerovnosti $\frac{p}{q} < \frac{1}{n_1 - 1}$.

Náš algoritmus tedy nemůže běžet věčně. Počítá ale skutečně egyptskou formu? Pro důkaz zkusíme použít silný princip indukce. Postupujeme indukcí podle $p \geq 1$.

1. **Základní krok.** Pro $p = 1$ máme egyptskou formu.

2. **Indukční krok.** Předpokládejme, že pro každý přípustný zlomek $\frac{k}{q'}$, kde $1 \leq k < p + 1$ (q' libovolné), počítá náš algoritmus egyptskou formu zlomku $\frac{k}{q}$. Hledáme egyptskou formu zlomku $\frac{p+1}{q}$.

   Stejně jako výše, nalezneme číslo n_1. Musíme rozlišit dva případy:

   $\frac{p+1}{q} = \frac{1}{n_1}$. V tomto případě algoritmus končí a my máme egypstkou formu.

   $\frac{p+1}{q} > \frac{1}{n_1}$. V tomto případě provádí algoritmus dekomposici $\frac{p+1}{q} - \frac{1}{n_1} = \frac{k}{q'}$. Víme, že platí $k < p + 1$ a proto máme egyptskou formu

   $$\frac{1}{m_1} + \frac{1}{m_2} + \ldots + \frac{1}{m_r}$$

   zlomku $\frac{k}{q'}$. Zbývá dokázat, že se číslo n_1 nemůže rovnat žádnému z čísel $m_1, \ldots, m_r$. To okamžitě plyne z nerovnosti $\frac{k}{q'} = \frac{p+1}{q} - \frac{1}{n_1} < \frac{1}{n_1}$, která je ekvivalentní nerovnosti $\frac{p+1}{q} < \frac{2}{n_1}$. Pro $n_1 = 2$ je poslední nerovnost triviální a pro $n_1 > 2$ platí nerovnost $\frac{1}{n_1 - 1} < \frac{2}{n_1}$. Hledaná nerovnost plyne z nerovnosti $\frac{p+1}{q} < \frac{1}{n_1 - 1}$.

Algoritmus pro hledání egyptské formy je korektní.

Metoda předchozího příkladu je typická. Vyzkoušejte si ji na svém oblíbeném rekursivním algoritmu. K rekursivním algoritmům se ještě vrátíme při úvahách o dělitelnosti (např. poznámka 2.1.17 a odstavec 4.2). Viz také příklad 1.3.12 a cvičení 1.6.20.

1.3 Lineární rekurentní rovnice

Při zkoumání složitostí rekursivních algoritmů se často objevují rekurentní rovnice. Uveďme nejprve typický příklad rekurentní rovnice.

1.3.1 Příklad Označme pro $n \geq 1$ jako $S(n)$ počet řetězců délky n utvořených ze znaků `0` a `1`, které neobsahují `00` jako podřetězec. Hledáme vzorec pro číslo $S(n)$, kde $n \geq 1$.

Explicitní vzorec pro číslo $S(n)$ zatím nemáme, lze si však všimnout následujících vztahů:

1. Je-li $n = 1$, je $S(n) = 2$. Oba možné řetězce délky 1 — totiž řetězce `0` a `1` — neobsahují `00` jako podřetězec.

2. Je-li $n = 2$, je $S(n) = 3$. Ze čtyř možných řetězců délky 2 pouze tři neobsahují `00` jako podřetězec. Jde o řetězce `11`, `01` a `10`.

3. Je-li $n > 2$, pak každý řetězec délky n, který neobsahuje `00` jako podřetězec, má buď `1` nebo `01` jako prefix. Proto platí $S(n) = S(n-1) + S(n-2)$.

Tato analýza problému nám dává takzvaný *rekurentní vztah* pro výpočet čísla $S(n)$:

$$\begin{aligned} S(1) &= 2, \\ S(2) &= 3, \\ S(n+2) &= S(n+1) + S(n), \text{ pro } n \geq 1 \end{aligned} \tag{1.2}$$

Není na první pohled zřejmé, jak explicitní vzorec pro $S(n)$ vypadá.

Matematickou indukcí se lze přesvědčit, že platí

$$S(n) = \frac{5+3\sqrt{5}}{2} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{5-3\sqrt{5}}{2} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n, \text{ pro } n \geq 1.$$

Předchozí příklad ukazuje, že explicitní řešení na pohled velmi jednoduché rekurentní rovnice může být velmi složité. V tomto odstavci podáme návod k řešení velké třídy rekurentních rovnic. Teorie řešení rekurentních rovnic je velmi podobná teorii řešení obyčejných diferenciálních rovnic. Protože vybudování potřebného aparátu však přesahuje rámec těchto poznámek, budeme všechna tvrzení uvádět bez důkazu. Zájemce odkazujeme na seznam literatury, uvedený na konci kapitoly.

1.3.2 Definice *Lineární rekurentní rovnice k-tého řádu s konstantními koeficienty* je zápis:

$$X(n+k) + c_1 \cdot X(n+k-1) + c_2 \cdot X(n+k-2) + \ldots + c_k \cdot X(n) = F(n), \ n \geq n_0 \tag{1.3}$$

kde $k \geq 1$ je pevné přirozené číslo, c_1, c_2,..., c_k jsou reálná čísla a $c_k \neq 0$.

Zápis

$$X(n+k) + c_1 \cdot X(n+k-1) + c_2 \cdot X(n+k-2) + \ldots + c_k \cdot X(n) = 0, \ n \geq n_0 \tag{1.4}$$

nazveme *homogenní rovnice příslušející k rovnici (1.3)*.

Charakteristická rovnice pro (1.4) je rovnice

$$\lambda^k + c_1 \cdot \lambda^{k-1} + c_2 \cdot \lambda^{k-2} + \ldots + c_k = 0 \tag{1.5}$$

1.3.3 Příklad Připomeňme rekurentní rovnici z příkladu 1.3.1:

$$\begin{aligned} S(1) &= 2, \\ S(2) &= 3, \\ S(n+2) &= S(n+1) + S(n), \text{ pro } n \geq 1 \end{aligned}$$

Je zřejmé, že poslední rovnici lze přepsat do následující ekvivalentní podoby:

$$S(n+2) - S(n+1) - S(n) = 0, \text{ pro } n \geq 1 \tag{1.6}$$

Jde tedy o homogenní rekurentní rovnici druhého řádu s konstantními koeficienty. Její charakteristická rovnice má tvar:

$$\lambda^2 - \lambda - 1 = 0 \tag{1.7}$$

1.3.4 Definice *Řešením rovnice (1.3)* (resp. rovnice (1.4)) je jakákoli posloupnost a_{n_0}, a_{n_0+1}, a_{n_0+2},..., (reálných nebo komplexních) čísel, která po dosazení $X(n) := a_n$ $(n \geq n_0)$ splňuje rovnost (1.3) (resp. rovnici (1.4)).

Hledání obecného řešení lineární rekurentní rovnice bude probíhat podobně jako hledání lineární *diferenciální* rovnice ve třech fázích:

1. Nalezneme nejprve obecné řešení příslušné homogenní rovnice. Toto řešení bude lineární kombinací „základních“ posloupností, které homogenní rovnici řeší. Systému těchto základních posloupností budeme říkat *fundamentální systém řešení homogenní rovnice*. Pro nalezení fundamentálního systému bude zapotřebí nalézt všechny kořeny charakteristické rovnice.

2. Nalezneme nějaké (zcela libovolné) řešení nehomogenní rovnice. Tomuto řešení budeme říkat *partikulární řešení*. Nalézt partikulární řešení může být poměrně těžké. Pokud je však pravá strana rekurentní rovnice (tj. posloupnost $F(n)$) dostatečně jednoduchá, lze tvar partikulárního řešení odhadnout. Přesný tvar partikulárního řešení lze potom snadno dopočítat.

3. Celkové (obecné) řešení nehomogenní rovnice je součtem partikulárního řešení a obecného řešení homogenní rovnice.

1.3.5 Hledání fundamentálního systému Předpokládejme, že jsme vyřešili charakteristickou rovnici

$$\lambda^k + c_1 \cdot \lambda^{k-1} + c_2 \cdot \lambda^{k-2} + \ldots + c_k = 0$$

a obdrželi jsme tak všechny její kořeny (psané i s násobnostmi):

kořen	λ_1	λ_2	$\ldots$	λ_r
násobnost	k_1	k_2	$\ldots$	k_r

.

Každý kořen rovnice přispěje do fundamentálního systému přesně tolika posloupnostmi, kolik činí jeho násobnost. Protože součet násobností kořenů je roven stupni charakteristické rovnice (platí $k_1+k_2+\ldots+k_r = k$), bude fundamentální systém obsahovat přesně tolik posloupností, kolik je řád rovnice.

Kořen λ_1 má násobnost k_1 a příslušné posloupnosti jsou

$$\begin{aligned}
&\lambda_1^n, \text{ pro } n \geq n_0\\
&n \cdot \lambda_1^n, \text{ pro } n \geq n_0\\
&n^2 \cdot \lambda_1^n, \text{ pro } n \geq n_0\\
&\vdots\\
&n^{k_1-1} \cdot \lambda_1^n, \text{ pro } n \geq n_0
\end{aligned}$$

Analogicky přispívají ostatní kořeny $\lambda_2, \ldots, \lambda_r$.

Uvedeme nyní několik příkladů hledání fundamentálních systémů a tím i hledání obecného řešení homogenních rovnic. V příkladech také ukážeme, co dělat, jestliže některé kořeny charakteristické rovnice jsou komplexní.

1.3.6 Příklady

1. Rovnice $S(n+2) - S(n+1) - S(n) = 0$, $n \geq 1$, má charakteristickou rovnici $\lambda^2 - \lambda - 1 = 0$. Poslední rovnice má dva jednoduché kořeny:

$$\begin{aligned}
\lambda_1 &= \frac{1+\sqrt{5}}{2}\\
\lambda_2 &= \frac{1-\sqrt{5}}{2}
\end{aligned}$$

Každý z kořenů dodá jednu posloupnost do fundamentálního sytému:

$$\begin{aligned}
\lambda_1 &\mapsto \left(\frac{1+\sqrt{5}}{2}\right)^n, \ n \geq 1\\
\lambda_2 &\mapsto \left(\frac{1-\sqrt{5}}{2}\right)^n, \ n \geq 1
\end{aligned}$$

Obecné řešení $S(n+2) - S(n+1) - S(n) = 0$, $n \geq 1$, je lineární kombinací fundamentálního systému. Jde tedy o posloupnost

$$a \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + b \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n, \ n \geq 1$$

kde a a b jsou reálné nebo komplexní konstanty (záleží na tom, zda rekurentní rovnici řešíme v reálném nebo komplexním oboru).

2. Rovnice $X(n+2) - 6X(n+1) + 9X(n) = 0$, $n \geq 10$, má charakteristickou rovnici $\lambda^2 - 6\lambda + 9 = 0$. Charakteristická rovnice má jediný kořen násobnosti dvě:

$$\lambda_1 = 3$$

Tento jediný kořen dodá dvě posloupnosti do fundamentálního sytému:

$$\begin{aligned}
\lambda_1 &\mapsto 3^n, \ n \geq 10\\
&\mapsto n \cdot 3^n, \ n \geq 10
\end{aligned}$$

Obecné řešení $X(n+2) - 6X(n+1) + 9X(n) = 0$, $n \geq 10$, je lineární kombinací posloupností z fundamentálního systému. Jde tedy o posloupnost

$$a \cdot 3^n + b \cdot n \cdot 3^n, \ n \geq 10$$

kde a a b jsou reálné nebo komplexní konstanty.

3. Rovnice $X(n+3) - 7X(n+2) + 15X(n+1) - 9X(n) = 0$, $n \geq 143$, má charakteristickou rovnici

$$\lambda^3 - 7\lambda^2 + 15\lambda - 9 = 0$$

Tato rovnice má jednoduchý kořen $\lambda_1 = 1$ a dvojnásobný kořen $\lambda_2 = 3$. Kořen λ_1 dodá jednu posloupnost do fundamentálního systému, kořen λ_2 dodá do fundamentálního systému posloupnosti dvě:

$$\begin{aligned} \lambda_1 &\mapsto 1^n, \ n \geq 143 \\ \lambda_2 &\mapsto 3^n, \ n \geq 143 \\ &\mapsto n \cdot 3^n, \ n \geq 143 \end{aligned}$$

Obecné řešení rovnice $X(n+3) - 7X(n+2) + 15X(n+1) - 9X(n) = 0$, $n \geq 143$, je lineární kombinací posloupností z fundamentálního systému. Jde tedy o posloupnost

$$a \cdot 1^n + b \cdot 3^n + c \cdot n \cdot 3^n, \ n \geq 143$$

kde a, b a c jsou reálné nebo komplexní konstanty.

4. Rovnice $X(n+2) + X(n) = 0$, $n \geq 8$, má charakteristickou rovnici $\lambda^2 + 1 = 0$. Charakteristická rovnice má dva komplexní kořeny násobnosti jedna:

$$\begin{aligned} \lambda_1 &= i \\ \lambda_2 &= -i \end{aligned}$$

kde i je imaginární jednotka. Každý z kořenů dodá jednu posloupnost do fundamentálního sytému:

$$\begin{aligned} \lambda_1 &\mapsto i^n, \ n \geq 8 \\ \lambda_2 &\mapsto (-i)^n, \ n \geq 8 \end{aligned}$$

Obecné řešení $X(n+2) + X(n) = 0$, $n \geq 8$, je lineární kombinací fundamentálního systému. Jde tedy o posloupnost

$$a \cdot (i)^n + b \cdot (-i)^n, \ n \geq 8.$$

Pozor! Koeficienty a a b v tomto případě musí být komplexní! Fundamentální systém je totiž tvořen komplexními posloupnostmi. To může být v aplikacích nežádoucí: rekurentní rovnice $X(n+2)+X(n) = 0$, $n \geq 8$, má reálné koeficienty a my můžeme vyžadovat její reálné řešení. Předvedeme nyní, jak reálný fundamentální systém nalézt. Postup je založen na pozorování, které známe z obecné algebry:

> Pokud má algebraická rovnice reálné koeficienty a je-li komplexní číslo α jejím kořenem násobnosti l, potom i komplexně sdružené číslo $\bar{\alpha}$ je jejím kořenem násobnosti l.

Komplexní kořeny algebraických rovnic jsou tedy „sdruženy do dvojic“. Z každé takové dvojice vybereme pouze jeden kořen a ten poslouží k výrobě reálného fundamentálního systému.

V našem příkladu vybereme kořen $\lambda_1 = i$ a jeho příslušnou fundamentální posloupnost

$$i^n = \left(\cos\frac{\pi}{2} + i \cdot \sin\frac{\pi}{2}\right)^n = \left(\cos\frac{n\pi}{2} + i \cdot \sin\frac{n\pi}{2}\right), \ n \geq 8$$

roztrhneme na reálnou a imaginární část (poslední rovnost je známá Moivreova rovnost). Dostaneme tak reálný fundamentální systém tvořený dvěma posloupnostmi:

$$\cos\frac{n\pi}{2}, \ n \geq 8$$
$$\sin\frac{n\pi}{2}, \ n \geq 8$$

Pokud rovnici $X(n+2)+X(n)=0$, $n\geq 8$, řešíme v reálném oboru, je její obecné řešení posloupnost

$$a\cdot\cos\frac{n\pi}{2}+b\cdot\sin\frac{n\pi}{2},\ n\geq 8$$

kde a a b jsou reálné konstanty.

5. Ukažme ještě jeden příklad, kdy kořeny charakteristické rovnice jsou komplexní. Předpokládejme, že vyřešením charakteristické rovnice jsme dostali následující kořeny:

$$\begin{aligned}\lambda_1 &= 2,\ \text{násobnosti jedna}\\ \lambda_2 &= 1+i,\ \text{násobnosti dvě}\\ \lambda_3 &= 1-i,\ \text{násobnosti dvě}\end{aligned}$$

Komplexní fundamentální systém bude obsahovat pět posloupností:

$$\begin{aligned}\lambda_1 &\mapsto 2^n\\ \lambda_2 &\mapsto (1+i)^n\\ &\mapsto n\cdot(1+i)^n\\ \lambda_3 &\mapsto (1-i)^n\\ &\mapsto n\cdot(1-i)^n\end{aligned}$$

Pro konstrukci reálného fundamentálního systému využijeme kořen λ_1 a z dvojice komplexně sdružených kořenů λ_2, λ_3 například kořen λ_2. Každou z komplexních posloupností, kterou kořen λ_2 vytváří, roztrhneme na reálnou a imaginární část. Kořen λ_2 tak přispěje do reálného fundamentálního systému celkem čtyřmi posloupnostmi:

$$\begin{aligned}\lambda_1 &\mapsto 2^n\\ \lambda_2 &\mapsto (\sqrt{2})^n\cdot\cos\frac{\pi n}{4}\\ &\mapsto (\sqrt{2})^n\cdot\sin\frac{\pi n}{4}\\ &\mapsto (\sqrt{2})^n\cdot n\cdot\cos\frac{\pi n}{4}\\ &\mapsto (\sqrt{2})^n\cdot n\cdot\sin\frac{\pi n}{4}\end{aligned}$$

První dvě posloupnosti pro kořen λ_2 jsou vytvořeny z posloupnosti $(1+i)^n$, druhé dvě z posloupnosti $n\cdot(1+i)^n$.

1.3.7 Hledání partikulárního řešení Partikulární řešení nehomogenní rovnice

$$X(n+k)+c_1\cdot X(n+k-1)+c_2\cdot X(n+k-2)+\ldots+c_k\cdot X(n)=F(n),\ n\geq n_0$$

se naučíme hledat pouze pro „dostatečně jednoduchou“ pravou stranu. Budeme předpokládat, že posloupnost $F(n)$, $n\geq n_0$ je ve tvaru

$$A^n\cdot P(n),\ n\geq n_0$$

kde A je nějaká konstanta (může být i komplexní) a $P(n)$ je nějaký polynom. Lze ukázat, že partikulární řešení pak má tvar

$$n^d\cdot A^n\cdot p(n),\ n\geq n_0$$

kde d je násobnost konstanty A jako kořene charakteristické rovnice (v případě, že A není kořenem charakteristické rovnice, klademe $d=0$) a $p(n)$ je polynom stejného stupně jako polynom $P(n)$. Koeficienty polynomu $p(n)$ je třeba dopočítat. K tomu využijeme požadavek, že posloupnost $n^d\cdot A^n\cdot p(n),\ n\geq n_0$ má být řešením rovnice

$$X(n+k)+c_1\cdot X(n+k-1)+c_2\cdot X(n+k-2)+\ldots+c_k\cdot X(n)=A^n\cdot P(n),\ n\geq n_0.$$

1.3.8 Příklady

1. Partikulární řešení rovnice $X(n+1)-2X(n)=3^n\cdot(2n+6)$, $n\geq 25$, odhadneme ve tvaru $n^0\cdot 3^n\cdot(an+b)$, $n\geq 25$. Číslo 3 není kořenem charakteristické rovnice $\lambda-2=0$.

   Potřebujeme nyní ještě určit koeficienty a a b. Protože posloupnost $n^0\cdot 3^n\cdot(an+b)$, $n\geq 25$, má být řešením rovnice
   $$X(n+1)-2X(n)=3^n\cdot(2n+6),\ n\geq 25$$
   musí po dosazení posloupnosti $n^0\cdot 3^n\cdot(an+b)$, $n\geq 25$, za $X(n)$ platit rovnost
   $$(n+1)^0\cdot 3^{n+1}\cdot(a(n+1)+b)-2\cdot n^0\cdot 3^n\cdot(an+b)=3^n\cdot(2n+6),\ n\geq 25$$
   Obě strany rovnosti můžeme vydělit výrazem 3^n. Dostáváme tak ekvivalentní rovnost (nulté mocniny již nevypisujeme)
   $$3\cdot(a(n+1)+b)-2\cdot(an+b)=2n+6,\ n\geq 25$$
   Porovnáním koeficientů u stejných mocnin napravo i nalevo dostáváme soustavu rovnic:
   $$\begin{array}{lrcrcl} \text{koeficienty u } n^1\text{:} & a & & & = & 2\\ \text{koeficienty u } n^0\text{:} & 3a & - & b & = & 6 \end{array}$$
   Řešením této soustavy je $a=2$, $b=0$.

   Partikulární řešení rovnice $X(n+1)-2X(n)=3^n\cdot(2n+6)$, $n\geq 25$, je tedy posloupnost
   $$3^n\cdot(2n),\ n\geq 25.$$

2. Partikulární řešení rovnice $X(n+2)-6X(n+1)+9X(n)=3^n\cdot(n-1)$, $n\geq 5$, odhadneme ve tvaru $n^2\cdot 3^n\cdot(an+b)$, $n\geq 5$, protože číslo 3 je dvojnásobným kořenem charakteristické rovnice.

   Při určení koeficientů a a b budeme již postupovat rychleji. Po dosazení odhadu řešení do rovnice dostáváme rovnost
   $$(n+2)^2\cdot 3^{n+2}\cdot(a(n+2)+b)-6\cdot(n+1)^2\cdot 3^{n+1}\cdot(a(n+1)+b)+9\cdot n^2\cdot 3^n\cdot(an+b)=3^n\cdot(n-1),\ n\geq 5.$$
   Vydělíme obě strany výrazem 3^n a porovnáme koeficienty u jednotlivých mocnin nalevo a napravo:
   $$\begin{array}{lrcrcl} \text{koeficienty u } n^3\text{:} & & & 0 & = & 0\\ \text{koeficienty u } n^2\text{:} & & & 0 & = & 0\\ \text{koeficienty u } n^1\text{:} & 54a & & & = & 1\\ \text{koeficienty u } n^0\text{:} & 54a & + & 18b & = & -1 \end{array}$$
   Řešením této soustavy je $a=\dfrac{1}{54}$, $b=-\dfrac{1}{9}$. Partikulární řešení rekurentní rovnice
   $$X(n+2)-6X(n+1)+9X(n)=3^n\cdot(n-1),\ n\geq 5,$$
   je tedy posloupnost $n^2\cdot 3^n\cdot(\dfrac{1}{54}n-\dfrac{1}{9})$, $n\geq 5$.

Víme, že metodu odhadu lze použít i v případě, že v pravé straně rekurentní rovnice $A^n\cdot P(n)$ je konstanta A komplexní. Následující příklad ukazuje, jak toho lze využít k nalezení partikulárního řešení některých rovnic.

1.3.9 Příklad Hledáme partikulární řešení rekurentní rovnice
$$X(n+1)-X(n)=(2n+1)\cdot\cos(9n),\ n\geq 13.$$
Pravá strana této rovnice není ve tvaru $A^n\cdot P(n)$. Povšimněme si však, že výraz $(2n+1)\cdot\cos(9n)$ je reálnou částí komplexního výrazu $(2n+1)\cdot e^{i9n}=(2n+1)\cdot(e^{9i})^n$. Poslední výraz je tvaru $A^n\cdot P(n)$ a partikulární řešení rekurentní rovnice
$$X(n+1)-X(n)=(2n+1)\cdot(e^{9i})^n,\ n\geq 13.$$

umíme metodou odhadu najít. Odhad tohoto řešení je ve tvaru $(an+b)\cdot(e^{9i})^n$, protože komplexní číslo e^{9i} není kořenem charakteristické rovnice. Hledané koeficienty a a b jsou nyní ovšem komplexní čísla. Jak uvidíme, obě komplexní čísla budeme potřebovat zapsat v algebraickém tvaru. Předpokládejme, že jsme čísla a a b již našli a že $a = a_1 + ia_2$ a $b = b_1 + ib_2$, kde všechna čísla a_1, a_2, b_1, b_2 jsou reálná.

Pro partikulární řešení rovnice

$$X(n+1) - X(n) = (2n+1)\cdot\cos(9n),\ n \geq 13,$$

je třeba vzít reálnou část z posloupnosti

$$((a_1 + ia_2)n + (b_1 + ib_2))\cdot(e^{9i})^n,\ n \geq 13.$$

Snadno zjistíme, že tato reálná část je posloupnost

$$(a_1 n + b_1)\cdot\cos(9n) - (a_2 n + b_2)\cdot\sin(9n),\ n \geq 13.$$

Zbývá nalézt hodnotu koeficientů a a b. To provedeme již známým způsobem. Dosazením odhadu do komplexní rekurentní rovnice dostaneme rovnost

$$(a(n+1)+b)\cdot(e^{9i})^{n+1} - (an+b)\cdot(e^{9i})^n = (2n+1)\cdot(e^{9i})^n,\ n \geq 13.$$

Vydělením výrazem $(e^{9i})^n$ a porovnáním koeficientů dostaneme soustavu rovnic

$$\begin{array}{lrcl} \text{koeficienty u } n^1\text{:} & a(e^{9i}-1) & = & 2 \\ \text{koeficienty u } n^0\text{:} & ae^{9i} + b(e^{9i}-1) & = & 1 \end{array}$$

Řešením této soustavy jsou komplexní čísla $a = \dfrac{2}{e^{9i}-1}$ a $b = -\dfrac{e^{9i}+1}{(e^{9i}-1)^2}$. Obě komplexní čísla potřebujeme v algebraickém tvaru. Číslo a tedy hledáme ve tvaru $a_1 + ia_2$ a číslo b hledáme ve tvaru $b_1 + ib_2$, kde všechna čísla a_1, a_2, b_1, b_2 jsou reálná.

$$\begin{aligned} a &= \frac{2}{e^{9i}-1} = \frac{2}{\cos 9 + i\sin 9 - 1} = \frac{2((\cos 9 - 1) - i\sin 9}{(\cos 9 - 1)^2 + \sin^2 9} = \frac{2((\cos 9 - 1) - i\sin 9}{2 - 2\cos 9} = \\ &= -1 + i\frac{\sin 9}{2(\cos 9 - 1)} \end{aligned}$$

Protože platí $b = -\dfrac{e^{9i}+1}{(e^{9i}-1)^2} = \dfrac{-(e^{9i}+1)}{2(e^{9i}-1)}\cdot a$, vyjádříme nejprve v algebraickém tvaru číslo $\dfrac{-(e^{9i}+1)}{2(e^{9i}-1)}$:

$$\begin{aligned} \frac{-(e^{9i}+1)}{2(e^{9i}-1)} &= -\frac{(\cos 9 + 1) + i\sin 9}{2((\cos 9 - 1) + i\sin 9)} = -\frac{((\cos 9 + 1) + i\sin 9)\cdot((2\cos 9 - 2) - i2\sin 9)}{(2\cos 9 - 2)^2 + 4\sin^2 9} = \\ &= i\cdot\frac{\sin 9}{2 - 2\cos 9} \end{aligned}$$

Potom

$$\begin{aligned} b &= i\cdot\frac{\sin 9}{2-2\cos 9}\cdot a = i\cdot\frac{\sin 9}{2-2\cos 9}\cdot\left(-1 + i\frac{\sin 9}{2(\cos 9 - 1)}\right) = \\ &= \frac{\cos 9 + 1}{2(\cos 9 - 1)} - i\frac{\sin 9}{2(1-\cos 9)} \end{aligned}$$

Z výše uvedených rovnic plyne hledaný výsledek:

$$\begin{aligned} a_1 &= -1 \\ a_2 &= \frac{\sin 9}{2(\cos 9 - 1)} \\ b_1 &= \frac{\cos 9 + 1}{2(\cos 9 - 1)} \\ b_2 &= \frac{\sin 9}{2(1-\cos 9)} \end{aligned}$$

Metodu odhadu partikulárního řešení lze kombinovat s metodou *superposice* partikulárních řešení. Nebudeme obecně metodu superposice formulovat — ukážeme ji na příkladu.

1.3.10 Příklad Máme vyřešit rekurentní rovnici $X(n+2) - 6X(n+1) + 9X(n) = 1 + 2^n$, $n \geq 20$. Pravá strana není vhodná pro odhad, povšimněme si ale, že je součtem 1 a 2^n. Pro tyto jednotlivé posloupnosti již partikulární řešení najít umíme. Budeme tedy postupovat následovně:

1. Nalezneme partikulární řešení rovnice $X(n+2) - 6X(n+1) + 9X(n) = 1$, $n \geq 20$. Označme takové partikulární řešení $A(n)$.
2. Nalezneme partikulární řešení rovnice $X(n+2) - 6X(n+1) + 9X(n) = 2^n$, $n \geq 20$. Označme takové partikulární řešení $B(n)$.
3. Utvořme součet $A(n) + B(n)$. Této posloupnosti říkáme *superposice řešení* $A(n)$ *a* $B(n)$.

   Princip superposice pak je tvrzení, že posloupnost $A(n)+B(n)$, $n \geq 20$, je opravdu partikulárním řešením rovnice $X(n+2) - 6X(n+1) + 9X(n) = 1 + 2^n$, $n \geq 20$.

Kroky 1. a 2. provedeme známým způsobem:

1. Partikulární řešení odhadneme ve tvaru $A(n) = a$, $n \geq 20$, protože číslo 1 není kořenem charakteristické rovnice. Dosazením a porovnáním koeficientů dostaneme $a = \frac{1}{4}$.

   Řešení $A(n)$ je tedy posloupnost $\frac{1}{4}$, $n \geq 20$.
2. Partikulární řešení odhadneme ve tvaru $B(n) = 2^n \cdot b$, $n \geq 20$, protože číslo 2 není kořenem charakteristické rovnice. Dosazením a porovnáním koeficientů dostaneme $b = 1$.

   Řešení $B(n)$ je tedy posloupnost 2^n, $n \geq 20$.

Podle principu superposice je partikulárním řešením rekurentní rovnice

$$X(n+2) - 6X(n+1) + 9X(n) = 1 + 2^n, \ n \geq 20$$

součet posloupností $A(n)$ a $B(n)$, neboli posloupnost $\frac{1}{4} + 2^n$, $n \geq 20$.

V příkladu 1.3.1 jsme měli najít řešení rekurentní rovnice $S(n+2) - S(n+1) - S(n) = 0$, $n \geq 1$, které navíc splňuje podmínky $S(1) = 2$ a $S(2) = 3$. Takovým podmínkám říkáme *počáteční podmínky rekurentní rovnice*. Hledání řešení splňující počáteční podmínky ukážeme v dalším příkladu.

1.3.11 Příklad Nalezněte řešení rekurentní rovnice

$$X(n+2) - 6X(n+1) + 9X(n) = 1 + 2^n, \ n \geq 1$$

splňující počáteční podmínky $X(1) = 0$, $X(2) = 1$.

Řešení rozdělíme do dvou kroků:

1. Nalezneme obecné řešení rovnice $X(n+2) - 6X(n+1) + 9X(n) = 1 + 2^n$, $n \geq 1$, bez počátečních podmínek.
2. Protože daná rekurentní rovnice je druhého řádu, vystupují v obecném řešení dvě konstanty. Hodnotu těchto konstant určíme ze zadaných počátečních podmínek.

Budeme postupovat velmi rychle, protože většinu dílčích úloh jsme již vyřešili.

1. Z příkladu 1.3.6(2) víme, že posloupnost

   $$a \cdot 3^n + b \cdot n \cdot 3^n, \ n \geq 1$$

   kde a a b jsou konstanty, je obecným řešením homogenní rovnice.

   Z příkladu 1.3.10 víme, že posloupnost

   $$\frac{1}{4} + 2^n, \ n \geq 1$$

je partikulárním řešením.

Obecné řešení rovnice $X(n+2)-6X(n+1)+9X(n)=1+2^n$, $n\geq 1$, bez počátečních podmínek je součet partikulárního řešení nehomogenní rovnice a obecného řešení homogenní rovnice. Jde o posloupnost

$$\frac{1}{4}+2^n+a\cdot 3^n+b\cdot n\cdot 3^n,\ n\geq 1.$$

2. Získané obecné řešení nyní dosadíme do počátečních podmínek. Dostaneme soustavu rovnic pro a a b:

$$\begin{aligned} 0 &= X(1) &&= \frac{1}{4}+2^1+a\cdot 3^1+b\cdot 1\cdot 3^1 \\ 1 &= X(2) &&= \frac{1}{4}+2^2+a\cdot 3^2+b\cdot 2\cdot 3^2 \end{aligned}$$

Řešením této soustavy je $a=-\frac{41}{36}$, $b=\frac{7}{18}$.

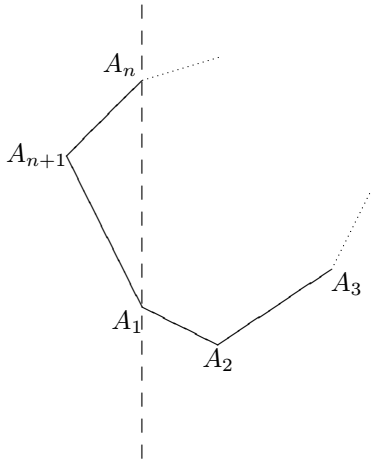
Hledané řešení splňující počáteční podmínky je posloupnost

$$\frac{1}{4}+2^n-\frac{41}{36}\cdot 3^n+\frac{7}{18}\cdot n\cdot 3^n,\ n\geq 1.$$

1.3.12 Příklad V tomto příkladu nejprve sestavíme *rekursivní* algoritmus pro výpočet počtu úhlopříček konvexního mnohoúhelníka a poté sestavíme *rekurentní* rovnici pro jejich počet.

Jako $U(n)$ označíme počet úhlopříček konvexního n-úhelníka pro $n\geq 4$. Zřejmě platí $U(4)=2$. Náš algoritmus tedy bude počítat počet úhlopříček konvexních čtyřúhelníků *nerekursivně*.

Je-li zadán konvexní $(n+1)$-úhelník $A_1A_2\ldots A_{n+1}$, kde $n\geq 4$, provede algoritmus nejprve dekomposici podle následujícího obrázku:



To jest, algoritmus „uřízne“ vrchol A_{n+1} a vzniklý n-úhelník $A_1A_2\ldots A_n$ je opět konvexní. Algoritmus se zavolá rekursivně na tento n-úhelník a k výsledku $U(n)$ se přičte číslo $(n-2)+1$: vynechali jsme totiž $n-2$ úhlopříček původního $(n+1)$-úhelníka $A_1A_2\ldots A_{n+1}$, které vedou z vrcholu A_{n+1} do vrcholů A_2, …, A_{n-1}, a ještě jsme vynechali 1 úhlopříčku A_nA_1.

Chod rekursivního algoritmu tedy dává následující rekurentní rovnici pro výpočet $U(n)$:

$$\begin{aligned} U(4) &= 2 \\ U(n+1) &= U(n)+n-1, \quad \text{pro } n\geq 4 \end{aligned}$$

1.4 Induktivní definice

V tomto odstavci zmíníme *princip strukturální indukce*, který je v computer science velmi důležitý. Formulovat jej budeme pro množiny slov nad konečnou abecedou, v odstavci 1.5 ukážeme, jak lze princip indukce zobecnit a použít na definici třídy *parciálně rekursivních funkcí*, které jsou jedním ze standardních modelů algoritmu.

Syntaxi programovacího jazyka často zapisujeme v *Backusově-Naurově formě*.[6] Tento zápis je příkladem *induktivní definice*. Ukažme to na následujícím příkladu.

$$\langle string \rangle \quad ::= \quad \mathtt{0} \mid \mathtt{1} \mid \langle string \rangle \mathtt{0} \mid \langle string \rangle \mathtt{1} \tag{1.8}$$

Takový způsob zápisu je zkratkou za následující tvrzení:

1. Definujeme objekty, kterým říkáme *string*. (Také se používá termín objekty typu *string*.)
2. `0` i `1` jsou objekty typu *string*. Takové objekty jsou nejjednoduššími objekty typu *string*.
3. Pokud jsme již zkonstruovali objekt typu *string*, lze za něj připsat `0` nebo `1` a dostáváme opět objekt typu *string*.
4. Objekt typu *string* vzniká pouze konečným použitím výše uvedených pravidel a jinak objekt typu *string* vzniknout nemohl.

Zápis (1.8) je běžný v literatuře o programovacích jazycích. V tomto textu budeme používat *matematickou variantu Backusovy-Naurovy formy*. Přepíšeme tímto způsobem (1.8):

$$\mathtt{S} \quad ::= \quad \mathtt{0} \mid \mathtt{1} \mid \mathtt{S0} \mid \mathtt{S1} \tag{1.9}$$

1. Volíme `S` jako *generické jméno* objektů, které konstruujeme, tedy `S` je typický objekt typu *string*.
2. `0` i `1` jsou konstruovanými objekty. (Používá se též terminologie: `0` i `1` jsou *správně vytvořenými objekty* typu *string*.) Takové objekty jsou nejjednoduššími konstruovanými objekty.
3. Pokud jsme již zkonstruovali objekt `S`, jsou objekty `S0` a `S1` *správně vytvořenými objekty* typu *string*.
4. Správně vytvořený objekt typu *string* vzniká pouze konečným použitím výše uvedených pravidel a jinak správně vytvořený objekt typu *string* vzniknout nemohl.

Jiná notace pro generování správně vytvořených objektů je zápis pomocí *odvozovacích pravidel*. Uvedeme popis odvozovacích pravidel pro objekty typu *string*:

$$\frac{}{\mathtt{0}} \quad \Big| \quad \frac{}{\mathtt{1}} \quad \Big| \quad \frac{\mathtt{S}}{\mathtt{S0}} \quad \Big| \quad \frac{\mathtt{S}}{\mathtt{S1}} \tag{1.10}$$

Výše uvedený zápis je třeba číst takto:

1. $\frac{}{\mathtt{0}}$ i $\frac{}{\mathtt{1}}$ jsou odvozovací pravidla. Taková pravidla, která nemají obsazen vrchol nad čarou chápeme jako *axiomy*: nepotřebujeme nic na to, abychom odvodili, že `0` i `1` jsou správně vytvořené objekty typu *string*.
2. Odvozovací pravidla $\frac{\mathtt{S}}{\mathtt{S0}}$ a $\frac{\mathtt{S}}{\mathtt{S1}}$ chápeme jako pravidla *deduktivní*. Například pravidlo $\frac{\mathtt{S}}{\mathtt{S0}}$ čteme takto: je-li řetězec `S` správně odvozen, je správně odvozen i řetězec `S0`.

Jednotlivá odvozovací pravidla bývá výhodné nějak pojmenovat. Jméno odvozovacího pravidla píšeme do závorek po pravé straně vodorovné oddělující čáry:

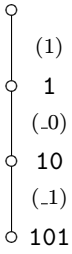
$$\frac{}{\mathtt{0}}\ (0) \quad \Big| \quad \frac{}{\mathtt{1}}\ (1) \quad \Big| \quad \frac{\mathtt{S}}{\mathtt{S0}}\ (\_0) \quad \Big| \quad \frac{\mathtt{S}}{\mathtt{S1}}\ (\_1) \tag{1.11}$$

[6] Tuto notaci vymyslel John Warner Backus (nar. 1924) v roce 1959 pro popis syntaxe jazyka ALGOL 60. Původně se jí říkalo Backusova normální forma, na Backusovu-Naurovu formu byla přejmenována později (Peter Naur se podílel na zprávě o ALGOLU v roce 1963). Je zajímavé, že pravidla stejné vyjadřovací schopnosti (metapravidla, transformace a rekursi) použil hindský gramatik Pāṇini (cca 6. století př.n.l.) pro popis *sanskrtu*. Jeho 3 959 veršů díla *Astādhyāyī* tak představuje první formální popis přirozeného jazyka.

Zápis syntaxe odvozovacími pravidly má tu výhodu, že umožňuje sledovat tvorbu správně vytvořených objektů pomocí jednotlivých pravidel. Například řetězec `101` je jistě správně vytvořený objekt typu *string*, protože použitím axiomu (1) byl vytvořen řetězec `1`, aplikací pravidla (_0) na řetězec `1` vzniká řetězec `10`, a konečně aplikací pravidla (_1) na řetězec `10` vzniká řetězec `101`. Takový slovní popis může být zkráceně zapsán následovně:

$$\dfrac{\dfrac{\dfrac{}{\texttt{1}}\,(1)}{\texttt{10}}\,(\_0)}{\texttt{101}}\,(\_1) \tag{1.12}$$

Tento zápis chápeme jako kořenový strom. Jeho kořenem je řetězec `101`, jeho jediný list je neobsazen a vrcholy $v_1,\dots v_k$ na vyšší hladině jsou spojeny hranou označenou (p) s vrcholem v na nižší hladině, pokud jde o instanci odvozovacího pravidla (p) s předpoklady $v_1, \dots, v_k$ a závěrem v. Stromům, které mají tyto vlastnosti, říkáme *syntaktické stromy*.[7] V teorii grafů by strom (1.12) byl znázorněn takto:



My budeme používat notaci syntaktického stromu, která je obvyklá v computer science. Přesnou definici syntaktického stromu ponecháváme jako cvičení (viz 1.6.23).

Ukážeme ještě jeden příklad syntaktického stromu.

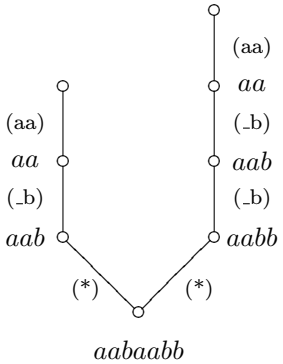
1.4.1 Příklad Ať $\Sigma = \{a, b\}$ je konečná abeceda. Uvažujme o následující sadě odvozovacích pravidel:

$$\dfrac{}{aa}\,(\text{aa}) \quad | \quad \dfrac{w}{wb}\,(\_\text{b}) \quad | \quad \dfrac{w_1 \quad w_2}{w_1 w_2}\,(*) \tag{1.13}$$

Sestrojíme nyní syntaktický strom řetězce $aabaabb$, a tím ukážeme, že je správně vytvořen.

$$\dfrac{\dfrac{\dfrac{}{aa}\,(\text{aa})}{aab}\,(\_\text{b}) \qquad \dfrac{\dfrac{}{aa}\,(\text{aa})}{aab}\,(\_\text{b})}{aabaabb}\,(*) \tag{1.14}$$

Strom (1.14) by v teorii grafů byl znázorněn takto:



1.4.2 Poznámka Povšimněme si, že každý netriviální podstrom syntaktického stromu je opět syntaktickým stromem.

Notace odvozovacích pravidel slouží k takzvanému *induktivnímu zadávání množin*. Ukažme to na příkladu.

[7] V literatuře o programování se používá termín *parsing tree*.

1.4.3 Příklad Ať $\Sigma = \{a, b\}$ je konečná abeceda. Ať G je sada odvozovacích pravidel:

$$\frac{}{aa}\ (\text{aa}) \quad | \quad \frac{w}{wb}\ (\_\text{b}) \tag{1.15}$$

Tato dvě pravidla jednoznačně definují množinu L, která je podmnožinou množiny všech slov Σ^* a která je nejmenší množinou s následujícími vlastnostmi:

1. Slovo aa patří do L.
2. Jestliže slovo w patří do L, pak i slovo wb patří do L.

Značení G a L jsme nezvolili náhodou: uvědomme si, že na sadu pravidel G se lze dívat jako na *gramatiku* a na množinu L jako na *jazyk* (anglicky: *language*), který je danou gramatikou popsán.

Sada odvozovacích pravidel G tedy určuje množinu L. Všimněme si ale, že obecně není zadána žádná charakteristická vlastnost prvků množiny L. Prvky množiny L jsou zadány následovně: nejprve se řekne, které řetězce v množině L zaručeně leží a poté se sdělují informace typu „pokud v L leží toto, pak tam leží i toto“. Takový způsob velmi připomíná princip matematické indukce, proto hovoříme o induktivním zadání množiny L.

1.4.4 Příklad Vraťme se ještě k zadání gramatiky G z příkladu 1.4.3. Máme vyřešit následující dva problémy:

1. Ukažte, že slovo $aabb$ patří do množiny L.
2. Ukažte, že slovo $baabb$ nepatří do množiny L.

První problém snadno vyřešíme tím, že sestavíme syntaktický strom řetězce $aabb$:

$$\cfrac{\cfrac{\cfrac{}{aa}\ (\text{aa})}{aab}\ (\_\text{b})}{aabb}\ (\_\text{b})$$

Protože fakt $aabb \in L$ je ekvivalentní existenci syntaktického stromu, je první otázka zodpovězena.

Vyřešit druhý problém by znamenalo ukázat, že syntaktický strom řetězce $baabb$ *neexistuje*. Lze však postupovat i takto: řetězec $baabb$ nemá vlastnost, kterou každý řetězec z množiny L musí mít, totiž začínat řetězcem aa.

Nalézt takovou vlastnost, společnou všem prvkům množiny L, bývá často velmi jednoduché. Je jí totiž jakákoli vlastnost V, která je „invariantní na průchod gramatikou G“ v následujícím smyslu:

1. Závěr každého axiomu z G má vlastnost V.
2. Pro každé deduktivní pravidlo gramatiky G platí následující:

   Jestliže všechny předpoklady mají vlastnost V, potom i závěr má vlastnost V.

V našem případě zvolíme za V vlastnost začínat řetězcem aa. Je snadné ověřit, že V je ve výše uvedeném smyslu invariantní na průchod gramatikou G.

Proč nyní můžeme usoudit, že každý řetězec $w \in L$ má vlastnost V? Je-li totiž $w \in L$, potom existuje syntaktický strom T_w řetězce w: listy stromu T_w jsou neobsazeny, na další hladině jsou axiomy G a na další hladiny stromu T_w se dostáváme aplikací odvozovacích pravidel G, dokud nedorazíme ke kořeni stromu T_w — tím je řetězec w.

Protože vlastnost V se nemění při průchodu stromem T_w směrem ke kořeni, má kořen stromu T_w vlastnost V.

Princip dokazování naznačený v předchozím příkladu je opět indukčním principem: takzvaným *principem strukturální indukce*. Tento princip nyní zformulujeme obecně pro množiny slov zadané induktivně:

1.4.5 Věta (Strukturální indukce)
Ať Σ je konečná abeceda a G je konečná sada odvozovacích pravidel, která induktivně zadává množinu slov L. Označme jako A množinu všech odvozovacích pravidel z G, která jsou axiomy (tj. pravidla z A nemají žádné předpoklady) a označme jako D množinu všech odvozovacích pravidel z G, která jsou deduktivními pravidly (tj. každé pravidlo z D má neprázdnou konečnou množinu předpokladů).

Ať V je nějaká vlastnost slov nad abecedou Σ a ať vlastnost V splňuje následující dvě podmínky:

(SI1) *Závěr každého axiomu z množiny A má vlastnost V.*

(SI2) *Pro každou instanci libovolného deduktivního pravidla v množině D platí:*

> *Jestliže všechny předpoklady pravidla mají vlastnost V, potom i závěr tohoto pravidla má vlastnost V.*

Potom každé slovo v množině L má vlastnost V.

Důkaz. Vezměme libovolné slovo w z množiny L. Chceme ukázat, že slovo w má vlastnost V. Budeme postupovat silným principem matematické indukce podle výšky $h(T_w)$ syntaktického stromu T_w slova w.

1. **Základní krok:** $h(T_w) = 1$. Potom w je závěrem nějakého axiomu z množiny A. Podle předpokladu (SI1) má slovo w vlastnost V.

2. **Indukční krok:** Mějme slovo w se syntaktickým stromem T_w takovým, že $h(T_w) = n + 1$. Indukční předpoklad formulujeme následovně:

   Každé slovo v se syntaktickým stromem T_v výšky $h(T_v) < n + 1$ má vlastnost V.

   Protože $h(T_w) = n + 1$, je syntaktický strom T_w tvaru $\dfrac{T_{w_1} \quad \dots \quad T_{w_k}}{w}$ (d) , kde (d) je některé odvozovací pravidlo z množiny D a $T_{w_1}, \dots, T_{w_k}$ jsou syntaktické stromy slov $w_1, \dots, w_k$ výšky menší než $n + 1$. Podle indukčního předpokladu mají tedy všechna slova $w_1, \dots, w_k$ vlastnost V.

   Podle předpokladu (SI2) má i slovo w vlastnost V.

Důkaz je nyní (podle silného principu matematické indukce) hotov. ■

V dalším se zaměříme na následující dvě otázky:

1. Jak induktivně zadanou množinu popsat neinduktivně? To jest — jak najít vlastnost, charakteristickou pro slova ze vzniklého jazyka?

2. Jak pro danou množinu slov najít sadu odvozovacích pravidel tak, aby danou množinu šlo chápat jako induktivně definovaný jazyk?

1.4.6 Příklad Zkusme zapsat množinu L z příkladu 1.4.3 neinduktivně. To znamená, že hledáme vlastnost V takovou, že $\{w \in \Sigma^* \mid w \text{ má vlastnost } V\} = L$.

Intuitivně je zřejmé, že hledaná vlastnost V je:

> existuje přirozené číslo k takové, že $w = aab^k$

(kde zápisem b^k rozumíme řetězec k po sobě jdoucích symbolů b).

Označme $S = \{w \in \Sigma^* \mid w \text{ má vlastnost } V\}$. Abychom ukázali, že $S = L$, stačí ukázat, že $S \subseteq L$ a $L \subseteq S$. Obě inkluse budeme dokazovat matematickou indukcí.

$S \subseteq L$. Chceme ukázat, že pokud $w \in S$, pak $w \in L$. Vezměme tedy slovo $w \in S$. Slovo w je tedy ve tvaru aab^k pro nějaké přirozené číslo k. *Indukcí podle k* ukážeme, že $w \in L$. K tomu stačí ukázat, že pro každé k existuje syntaktický strom vytvořený podle pravidel G.

1. **Základní krok:** $k = 0$. Potom $w = aa$ a strom $\dfrac{}{aa}$ (aa) je hledaný syntaktický strom.

2. Ať $w = aab^{k+1}$. **Indukční předpoklad:** předpokládejme, že pro každé slovo $v = aab^k$ existuje syntaktický strom T_v. Potom zřejmě $w = w'b$, kde $w' = aab^k$ a podle indukčního předpokladu slovo w' má syntaktický strom $T_{w'}$. Potom strom $\dfrac{T_{w'}}{w}$ (_b) je syntaktický strom slova w. Tedy $w \in L$.

Dokázali jsme, že z $w \in S$ plyne $w \in L$.

$L \subseteq S$. Chceme ukázat, že pokud $w \in L$, pak $w \in S$. Vezměme tedy slovo $w \in L$. Ukážeme, že existuje přirozené číslo k takové, že $w = aab^k$. Budeme postupovat *strukturální indukcí*. K tomu stačí dokázat, že vlastnost

existuje přirozené číslo k takové, že $w = aab^k$

je invariantní na průchod gramatikou G.

1. **Základní krok:** gramatika G obsahuje pouze jeden axiom: $\dfrac{}{aa}$ (aa) . Závěrem tohoto axiomu je slovo $w = aa$ a to je tvaru aab^k pro $k = 0$.
2. Gramatika G obsahuje pouze jedno deduktivní pravidlo: $\dfrac{w}{wb}$ (_b) . **Indukční předpoklad strukturální indukce:** předpokládejme, že pro slovo w existuje přirozené číslo k takové, že $w = aab^k$.

   Potom slovo wb je tvaru aab^kb, tudíž má hledanou vlastnost.

Dokázali jsme, že z $w \in L$ plyne $w \in S$.

Uveďme trochu složitější příklad.

1.4.7 Příklad Ať $\Sigma = \{a, b\}$ a $S = \{w \in \Sigma^* \mid w$ obsahuje stejný počet symbolů a a $b\}$. Zadejme množinu S induktivně.

Hledáme tedy sadu odvozovacích pravidel G takovou, že pro příslušný jazyk L platí $L = S$.

Ukážeme, že hledaná sada G je například:

$$\frac{}{\varepsilon}\ (0) \quad | \quad \frac{w}{awb}\ (a) \quad | \quad \frac{w}{bwa}\ (b) \quad | \quad \frac{w_1 \quad w_2}{w_1 w_2}\ (*)$$

kde ε značí *prázdné slovo*.

Než dokážeme, že $S = L$, označme jako $|w|_a$ a $|w|_b$ počet symbolů a a b ve slově w.

$S \subseteq L$. Chceme pro každé $w \in S$ zkonstruovat syntaktický strom vytvořený podle sady odvozovacích pravidel G. Protože $w \in S$, platí $|w|_a = |w|_b$. Budeme postupovat *indukcí podle* $|w|_a$.

1. **Základní krok:** $|w|_a = 0$. Pak ale nutně $w = \varepsilon$ a $\dfrac{}{\varepsilon}$ (0) je hledaný syntaktický strom.
2. Ať $|w|_a = n + 1$. **Indukční předpoklad:** předpokládejme, že pro každé slovo v, pro které platí $|v|_a = |v|_b = n$, dovedeme najít syntaktický strom T_v. Protože w obsahuje alespoň dva znaky ($|w|_a > 0$), může nastat právě jeden ze dvou případů:

   (a) w začíná i končí stejným znakem.

   (b) w začíná i končí různými znaky.

   Probereme každý případ zvlášť.

   (a) Vyřešíme situaci, kdy w začíná i končí znakem a. Situace, kdy w začíná i končí znakem b, se řeší analogicky. Předpokládáme tedy, že w je ve tvaru $aw'a$ pro nějaké slovo w'. Potom zřejmě existují neprázdná slova w_1, w_2 taková, že $w = w_1w_2$, $|w_1|_a = |w_1|_b$ a $|w_2|_a = |w_2|_b$ (viz cvičení 1.6.29). Podle indukčního předpokladu existují syntaktické stromy T_{w_1}, T_{w_2} pro slova w_1, w_2. Potom strom $\dfrac{T_{w_1} \quad T_{w_2}}{w}$ (*) je zřejmě syntaktickým stromem slova w.

   (b) Vyřešíme situaci, kdy w začíná znakem a a končí znakem b. Situace, kdy w začíná znakem b a končí znakem a, se řeší analogicky. Předpokládáme tedy, že w je ve tvaru $aw'b$ pro nějaké slovo w'. Slovo w' obsahuje zřejmě stejný počet symbolů a a b, navíc $|w'|_a = n$. Podle indukčního předpokladu dovedeme tedy najít syntaktický strom $T_{w'}$ slova w'. Potom strom $\dfrac{T_{w'}}{w}$ (a) je zřejmě syntaktickým stromem slova w.

$L \subseteq S$. Chceme pro každé $w \in L$ dokázat, že $|w|_a = |w|_b$. Budeme postupovat strukturální indukcí a ukážeme, že vlastnost V

pro slovo w platí rovnost $|w|_a = |w|_b$

je invariantní na průchod danou gramatikou.

Základní krok: jediný axiom dané gramatiky je $\dfrac{}{\varepsilon}$ (0) . Závěr tohoto axiomu vlastnost V má: $|\varepsilon|_a = |\varepsilon|_b = 0$.

Indukční krok: gramatika má tři deduktivní pravidla, rozdělíme tedy výpočet na tři části:

1. Pravidlo $\dfrac{w}{awb}$ (a) . Předpokládejme, že $|w|_a = |w|_b$. Potom $|awb|_a = |w|_a + 1$ a $|awb|_b = |w|_b + 1$, tudíž $|awb|_a = |awb|_b$ a vlastnost V je invariantní na průchod daným odvozovacím pravidlem.
2. Pravidlo $\dfrac{w}{bwa}$ (b) . Předpokládejme, že $|w|_a = |w|_b$. Potom $|bwa|_a = |w|_a + 1$ a $|bwa|_b = |w|_b + 1$, tudíž $|bwa|_a = |bwa|_b$ a vlastnost V je invariantní na průchod tímto odvozovacím pravidlem.
3. Pravidlo $\dfrac{w_1 \quad w_2}{w_1 w_2}$ (*) . Předpokládejme, že $|w_1|_a = |w_1|_b = n_1$ a $|w_2|_a = |w_2|_b = n_2$. Potom $|w_1 w_2|_a = n_1 + n_2$ a $|w_1 w_2|_b = n_1 + n_2$, tudíž $|w_1 w_2|_a = |w_1 w_2|_b$ a vlastnost V je invariantní na průchod tímto odvozovacím pravidlem.

Ukázali jsme, že $S = L$.

Je-li nějaká množina M induktivně zadaná, lze příslušných odvozovacích pravidel využít k definici funkcí s definičním oborem M. Ukažme to na příkladu.

1.4.8 Příklad Ve cvičení 1.6.28 se ukazuje, že sada pravidel

$$\frac{}{\varepsilon}\ (0) \quad | \quad \frac{w}{wa_1}\ (1) \quad | \quad \frac{w}{wa_2}\ (2) \quad | \quad \ldots \quad | \quad \frac{w}{wa_k}\ (k) \tag{1.16}$$

definuje induktivně množinu Σ^* všech slov nad abecedou $\Sigma = \{a_1, a_2, \ldots, a_k\}$ $(k \geq 1)$.

Pro každé slovo $w \in \Sigma$ definujeme slovo $rev(w)$ následovně:

$$\begin{aligned} rev(\varepsilon) &= \varepsilon \\ rev(x_1 x_2 \ldots x_{n-1} x_n) &= x_n x_{n-1} \ldots x_2 x_1 \end{aligned}$$

Ukážeme, že funkce $f : \Sigma^* \longrightarrow \Sigma^*$, která je definována *rekursivně* jako:

$$\begin{aligned} f(\varepsilon) &= \varepsilon \\ f(wa_1) &= a_1 f(w) \\ f(wa_2) &= a_2 f(w) \\ &\vdots \\ f(wa_k) &= a_k f(w) \end{aligned}$$

vyhovuje rovnici $f(w) = rev(w)$ pro všechna $w \in \Sigma^*$.

Označíme-li jako V vlastnost $f(w) = rev(w)$, chceme ukázat, že všechna $w \in \Sigma^*$ mají vlastnost V. Budeme postupovat strukturální indukcí. Stačí tedy ukázat, že vlastnost V splňuje podmínky (SI1) a (SI2) z věty 1.4.5.

(SI1) Jediným axiomem v sadě odvozovacích pravidel (1.16) je pravidlo $\dfrac{}{\varepsilon}$ (0) a závěr tohoto axiomu, prázdné slovo ε, má vlastnost V.

(SI2) Vezměme libovolné deduktivní pravidlo ze sady (1.16). Platnost podmínky (SI2) ověříme pro pravidlo (1), pro ostatní pravidla je důkaz zcela analogický.

Ať (jediný) předpoklad w pravidla (1) má vlastnost V. Závěrem tohoto pravidla je slovo wa_1. Slovo wa_1 má také vlastnost V, protože

$$f(wa_1) = a_1 f(w) = a_1 rev(w) = rev(wa_1).$$

1.4.9 Poznámka Teoreticky důležitou funkcí, definovanou rekursivně, je *Ackermannova funkce*:[8]

$$\begin{aligned} A(0, n) &= n + 1, \\ A(m + 1, 0) &= A(m, 1), \\ A(m + 1, n + 1) &= A(m, A(m + 1, n)). \end{aligned}$$

[8] Wilhelm Friedrich Ackermann (1896–1962), kromě po něm nazvané funkce, přispěl do pokladnice matematiky i nejrůznějšími pracemi z matematické logiky.

Jde o funkci dvojice přirozených čísel a na první pohled není zřejmé, že je definována korektně. Teoretický význam Ackermannovy funkce spočívá v tom, že jde o *parciálně rekursivní funkci*, která není *primitivně rekursivní*. V praxi to znamená, že pokud chceme výpočet Ackermannovy funkce naprogramovat, musíme k tomu použít cyklus `while`, který *nelze* zredukovat na cyklus `for`. To je totiž přesně rozdíl mezi parciálně a primitivně rekursivními funkcemi, viz skriptum

☞ K. Richta a J. Velebil, *Sémantika programovacích jazyků*, Karolinum, Praha 1997

a odstavec 1.5.

Vyzkoušejte si, jak vypadá výpočet hodnoty $A(2,2)$. Viz také cvičení 1.6.5 a 1.6.6.

Připomeňme, že dalším problémem, který jsme studovali, je otázka, zda *danou* množinu slov lze zadat induktivně. Tato otázka nemá obecně kladnou odpověď, jak ukazuje následující věta.

1.4.10 Věta *Ať $k \geq 1$ a ať $\Sigma = \{a_1, a_2, \ldots, a_k\}$ je konečná abeceda. Potom existuje množina $M \subseteq \Sigma^*$, kterou nelze induktivně zadat.*

Důkaz. Připomeňme, že induktivním zadáním množiny S rozumíme konečnou sadu odvozovacích pravidel G. Každé z těchto odvozovacích pravidel může mít pouze konečně mnoho „předpokladů“. V odvozovacím pravidle se smí vyskytovat konečný počet proměnných a konečný počet prvků abecedy.

Myšlenka důkazu spočívá v tom, že napíšeme sadu pravidel G jako slovo v jakési konečné abecedě Φ.

1. Zvolíme písmeno w a znak $'$ pro značení proměnných. Přesněji: proměnné jsou znaky w', w'', w''', ...
2. Namísto vodorovné čáry oddělující předpoklady a závěr odvozovacího pravidla budeme používat znak $=>$ a jednotlivé přepoklady oddělíme znakem &. Začátek a konec odvozovacího pravidla budeme značit symboly [a].

   Například pravidlo $\dfrac{w_1 \quad w_2}{w}$ zapíšeme jako $[w''\&w''' => w']$.

Definujeme $\Phi = \Sigma \cup \{w, ', \&, [,], =, >\}$. (Zde předpokládáme, že původní abeceda Σ neobsahuje symboly w, $'$, &, [,], = a >. To lze ovšem vždy zařídit.) Potom je jasné, že každá sada pravidel je slovo v konečné abecedě Φ. Samozřejmě, ne každé slovo v abecedě Φ je zakódovanou sadou odvozovacích pravidel. Co je podstatné, je fakt, že sad odvozovacích pravidel je pouze spočetně mnoho. Podmnožin Σ^*, které jsou induktivně zadány, je tedy také spočetně mnoho. Všech podmnožin Σ^* je ale nespočetně mnoho (protože množina Σ^* je spočetná). ■

1.5 Odbočka — funkce jako model výpočtu

Induktivní definice jsme v odstavci 1.4 zavedli pouze pro množiny slov nad konečnou abecedou. Cítíme ovšem, že bychom induktivní definice mohli definovat mnohem obecněji. Jedno takové obecnější použití vidíme při induktivním zadání *nejmenší kongruence* v poznámce 5.5.9.

Jiným (pro computer science velmi důležitým) příkladem je induktivní definice množiny *primitivně rekursivních* a *parciálně rekursivních* funkcí. Tyto třídy modelují funkce, které lze naprogramovat buď pouze pomocí cyklu `for` nebo pomocí celé síly cyklu `while` (viz poznámku 1.4.9). V tomto odstavci obě třídy přesně zavedeme (definice 1.5.2) a ukážeme jejich souvislost s programováním.

Parciální funkcí f z $\mathbb{N}^n$ do $\mathbb{N}$ (značení $f : \mathbb{N}^n \rightsquigarrow N$) rozumíme funkci $n \geq 0$ proměnných, jejíž hodnota ovšem nemusí být definována. Příklad:

$$div(x_1, x_2) = \begin{cases} \left[\dfrac{x_1}{x_2}\right], & \text{pro } x_2 \neq 0 \\ \text{nedefinováno}, & \text{pro } x_2 = 0 \end{cases}$$

je parciální funkce $div : \mathbb{N}^2 \rightsquigarrow \mathbb{N}$ (celočíselné dělení).

Co rozumíme parciální funkcí v případě, že $n = 0$? Funkce $f : \mathbb{N}^0 \rightsquigarrow \mathbb{N}$ je (pokud je definována) *výběr konstanty* v $\mathbb{N}$.

Definujme nyní množinu Par *všech parciálních funkcí* takto:

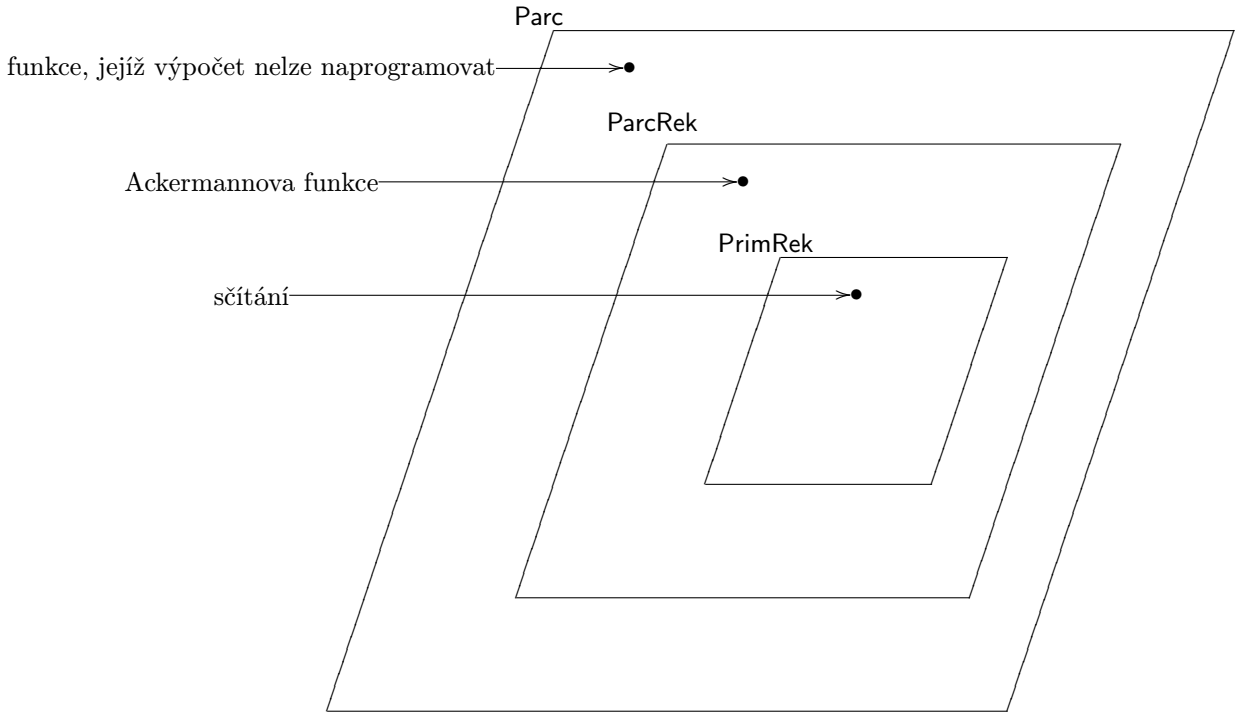
$$\mathsf{Par} = \{f \mid \text{existuje } n \text{ tak, že } f : N^n \rightsquigarrow \mathbb{N}\}$$

a pomocí deduktivních pravidel zadáme dvě její podmnožiny PrimRek (množina *primitivně rekursivních funkcí*) a ParcRek (množina *parciálně rekursivních funkcí*). Naznačíme, proč platí důležitý výsledek:

1.5.1 Věta

1. PrimRek jsou přesně ty funkce, jejichž výpočet můžeme naprogramovat v jazyce, kde máme dovoleno použít pouze cyklus `for`.
2. ParcRek jsou přesně ty funkce, jejichž výpočet můžeme naprogramovat v jazyce, kde máme dovoleno použít i cyklus `while`. Navíc, výpočet žádné obecnější funkce než funkce z množiny ParcRek naprogramovat nelze!

V dalším tak lépe vysvětlíme tento obrázek:



Nejprve zformulujeme jednotlivá pravidla pro tvorbu funkcí:

1. Axiom pro *konstantní nulu*, tj. pro funkci $zero : \mathbb{N}^0 \rightsquigarrow \mathbb{N}$ vybírající konstantu $0 \in \mathbb{N}$:

$$\frac{}{zero}\ (\text{zero})$$

2. Axiom pro *následníka*, tj. pro funkci definovanou rovností $succ(x_1) = x_1 + 1$ pro všechna x_1:

$$\frac{}{succ(x_1)}\ (\text{succ})$$

3. Axiom pro *projekci na i-tou souřadnici*, tj. pro funkci definovanou rovností $proj_i^n(x_1, \ldots, x_n) = x_i$ pro všechna n, všechna $i \leq n$ a všechna $x_1, \ldots, x_n$:

$$\frac{}{proj_i^n(x_1, \ldots, x_n)}\ (\text{proj})$$

4. Deduktivní pravidlo pro *skládání*, tj. pro každé m a n pravidlo

$$\frac{h(y_1,\ldots,y_m) \quad g_1(x_1,\ldots,x_n),\ldots,g_m(x_1,\ldots,x_n)}{h(g_1(x_1,\ldots,x_n),\ldots,g_m(x_1,\ldots,x_n))}\ \text{(comp)}$$

Toto deduktivní pravidlo tedy z funkcí h (m proměnných) a m funkcí g_1, …, g_m (každá z nich má n proměnných) vytváří novou funkci n proměnných tak, že za argumenty h dosadíme funkční hodnoty jednotlivých funkcí g_1, …, g_m.

5. Deduktivní pravidlo pro *primitivní rekursi*, tj. pravidlo

$$\frac{h(x_1,\ldots,x_{n+2}) \quad g(x_1,\ldots,x_n)}{\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}(x_1,\ldots,x_{n+1})}\ \text{(p-rec)}$$

Toto pravidlo vysvětlíme podrobněji: pro funkci $g(x_1,\ldots,x_n)$ (počáteční hodnota rekurse) vytvoří pravidlo *funkci rekursivně definovanou* „pravidlem“ $h(x_1,\ldots,x_{n+2})$ takto: funkce $\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}$ má $n+1$ proměnných a je definována následovně:

základní krok: pro všechna x_1, …, x_n platí rovnost

$$\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}(x_1,\ldots,x_n,0) = g(x_1,\ldots,x_n)$$

neboli funkční hodnota $g(x_1,\ldots,x_n)$ „odstartuje“ rekursi v poslední proměnné funkce $\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}$.

indukční krok: pro všechna x_1, …, x_n a y platí rovnost

$$\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}(x_1,\ldots,x_n,y+1) = h(x_1,\ldots,x_n,y,\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}(x_1,\ldots,x_n,y))$$

To znamená, že funkce h je „pravidlo rekurse“, které máme použít na hodnoty x_1, …, x_n, y a $\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}(x_1,\ldots,x_n,y)$, abychom dostali hodnotu funkce $\mathsf{rec}\,h\,\mathsf{from}\,g\,\mathsf{end}$ o „kousek dál“.

6. Deduktivní pravidlo pro μ-*operátor*, tj. pravidlo

$$\frac{g(x_1,\ldots,x_{n+1})}{\mu g(x_1,\ldots,x_n)}\ (\mu)$$

Zhruba řečeno, funkce $g(x_1,\ldots,x_{n+1})$ je „podmínka“ cyklu `while`, který prohledává poslední souřadnici x_{n+1} a který se zastaví právě tehdy, když $g(x_1,\ldots,x_{n+1})=0$. Přesněji:[9]

$$\mu g(x_1,\ldots,x_n) = \begin{cases} y, & \text{když } y \text{ je nejmenší hodnota, pro kterou je } g(x_1,\ldots,x_n,y)=0 \\ \text{nedefinováno}, & \text{jindy} \end{cases}$$

1.5.2 Definice Třída PrimRek *primitivně rekursivních funkcí* je induktivně zadaná axiomy (zero), (succ), (proj) a deduktivním pravidlem (p-rec).

Přidáme-li k těmto pravidlům ještě pravidlo (μ), dostaneme třídu ParcRek všech *parciálně rekursivních funkcí.*

1.5.3 Příklad Ukážeme, že funkce $plus(x,y)=x+y$ je primitivně rekursivní. Víme, že k tomu stačí nalézt syntaktický strom používající pouze pravidla (zero), (succ), (proj) a (p-rec). Intutivně chceme definovat

$$plus(x,0)=x \quad \text{a} \quad plus(x,y+1)=plus(x,y)+1$$

Přesný zápis výše uvedeného pomocí deduktivních pravidel je:

$$\dfrac{\dfrac{\dfrac{}{proj^3_3(x,y,z)}\,(\text{proj}) \quad \dfrac{}{proj^1_1(x)}\,(\text{proj}) \quad \dfrac{}{proj^1_1(y)}\,(\text{proj}) \quad \dfrac{}{succ(z)}\,(\text{succ})}{proj^3_3(proj^1_1(x),proj^1_1(y),succ(z))}\,(\text{comp}) \quad \dfrac{}{proj^1_1(x)}\,(\text{proj})}{\mathsf{rec}\,proj^3_3(proj^1_1(x),proj^1_1(y),succ(z))\,\mathsf{from}\,proj^1_1(x)\,\mathsf{end}}\,(\text{p-rec})$$

Pojďme projít jednotlivé kroky detailněji: funkci $proj^3_3(proj^1_1(x),proj^1_1(y),succ(z))$ hodláme použít jako pravidlo rekurse $h(x,y,z)$ pro definici sčítání. Náš formalismus nám nedovolil zapsat přímo $h(x,y,z)=z+1$. To je ale přesně význam funkce $proj^3_3(proj^1_1(x),proj^1_1(y),succ(z))$. Jako počátek rekurse použijeme funkci $proj^1_1(x)=x$ a rekursivní definice sčítání pak je:

[9] Zde se opět projevuje princip dobrého uspořádání: množina $\{x_{n+1}\in\mathbb{N} \mid g(x_1,\ldots,x_n,x_{n+1})=0\}$ je buď prázdná (pak je $\mu g(x_1,\ldots,x_n)$ nedefinováno) nebo neprázdná (a tudíž má nejmenší prvek $y=\mu g(x_1,\ldots,x_n)$).

1. Základní krok $plus(x,0) = x$, neboli

$$\mathsf{rec}\, proj_3^3(proj_1^1(x), proj_1^1(y), succ(z))\, \mathsf{from}\, proj_1^1(x)\, \mathsf{end}(x,0) = proj_1^1(x) = x$$

2. Rekursivní krok $plus(x, y+1) = plus(x,y) + 1$ je podle pravidel primitivní rekurse zapsán takto:

$$\begin{array}{l}\mathsf{rec}\, proj_3^3(proj_1^1(x), proj_1^1(y), succ(z))\, \mathsf{from}\, proj_1^1(x)\, \mathsf{end}(x, y+1) = \\ proj_3^3(proj_1^1(x), proj_1^1(y), succ(\mathsf{rec}\, proj_3^3(proj_1^1(x), proj_1^1(y), succ(z))\, \mathsf{from}\, proj_1^1(x)\, \mathsf{end})) = \\ succ(\mathsf{rec}\, proj_3^3(proj_1^1(x), proj_1^1(y), succ(z))\, \mathsf{from}\, proj_1^1(x)\, \mathsf{end}) = \\ \mathsf{rec}\, proj_3^3(proj_1^1(x), proj_1^1(y), succ(z))\, \mathsf{from}\, proj_1^1(x)\, \mathsf{end} + 1\end{array}$$

Předchozí příklad jasně ukazuje, že zapsání složitějších funkcí pomocí našeho formalismu může být velmi obtížné. Uvědomme si ale, že každé z pravidel odpovídá některé z konstrukcí imperativního programovacího jazyka:[10]

1. Axiomy pro nulu, následníka a projekce jsou „atomické programy“ pro výpočet hodnot funkcí.
2. Deduktivní pravidlo pro skládání odpovídá „řetězení“ jednotlivých programů v imperativním paradigmatu.
3. Deduktivní pravidlo pro primitivní rekursi je cyklus `for`: výpočet hodnoty funkce $\mathsf{rec}\, h\, \mathsf{from}\, g\, \mathsf{end}$ v bodě $(x_1, \ldots, x_n, y+1)$ se dá počítat cyklem `for` s řídící proměnnou y pomocí funkce h z hodnot $x_1, \ldots, x_n$, y a $\mathsf{rec}\, h\, \mathsf{from}\, g\, \mathsf{end}(x_1, \ldots, x_n, y)$.
4. Deduktivní pravidlo pro μ-operátor je cyklus `while` (přesněji, jde o cyklus `until`): cyklem počítáme hodnoty y, dokud neplatí rovnost $g(x_1, \ldots, x_n, y) = 0$.

Proto můžeme říci, že jsme v předchozím příkladu vlastně napsali program, který počítá hodnoty funkce $plus(x,y)$. Podobně tak můžeme naprogramovat výpočet jakékoli parciálně rekursivní funkce. Co je důležité, je tvrzení, že výpočet hodnot žádných obecnějších funkcí naprogramovat *nelze*:

1.5.4 Churchova-Turingova teze

Třída parciálních funkcí, spočítatelných nějakým algoritmem, je přesně třída parciálně rekursivních funkcí.

1.5.5 Poznámka Naše formulace Churchovy-Turingovy teze se týká parciálně rekursivních funkcí. Přísně vzato, tato teze není větou ve smyslu matematiky. V úplné obecnosti teze tvrdí, že *všechny* navržené formalizace pojmu algoritmus jsou navzájem ekvivalentní. Tak například třída problémů řešitelných programy v jazyce C++ je *stejná* jako třída parciálně rekursivních funkcí. Tento poslední výsledek ovšem matematickou větou *je* a lze jej přesně dokázat.

Churchova-Turingova teze je *nezávislá* na zvoleném paradigmatu. Tak například třída problémů řešitelných pomocí *kvantového paradigmatu* (viz poznámku 3.5.3) je stejná jako třída problémů řešitelných pomocí *sekvenčního paradigmatu*.

1.5.6 Důsledek *Existují problémy, které nelze algoritmicky řešit.*

Důkaz. Nejprve ukážeme, že množina Parc je *nespočetná*. To okamžitě plyne z toho, že nespočetná množina všech funkcí jedné přirozené proměnné je podmnožinou Parc.

Protože množina ParcRek je induktivně zadaná, je spočetná. Proto rozdíl $\mathsf{Parc} \setminus \mathsf{ParcRek}$ je nespočetná, speciálně je neprázdná. Problém výpočtu hodnot jakékoli funkce $f \in \mathsf{Parc} \setminus \mathsf{ParcRek}$ je tedy algoritmicky neřešitelná úloha. ■

[10] Tím naznačujeme důkaz věty 1.5.1. Pro přesný důkaz odkazujeme na K. Richta a J. Velebil, *Sémantika programovacích jazyků*, Karolinum, Praha 1997.

1.5.7 Poznámka Předchozí důkaz nám ukázal, že existuje alespoň jedna parciální funkce (dokonce nespočetně mnoho funkcí), jejíž funkční hodnoty nelze počítat algoritmem. Tento důkaz byl *nekonstruktivní* (viz poznámku 2.1.20) a my si můžeme položit otázku, zda můžeme nějaký konkrétní algoritmicky neřešitelný problém zformulovat. Odpověď je kladná a často jde o praktické problémy (například tzv. *Halting Problem*, neboli *problém zastavení Turingova stroje*). Přesný popis těchto problémů však přesahuje rámec těchto skript. Odkazujeme na standardní učebnici

☞ J. E. Hopcroft, R. Motwani a J. D. Ullman, *Introduction to Automata and Language Theory*, Addison-Wesley, New York, 2000

1.5.8 Poznámka Existence algoritmicky neřešitelných problémů dovoluje pochopit rozdíl mezi *determinismem* a možností *počítačové simulace*. Následující příklad *deterministického vesmíru, který nelze simulovat na počítači*, je malým zobecněním příkladu z knihy

☞ R. Penrose, *Shadows of the Mind: A Search for the Missing Science of Consciousness*, Vintage Books, 1995

Vezměme si libovolný algoritmicky neřešitelný problém $\mathbb{P}$. Pro jednoduchost předpokládejme, že jde o *rozhodovací problém*, tj. pro každou instanci vstupních dat je řešením problému buď odpověď Ano nebo Ne. Dále předpokládejme, že jsme všechny instance vstupních dat problému $\mathbb{P}$ seřadili do nekonečné posloupnosti

$$\mathbb{D}_0, \mathbb{D}_1, \ldots \mathbb{D}_n, \ldots$$

Tato posloupnost bude kódovat možné stavy vesmíru, který se bude vyvíjet v čase podle následujících *deterministických* pravidel:

1. V čase $t = 0$ je vesmír ve stavu $\mathbb{D}_0$.
2. Je-li vesmír v čase t ve stavu $\mathbb{D}_n$ a jestliže problém $\mathbb{P}$ odpoví na data $\mathbb{D}_n$ Ano, pak v čase $t + 1$ bude ve stavu $\mathbb{D}_{n+1}$.
3. Je-li vesmír v čase t ve stavu $\mathbb{D}_n$ a jestliže problém $\mathbb{P}$ odpoví na data $\mathbb{D}_n$ Ne, pak v čase $t + 1$ bude ve stavu $\mathbb{D}_{n+2}$.

Tato pravidla pro vývoj vesmíru jsou deterministická: v každém čase je přesně popsán následující stav vesmíru. Přesto vývoj tohoto vesmíru *nelze* simulovat na počítači. Počítačová simulace by totiž znamenala, že máme k disposici program, který umí řešit problém $\mathbb{P}$.

1.6 Cvičení

1.6.1 Cvičení Zformulujte v moderním jazyce příklad rovnosti, kterou popisuje Leibniz v dopise královně Žofii Charlottě (poznámka 1.1.5), a dokažte ji indukcí. Vyznačte pečlivě indukční předpoklad a dekomposici problému.

1.6.2 Cvičení Dokažte matematickou indukcí:

1. $\sum_{k=0}^{n} k! \cdot k \; = (n+1)! - 1$ pro všechna přirozená čísla n.
2. $\sum_{k=1}^{n} (-1)^{k+1} \cdot k^2 = \frac{(-1)^{n+1} n(n+1)}{2}$ pro všechna $n \geq 1$.
3. $\sqrt{0} + \sqrt{1} + \ldots + \sqrt{n} < \frac{2}{3}(n+1)\sqrt{n+1}$ pro všechna přirozená čísla n.
4. $\frac{1}{n+1} + \frac{1}{n+2} + \ldots + \frac{1}{2n} > \frac{1}{2}$ pro všechna přirozená čísla $n \geq 2$.

5. $11^n - 6$ je dělitelné 5 pro všechna $n \geq 1$.
6. $3^n + 7^n - 2$ je dělitelné 8 pro všechna $n \geq 1$.
7. $(1+x)^n \geq 1 + nx$ pro libovolné $x \geq 0$ a libovolné $n \geq 1$.
8. $\left|\sum_{i=1}^{n} x_i\right| \leq \sum_{i=1}^{n} |x_i|$ pro libovolné $n \geq 1$ a libovolná reálná čísla $x_1, \ldots, x_n$.

U všech důkazů pečlivě vyznačte indukční předpoklad a dekomposici problému.

1.6.3 Cvičení Využitím principu indukce odvoďte vzorec pro $\sum_{i=0}^{n} i^2$.

Návod:

1. Suma se „chová jako integrál“, proto platí $\sum_{i=0}^{n} i^2 = An^3 + Bn^2 + Cn + D$. (Uvedený vztah lze odvodit z rovnosti $\sum_{i=0}^{n+1} i^2 - \sum_{i=0}^{n} i^2 = (n+1)^2$.)
2. Protože $\sum_{i=0}^{0} i^2 = 0$, platí $D = 0$.
3. Protože platí $\sum_{i=0}^{n+1} i^2 = (n+1)^2 + \sum_{i=0}^{n} i^2$, musí platit rovnost

$$A(n+1)^3 + B(n+1)^2 + C(n+1) = (n+1)^2 + An^3 + Bn^2 + Cn$$

pro všechna $n \geq 0$. Odtud porovnáním koeficientů u jednotlivých mocnin odvoďte soustavu rovnic

$$\begin{array}{rcrcrcl} 3A & & & & & = & 1 \\ 3A & + & 2B & & & = & 2 \\ A & + & B & + & C & = & 1 \end{array}$$

a vyřešte ji.

1.6.4 Cvičení Pro $t \geq 1$, $s \geq 1$ označte $A(t,s) = \sum_{k=1}^{t} \sum_{j=k}^{s} 1$. Odhadněte vzorec pro $A(t,s)$, který neobsahuje sumu. Svoji hypotézu dokažte indukcí.

1.6.5 Cvičení Dokažte, že pro Ackermannovu funkci z poznámky 1.4.9 platí:

1. $A(1,n) = 2 + (n+3) - 3$, pro všechna $n \geq 0$.
2. $A(2,n) = 2 \cdot (n+3) - 3$, pro všechna $n \geq 0$.
3. $A(3,n) = 2 \uparrow (n+3) - 3$, pro všechna $n \geq 0$, kde $a \uparrow b$ značí mocninu a^b.
4. $A(4,n) = 2 \uparrow\uparrow (n+3) - 3$, pro všechna $n \geq 0$. Symbolem $2 \uparrow\uparrow (n+3)$ značíme „věž“ mocnin čísla 2 výšky $(n+3)$, například $2 \uparrow\uparrow 3 = 2^{2^2} = 2^4 = 16$, $2 \uparrow\uparrow 4 = 2^{2^{2^2}} = 2^{16} = 65\,536$.

Z předchozího je vidět, že Ackermannova funkce roste velmi rychle (rychleji než každá primitivně rekursivní funkce). Navrhněte explicitní vzorec pro výpočet $A(5,n)$. (Nabízí se formulka $A(5,n) = 2 \uparrow\uparrow\uparrow (n+3) - 3$, co ale znamená? [11] Zkuste na to přijít před přečtením cvičení 1.6.6.)

[11] Exponenciální značení s využitím jistého počtu šipek $\uparrow$ pochází od Donalda Ervina Knutha (nar. 1938), stejně jako typografický systém TEX, s jehož pomocí je tento text napsán.

1.6.6 Cvičení Indukcí dokažte, že pro všechna přirozená čísla a, b platí (značení s využitím $\uparrow$ je vysvětleno ve cvičení 1.6.5):

$$a \uparrow\uparrow 1 = a \qquad a \uparrow\uparrow (b+1) = a \uparrow (a \uparrow\uparrow b)$$

A obecně pro přirozená čísla a, b definujte

$$a \underbrace{\uparrow \ldots \uparrow}_{(k+1)\text{-krát}} 1 = a \qquad a \underbrace{\uparrow \ldots \uparrow}_{(k+1)\text{-krát}} (b+1) = a \underbrace{\uparrow \ldots \uparrow}_{k\text{-krát}} (a \underbrace{\uparrow \ldots \uparrow}_{(k+1)\text{-krát}} b)$$

kde k je přirozené číslo a kde „nula šipek“ mezi a a b znamená součin a a b.

Dokažte, že pro Ackermannovu funkci platí obecný vzorec:

$$A(m,n) = 2 \underbrace{\uparrow \ldots \uparrow}_{(m-2)\text{-krát}} (n+3) - 3, \qquad m \geq 2$$

O vztahu Ackermannovy funkce s obrovskými čísly a o zobecňování mocnin se lze dočíst například na webové stránce

☞ http://home.earthling.net/~mrob/pub/math/largenum-3.html

a o největším přirozeném čísle s praktickým využitím v matematice (takzvaném *Grahamově čísle*) na

☞ http://mathworld.wolfram.com/GrahamsNumber.html

1.6.7 Cvičení Reálné funkci f, která vyhovuje rovnosti $f(x+y) = f(x) + f(y)$ pro všechna reálná čísla x, y, se říká *aditivní*.

1. Udejte *dva* různé příklady aditivní funkce.
2. Dokažte, že pro každou aditivní funkci f, každé přirozené číslo $n > 0$ a každé reálné číslo x platí rovnost $f(nx) = nf(x)$.

1.6.8 Cvičení Ať $n \geq 2$. Termínem *n-turnaj* v tomto cvičení označíme turnaj n družstev, kde hraje každý s každým a žádná hra nemůže skončit remízou.

Označte jako $P(n)$ následující tvrzení:

> *Po ukončení jakéhokoli n-turnaje lze setřídit zúčastněná družstva do posloupnosti d_1, d_2, …, d_n tak, že d_1 porazilo d_2, d_2 porazilo d_3, …, d_{n-1} porazilo d_n.*

Dokažte, že pro všechna $n \geq 2$ platí $P(n)$.

1.6.9 Cvičení Ať $n \geq 2$. Termínem *n-turnaj* v tomto cvičení označíme turnaj n družstev, kde hraje každý s každým a žádná hra nemůže skončit remízou.

Družstvu d říkáme *pseudovítěz* n-turnaje, když pro libovolné družstvo $d' \neq d$ platí

> buď: družstvo d porazilo družstvo d',
>
> nebo: existuje družstvo d'' takové, že družstvo d'' porazilo družstvo d'.

Vyřešte následující problémy:

1. Indukcí dokažte, že každý n-turnaj ($n \geq 2$) má pseudovítěze.
2. Navrhněte výsledky 5-turnaje, kde každé družstvo je pseudovítězem.
3. Indukcí dokažte, že pro každé $n \geq 5$ existuje výsledek n-turnaje, kde každé družstvo je pseudovítězem.

1.6.10 Cvičení Co je špatně na následujícím důkazu?

Definujme množinu S takto: S obsahuje číslo 1 a všechna přirozená čísla $n \geq 1$, pro která platí, že $n = n+1$. Zřejmě platí:

1. $1 \in S$ (tak jsme množinu S definovali).

2. Jestliže $n \in S$, pak $n = n + 1$. Tedy $n + 1 = (n + 1) + 1$. Tudíž platí $n + 1 \in S$.

Podle slabého principu indukce je $S = \{1, 2, \ldots\}$.

1.6.11 Cvičení Co je špatně na následujícím důkazu?

1. Je-li maximum dvou přirozených čísel 0, pak jsou si obě čísla rovna.

2. Předpokládejme, že je-li maximum dvou přirozených čísel n, pak jsou si rovna. Vezměme nyní dvě přirozená čísla a, b taková, že jejich maximum je $n + 1$. Pak maximum čísel $a - 1$ a $b - 1$ je n a podle předpokladu je $a - 1 = b - 1$. Tudíž $a = b$.

Podle slabého principu indukce jsou si všechna přirozená čísla rovna.

1.6.12 Cvičení Ukažte, že výraz $S_n = (n + 2)(n - 1)$ vyhovuje indukčnímu kroku pro (nesprávný) vzorec $2 + 4 + \ldots + 2n = S_n$.

1.6.13 Cvičení Přepište silný princip indukce do množinového tvaru. Návod: postupujte analogicky jako v poznámce 1.1.10. Automat na množině přirozených čísel, který byste měli sestavit, nebude pracovat deterministicky, tj. z čísla n do čísla i se lze dostat v jednom výpočetním kroku právě tehdy, když $i < n$.

1.6.14 Cvičení *Princip silné indukce pro reálná čísla:* Abychom dokázali, že všechna reálná čísla $x \geq 0$ mají vlastnost V, stačí ukázat následující dvě věci:

1. Základní krok: číslo 0 má vlastnost V.

2. Jestliže všechna čísla y, pro která platí $0 \leq y < x$, mají vlastnost V, pak i číslo x má vlastnost V.

Je tento princip silné indukce pro reálná čísla správný?

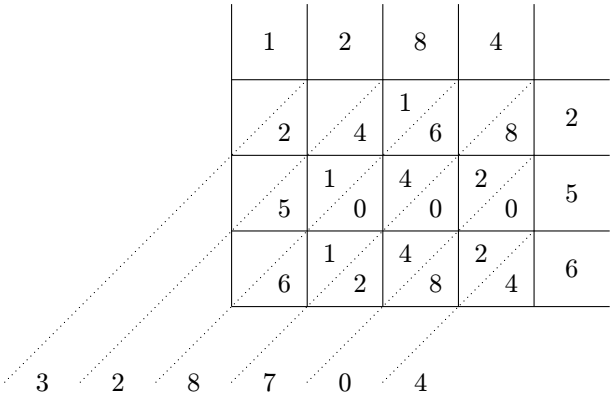
1.6.15 Cvičení Následující algoritmus pro násobení čísel popisuje Abu Ja'far Mohammed ibn Mûsa al-Khowârizm v knize *Kitab al-jabr wa'l-muquabala*. Vynásobíme čísla 1 284 a 256. Nejprve napíšeme obě čísla do záhlaví následující tabulky:

1	2	8	4	
				2
				5
				6

Nyní vynásobíme záhlaví každého řádku se záhlavím každého sloupce. Jednotky přitom píšeme do spodního a desítky do horního trojúhelníku příslušného políčka. Dostaneme tak tabulku:

1	2	8	4	
2	4	1 / 6	8	2
5	1 / 0	4 / 0	2 / 0	5
6	1 / 2	4 / 8	2 / 4	6

Nyní jednotlivé „pásy“ (v pořadí zprava doleva) sečteme a známým způsobem přenášíme desítky:



Spočetli jsme, že $1\,284 \cdot 256 = 328\,704$. Proč je tento algoritmus korektní?

1.6.16 Cvičení Ukažte, že každá z posloupností $\left(\frac{1+\sqrt{5}}{2}\right)^n$ a $\left(\frac{1-\sqrt{5}}{2}\right)^n$, $n \geq 1$ řeší rekurentní rovnici

$$S(n+2) - S(n+1) - S(n) = 0, \text{ pro } n \geq 1.$$

1.6.17 Cvičení Zformulujte přesně (podobně jako v 1.3.5) jak hledat reálný fundamentální systém.

1.6.18 Cvičení Pro následující homogenní rekurentní rovnice nalezněte fundamentální systém. V případě, že fundamentální systém vyjde komplexní, nalezněte i reálný fundamentální systém.

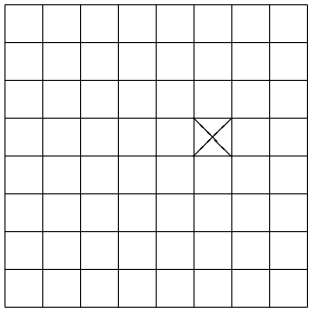
1. $X(n+1) + 3X(n) = 0$, $n \geq 1$.
2. $X(n+2) - 5X(n+1) + 6X(n) = 0$, $n \geq 11$.
3. $X(n+2) + X(n+1) + X(n) = 0$, $n \geq 17$.
4. $X(n+3) - 7X(n+2) + 16X(n+1) - 12X(n) = 0$, $n \geq 100$.
5. $X(n+3) + 2X(n+2) + 2X(n+1) + X(n) = 0$, $n \geq 5$.

1.6.19 Cvičení Každou z rovnic ze cvičení 1.6.18 považujte za homogenní rovnici příslušnou následujícím pravým stranám:

1. $(-3)^n$.
2. $(2)^n \cdot (n-7)$.
3. $\sin n \cdot (n^2 - 4)$.
4. $1 + 2^n + 3^n$.
5. $(n+6)(1+3^{2n})$.

U každé z pětadvaceti takto vzniklých nehomogenních rovnic nalezněte partikulární řešení.

1.6.20 Cvičení V tomto cvičení vyřešíte jednu úlohu o *parketáži*. *Místností rozměru n* budeme rozumět šachovnici rozměru $2^n \times 2^n$, ze které je jedno (libovolné) pole vyjmuto. Toto je příklad místnosti rozměru 3:



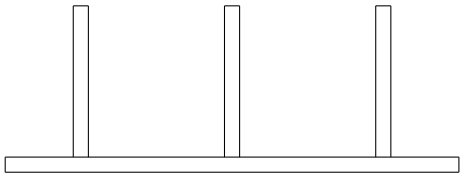
Trimino je parketa následujícího tvaru:

Vyřešte následující problémy:

1. Sestavte rekursivní algoritmus, který vyparketuje libovolnou místnost rozměru $n \geq 2$ triminy. Vyparketováním triminy rozumíme rozložení trimin po místnosti tak, aby žádné pole nezbylo volné. Trimina se přitom mohou pouze dotýkat hranou a nesmí pokrýt vyjmuté pole.
2. Ukažte (indukcí) totální korektnost tohoto algoritmu.
3. Označte jako $P(n)$ počet triminových parket, nutných k vyparketování místnosti rozměru $n \geq 2$. Na základě rekursivního algoritmu sestavte rekurentní rovnici pro $P(n)$ a vyřešte ji.

1.6.21 Cvičení *Problém hanojských věží* (anglicky: *Towers of Hanoi*) pro n disků je následující:

1. Na vodorovné desce jsou tři svislé jehly.



2. Je dáno n disků s navzájem různými průměry. Každý disk má uprostřed otvor, aby šel navléci na jehlu.

   Těchto n disků je na počátku navlečeno na levou jehlu. Přitom největší disk je vespodu a pak jsou postupně na jehlu navlečeny disky se zmenšujícími se průměry.
3. *Povolený tah* je přemístění jednoho disku z jedné jehly na jinou. Po ukončení tahu *nesmí* být disk s větším průměrem navlečen *nad* disk s menším průměrem.
4. Úloha zní: povolenými tahy přestěhujte n disků na pravou jehlu. Ve výsledné konfiguraci potom musí být největší disk opět vespodu a na něm postupně na jehlu navlečeny disky se zmenšujícími se průměry.

Vyřešte následující problémy:

1. Sestavte rekursivní algoritmus, který vyřeší problém hanojských věží pro n disků, $n \geq 1$.
2. Ukažte (indukcí) totální korektnost tohoto algoritmu.
3. Označte jako $H(n)$ počet povolených tahů, nutných k vyřešení problému hanojských věží pro n disků, $n \geq 1$. Na základě rekursivního algoritmu sestavte rekurentní rovnici pro $H(n)$ a vyřešte ji.

1.6.22 Cvičení Posloupnost $\{F(n)\}_{n=0}^{\infty}$ *Fibonacciho čísel* je rekurentně definována takto:

$$\begin{aligned} F(n+2) &= F(n+1) + F(n),\ n \geq 0 \\ F(0) &= 0 \\ F(1) &= 1 \end{aligned}$$

Odvoďte explicitní vzorec pro n-té Fibonacciho číslo $F(n)$:

$$F(n) = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right), \quad n \geq 0 \tag{1.17}$$

Vyřešte následující dvě úlohy jak s použitím pouhé rekurentní definice čísel $F(n)$, tak s použitím explicitního vzorce pro $F(n)$:

1. Vyjádřete součet $\sum_{k=0}^{n} F(k)$ pomocí $F(n+2)$.
2. Ukažte, že pro každé $n \geq 1$ platí nerovnost
$$F(n) \geq \left(\frac{1+\sqrt{5}}{2}\right)^{n-1}$$
3. Ukažte, že pro každé $n \geq 0$ platí rovnost $F(n+1)F(n+1) - F(n)F(n+2) = (-1)^n$.

1.6.23 Cvičení Definujte pojem syntaktického stromu.

1.6.24 Cvičení Z předmětu *Matematická logika* znáte syntaxi formulí výrokové logiky. Tyto formule jsou vytvářené z neprázdné množiny *At atomických formulí* pomocí *základních logických spojek* $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$. Zadejte syntaxi formulí výrokové logiky pomocí deduktivních pravidel a pomocí syntaktických stromů.

1.6.25 Cvičení Ať $\Sigma = \{a, b, c\}$ je abeceda. Zadejte induktivně množinu L všech slov, která obsahují slovo *bbca* jako podslovo. Indukcí dokažte korektnost definice.

1.6.26 Cvičení Ať $\Sigma = \{a, b\}$ je abeceda. Zadejte induktivně množinu L všech slov, která se čtou zleva doprava stejně jako zprava doleva (takovým slovům se říká *palindromy*). Indukcí dokažte korektnost definice.

1.6.27 Cvičení Ať $\Sigma = \{0, 1, 2, \ldots, 9\}$.

1. Zadejte induktivně množinu N všech slov nad Σ, která representují kladná přirozená čísla. (Tato úloha vyžaduje přesně zformulovat vlastnost „slovo $w \in \Sigma^*$ representuje kladné přirozené číslo“.)
2. Definujte rekursivně funkci $s : N \longrightarrow N$ tak, aby slovo $s(w)$ representovalo následníka toho přirozeného čísla, které je representováno slovem w.

1.6.28 Cvičení Ať $k \geq 1$ a ať $\Sigma = \{a_1, a_2, \ldots, a_k\}$ je konečná abeceda. Ukažte, že sada pravidel
$$\frac{}{\varepsilon}\ (0) \quad | \quad \frac{w}{wa_1}\ (1) \quad | \quad \frac{w}{wa_2}\ (2) \quad | \quad \ldots \quad | \quad \frac{w}{wa_k}\ (\mathrm{k})$$
definuje induktivně množinu Σ^* všech slov nad abecedou Σ.

1.6.29 Cvičení Pro libovolné přirozené číslo $k \geq 2$ platí: pokud slovo v začíná a končí stejným znakem, a pokud $|v|_a = |v|_b = k$, pak existují neprázdná slova v_1, v_2 taková, že $v = v_1 v_2$, $|v_1|_a = |v_1|_b$ a $|v_2|_a = |v_2|_b$. Dokažte.

Revize kapitoly

Dozvěděli jsme se:

- ✔ Existuje řada principů indukce pro přirozená čísla, všechny jsou *navzájem ekvivalentní.*
- ✔ Důkaz indukcí *je* rekursivní algoritmus. Jako takový pracuje směrem dolů: od $n+1$ k n pomocí *dekomposice.*
- ✔ Indukcí lze dokazovat pouze tvrzení mající formu

  Pro všechna přirozená čísla $n \geq n_0$ platí tvrzení $V(n)$.
- ✔ Princip indukce má v computer science výsadní postavení: zaručuje například *terminaci rekursivních algoritmů*, viz pojem *variantu.* Nalezení variantu může být velmi obtížné, viz *Collatzův problém.*

- ✔ S principy indukce úzce souvisí *rekurentní rovnice*. Lineární rekurentní rovnice se řeší velmi podobně jako lineární *diferenciální* rovnice.
- ✔ Indukční principy lze chápat i jako *definiční techniku*. Lze tak například induktivně definovat nejrůznější množiny slov nad zadanou (konečnou) abecedou. Chceme-li ukázat *korektnost* induktivní definice pro množiny slov, je zapotřebí provést *dva* důkazy indukcí.

Pro přípravu na zkoušku zkuste zodpovědět následující otázky:

- ✔ Zvolte si svůj oblíbený rekursivní algoritmus (ne výpočet faktoriálu) a oddůvodněte jeho terminaci pomocí nalezení variantu. Vysvětlete přesně, kde používáte princip indukce.
- ✔ Navrhněte rekursivní algoritmus, který *neterminuje*. Vysvětlete proč.
- ✔ Podrobně si promyslete způsob řešení lienárních rekurentních rovnic s konstantními koeficienty.

Doplňující literatura

O principech indukce se lze dočíst v téměř každé učebnici matematiky pro computer science, doporučujeme například knihu

- ☞ L. Childs, *A Concrete Introduction to Higher Algebra*, Springer, New York, 1995

a pro zásobu „klasických“ příkladů jakoukoli sbírku pro střední školu. K literatuře o indukci je možné ještě přidat knihu

- ☞ T. Forster, *Logic, Induction and Sets*, London Mathematical Society, 2003

ve které je strukturální indukce dále zobecněna.

Koho zajímají otázky okolo korektnosti algoritmů, tomu lze doporučit například knihu

- ☞ R. C. Backhouse, *Program Construction and Verification*, Prentice-Hall, London, 1986

kde je podrobně vysvětlena práce s variantem a invariantem nebo

- ☞ J. Velebil, *Logika programů*, http://math.feld.cvut.cz/velebil/

což je text věnovaný aplikacím (modální) logiky na korektnost programů.

O vztahu rekursivních algoritmů a principů indukce pojednává například kapitola 2 knihy:

- ☞ S. Dvořák, *Dekompozice a rekursivní algoritmy*, nakladatelství Grada, Praha, 1992

kde se dozvíte i další o problému hanojských věží, viz cvičení 1.6.21.

Více se o teorii, způsobech řešení a aplikacích *rekurentních rovnic* lze dočíst v knihách

- ☞ J. Kaucký, *Kombinatorické identity*, Veda, Bratislava, 1975
- ☞ W. G. Kelley a A. C. Peterson, *Difference Equations: An Introduction with Applications*, Academic Press Inc., New York, 1991

Přímé aplikace rekurentních rovnic v teorii algoritmů lze najít například v knize

- ☞ T. H. Cormen, C. E. Leiserson, R. L. Rivest a C. Stein, *Introduction to Algorithms*, MIT Press, Massachusetts, 2005

O obecném problému parketáže (*tiling problem*), načatém ve cvičení 1.6.20 (viz také příklad 2.1.18), se lze mnoho zajímavého dozvědět v knihách

- ☞ R. Penrose, *Emperor's New Mind: Concerning Computers, Minds and The Laws of Physics*, Oxford University Press, 1989
- ☞ R. Penrose, *Shadows of the Mind: A Search for the Missing Science of Consciousness*, Vintage Books, 1995

Kapitola 2

Počítání modulo

When the clock strikes twelve, we'll cool off then,
Start a rockin' round the clock again.
Bill Haley, *Rock Around The Clock*

V této části zavedeme nová „čísla“, která budou později hrát roli reálných čísel při řešení problémů, které jsme zvyklí řešit v lineární algebře.

Naše úvahy nás povedou k následující otázce: které vlastnosti jsou pro klasickou lineární algebru (počítání s reálnými čísly) podstatné? Výsledné definice *okruhu* — definice 2.3.14 — a *tělesa* — definice 2.3.17 pak jsou abstrakcí těchto vlastností.

2.1 Dělitelnost v oboru přirozených a v oboru celých čísel

Nejprve připomeneme některé známé pojmy z dělitelnosti přirozených a celých čísel.

2.1.1 Definice Řekneme, že *přirozené číslo a dělí přirozené číslo b* (tento fakt značíme $a \mid b$), pokud existuje přirozené číslo n takové, že $b = n \cdot a$.

Řekneme, že *celé číslo a dělí celé číslo b*, (tento fakt též značíme $a \mid b$), pokud existuje celé číslo n takové, že $b = n \cdot a$.

2.1.2 Poznámka Pokud a dělí číslo b (v oboru přirozených čísel nebo celých čísel), pak číslo a nazveme *dělitelem* čísla b.

Zdůrazněme, že podle výše uvedené definice je číslo 0 dělitelné číslem 0.

Připomeňme, že přirozenému číslo p většímu než 1 říkáme *prvočíslo*, pokud je číslo p dělitelné pouze čísly 1 a p.

2.1.3 Lemma *Množina všech prvočísel $\mathbb{P}$ je nekonečná množina.*

Důkaz. [1] Předpokládejme, že $\mathbb{P} = \{p_1, \ldots, p_n\}$. Definujeme přirozené číslo p takto: $p = p_1 \cdot p_2 \cdot \ldots \cdot p_{n-1} \cdot p_n + 1$. Zřejmě $p \notin \mathbb{P}$. Je-li p prvočíslo, jsme hotovi — původní množina $\{p_1, \ldots, p_n\}$ nemohla obsahovat všechna prvočísla. Je-li p složené číslo, musí se v jeho prvočíselném rozkladu vyskytovat prvočíslo, které není v množině $\{p_1, \ldots, p_n\}$.

V každém případě množina $\{p_1, \ldots, p_n\}$ neobsahuje všechna prvočísla, množina $\mathbb{P}$ musí být nekonečná. ■

[1] Tento důkaz je připisován Eukleidovi. Eukleides byl řecký matematik, který žil přibližně ve 3. století př.n.l., a který je autorem pozoruhodné knihy $\Sigma\tau o\iota\chi\epsilon\iota\omega\eta$ (*Základy*, v českém překladu kniha vyšla v roce 1907), ze které pochází nejen tento důkaz, ale i algoritmus pro výpočet největšího společného dělitele, viz stranu 49. V knize *Základy* také Eukleides zformuloval axiomatickou geometrii, viz poznámku 6.3.7.

2.1.4 Definice *Prvočíselným rozkladem* přirozeného čísla x rozumíme rovnost

$$x = p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r},$$

kde $r \geq 1$ je přirozené číslo, $p_1 < p_2 < \ldots < p_r$ jsou navzájem různá prvočísla a n_1, n_2, …, n_r jsou kladná přirozená čísla.

Následující věta říká, že prvočísla tvoří základní „stavební kameny“ přirozených čísel.

2.1.5 Věta (Základní věta elementární teorie čísel)
Pro každé přirozené číslo $x \geq 2$ existuje jednoznačný prvočíselný rozklad.

Důkaz. Existenci prvočíselného rozkladu jsme dokázali v příkladu 1.1.11. Jednoznačnost prvočíselného rozkladu ukážeme indukcí podle x.

1. **Základní krok.** Pro $x = 2$ zřejmě tvrzení platí: $2 = 2^1$ je jednoznačně určený prvočíselný rozklad.

2. **Indukční krok.** Předpokládejme, že všechna přirozená čísla k, pro která je $2 \leq k < x + 1$, mají jednoznačně určený prvočíselný rozklad. Dokazujeme nyní jednoznačnost prvočíselného rozkladu pro přirozené číslo $x + 1$.

   Ať $x + 1 = p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r}$ a $x + 1 = q_1^{m_1} \cdot q_2^{m_2} \cdot \ldots \cdot q_s^{m_s}$ jsou dva prvočíselné rozklady čísla $x + 1$. Z definice je $p_1 < p_2 < \ldots < p_r$ a $q_1 < q_2 < \ldots < q_s$.

   Mohou nastat dva případy:

   (a) $x + 1$ je prvočíslo. Pak $s = r = 1$ a $x + 1 = p_1 = q_1$. Rozklad je určen jednoznačně.

   (b) $x + 1$ je složené číslo. Pak mohou nastat tři případy:

      i. $p_1 = q_1$. Označme tuto společnou hodnotu jako p. Potom lze číslo $x + 1$ vydělit číslem p^n, kde n je menší z čísel n_1, m_1.
      Označme $\dfrac{x+1}{p^n} = a$. Podle indukčního předpokladu má číslo a jednoznačně určený prvočíselný rozklad. Protože $a = p^{n_1-n} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r}$ a $a = p^{m_1-n} \cdot q_2^{m_2} \cdot \ldots \cdot q_s^{m_s}$ jsou dva prvočíselné rozklady čísla a, musí být splněno $r = s$, $n_1 - n = m_1 - n$ a dále $p_2 = q_2$, $n_2 = m_2$, $\ldots p_r = q_r$, $n_r = m_r$. Protože z rovnosti $n_1 - n = m_1 - n$ plyne, že $n_1 = m_1$, je prvočíselný rozklad čísla $x + 1$ určen jednoznačně.

      ii. $p_1 < q_1$. Ukážeme, že tato situace nemůže nastat. Definujme číslo a následovně:

      $$a = q_1^{m_1} \cdot q_2^{m_2} \cdot \ldots \cdot q_s^{m_s} - p_1 \cdot q_1^{m_1-1} \cdot q_2^{m_2} \cdot \ldots \cdot q_s^{m_s}$$

      Protože platí

      $$a = p_1 \cdot (p_1^{n_1-1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r} - q_1^{m_1-1} \cdot q_2^{m_2} \cdot \ldots \cdot q_s^{m_s})$$

      musí být v prvočíselném rozkladu čísla a být použito prvočíslo p_1. Protože platí

      $$a = (q_1 - p_1) \cdot q_1^{m_1-1} \cdot q_2^{m_2} \cdot \ldots \cdot q_s^{m_s}$$

      a protože $p_1 \neq q_i$ pro $i \in \{1, \ldots, s\}$, musí p_1 dělit číslo $q_1 - p_1$. Tudíž pro nějaké přirozené číslo k platí $p_1 \cdot k = q_1 - p_1$, neboli $q_1 = p_1 \cdot (k + 1)$. To je spor, q_1 je přece prvočíslo.

      iii. $p_1 > q_1$. Tento případ se řeší analogicky jako poslední případ.

■

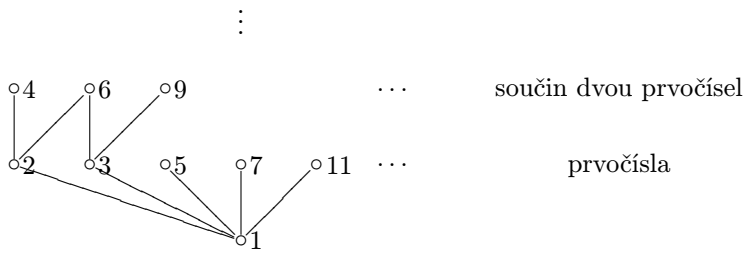
2.1.6 Příklad Existence a jednoznačnost prvočíselného rozkladu umožňuje systematicky vyhledávat dělitele daného přirozeného čísla. Například číslo 1960 má prvočíselný rozklad $2^3 \cdot 5 \cdot 7^2$. Každý dělitel čísla 1960 lze zakódovat jako vektor o třech složkách, první souřadnice tohoto vektoru říká, kolikrát bylo použito prvočíslo 2, druhá složka vektoru říká, kolikrát bylo použito prvočíslo 5, a třetí složka říká, kolikrát bylo použito prvočíslo 7. Vektor

$$(2, 0, 1)$$

tak například kóduje dělitele $2^2 \cdot 5^0 \cdot 7^1 = 28$.

Celkově tedy můžeme usoudit, že číslo 1960 má celkem $4 \cdot 2 \cdot 3 = 24$ různých dělitelů. To je totiž počet různých vektorů délky 3, jejichž první složka může být obsazena čtyřmi různými způsoby (totiž čísly 0, 1, 2, 3), druhá složka dvěma různými způsoby a třetí složka třemi různými způsoby. Viz také cvičení 2.4.6.

Ve cvičení 2.4.1 se ukazuje, že množina všech přirozených čísel $\mathbb{N}$ spolu s binární relací $\mid$ tvoří uspořádanou množinu (viz definici 6.1.1).[2] Můžeme tedy na fakt, že $a \mid b$ nahlížet abstraktně jako na fakt, že „a je menší nebo rovno b“. *Hasseho diagram* (viz poznámku 6.1.6) tohoto posetu vypadá takto:



Očíslujeme-li jednotlivé hladiny Hasseho diagramu vzestupně počínaje indexem 0, je hladina s indexem n obsazena přesně všemi čísly, která jsou součinem n prvočísel. Číslo 1960 je vytvořeno součinem šesti prvočísel (rozklad je $2 \cdot 2 \cdot 2 \cdot 5 \cdot 7 \cdot 7$), proto se v Hasseho diagramu objeví až na hladině s indexem 6.

2.1.7 Definice Řekneme, že *přirozené číslo d je největším společným dělitelem přirozených čísel a, b* (značení $d = \gcd(a, b)$), pokud jsou splněny následující dvě podmínky:[3]

1. Číslo d *je společným dělitelem čísel a, b*, tj. platí, $d \mid a$ a současně $d \mid b$ (v oboru přirozených čísel).
2. Číslo d *je největším ze všech společných dělitelů čísel a, b*, tj. platí následující: je-li c takové přirozené číslo, pro které platí $c \mid a$ a současně $c \mid b$, potom $c \mid d$.

Pokud $\gcd(a, b) = 1$, řekneme, že přirozená čísla a, b jsou *nesoudělná*.

2.1.8 Poznámka Využijeme-li faktu, že $\mathbb{N}$ spolu s relací $\mid$ tvoří poset (cvičení 2.4.1), potom definice největšího společného dělitele $\gcd(a, b)$ čísel a, b je přesně definice infima v tomoto posetu.

Známe-li prvočíselný rozklad čísel a, b, je nalezení největšího společného dělitele snadné.

2.1.9 Příklad Pro čísla $a = 1960 = 2^3 \cdot 5 \cdot 7^2$ a $b = 308 = 2^2 \cdot 7 \cdot 11$ je $\gcd(a, b) = 2^2 \cdot 7 = 28$. Z obou prvočíselných rozkladů stačí vzít všechna společná prvočísla v maximální společné mocnině.

Znalost prvočíselného rozkladu tedy umožňuje řešit celou řadu úloh o dělitelnosti. Nalezení prvočíselného rozkladu je však obecně velmi těžký problém — na tomto faktu je založen například šifrovací protokol RSA (šifrování s veřejným klíčem), viz odstavec 3.5.

Ukážeme, že při hledání největšího společného dělitele se znalosti prvočíselných rozkladů lze vyhnout. Následující tvrzení bude základem pro další úvahy o dělitelnosti.

2.1.10 Tvrzení (Dělení se zbytkem v oboru celých čísel)
Ať a, b jsou libovolná celá čísla, $b \neq 0$. Pak existují jednoznačně určená celá čísla q a r taková, že jsou splněny následující dvě podmínky:

1. *Platí rovnost $a = q \cdot b + r$.*
2. *Číslo r splňuje nerovnost $0 \leq r < |b|$.*

Důkaz. Nejprve ukážeme existenci takových čísel q a r. Protože platí $(-q) \cdot (-b) = q \cdot b$, můžeme bez újmy na obecnosti přepokládat, že $b > 0$. Budeme rozlišovat dva případy:

Číslo a je větší nebo rovno 0. Budeme postupovat indukcí podle a.

[2] Budeme používat obvyklejší slovo *poset* z anglického *partially ordered set*.
[3] Anglicky *greatest common divisor*, odtud pochází značení.

Je-li $a = 0$, je tvrzení triviální, protože můžeme zvolit $q = r = 0$.

Předpokládejme, že pro $a \geq 0$ existují celá čísla q, r taková, že $a = q \cdot b + r$ a $0 \leq r < |b| = b$.

Potom $a + 1 = q \cdot b + (r + 1)$, kde $r + 1 \leq b$. Je-li $r + 1 < b$, jsme hotovi. Je-li $r + 1 = b$, pak $a + 1 = q \cdot b + b = (q + 1) \cdot b + 0$.

Číslo a je menší než 0. Podle předchozího existují celá čísla q', r' taková, že $-a = q' \cdot b + r'$ a $0 \leq r' < b$. Opět rozlišíme dva případy.

Jestliže $r' = 0$, položme $q = -q'$ a $r = 0$. (Platí totiž $a = (-q') \cdot b + 0$.)

Jestliže $r' > 0$, položme $q = -q' - 1$ a $r = b - r'$. Platí totiž $a = (-q' - 1) \cdot b + (b - r')$ a $0 < (b - r') < b$.

Zbývá ukázat, že čísla q a r jsou určena čísly a a b jednoznačně. Předpokládejme, že jsme vyjádřili číslo a dvěma způsoby, tj. předpokládejme, že existují celá čísla q_1, q_2, r_1 a r_2 taková, že platí

$$a = q_1 \cdot b + r_1 \qquad 0 \leq r_1 < b$$
$$a = q_2 \cdot b + r_2 \qquad 0 \leq r_2 < b$$

Potom platí $q_1 \cdot b + r_1 = q_2 \cdot b + r_2$ a tudíž $(q_1 - q_2) \cdot b = r_2 - r_1$. Protože platí $0 \leq r_1 < b$ a $0 \leq r_2 < b$, je $|r_2 - r_1| < b$. Tudíž je $|(q_1 - q_2) \cdot b| < b$, tedy $q_1 = q_2$. Z toho plyne, že $r_1 = r_2$. ■

2.1.11 Definice Jednoznačně určené číslo r z předchozí věty nazveme *zbytkem po dělení čísla a číslem b.*

Na platnosti následujícího tvrzení založíme konstrukci rekursivního algorimu pro hledání největšího společného dělitele dvojice přirozených čísel.

2.1.12 Tvrzení *Předpokládejme, že pro přirozená čísla a, b platí $a \geq b > 0$. Vydělme číslo a číslem b se zbytkem. Pro nějaká q a r tedy platí*

$$a = q \cdot b + r, \text{ kde } 0 \leq r < b.$$

1. *Je-li $r = 0$, potom b je největším společným dělitelem čísel a, b.*
2. *Je-li $r > 0$, označme jako d jakéhokoli společného dělitele původních čísel a a b. Potom d je společný dělitel čísel b a r.*

Důkaz.

1. Pokud $a = q \cdot b + 0$, potom b je zřejmě největší společný dělitel čísel a, b.
2. Předpokládejme, že d je společný dělitel čísel a a b. To znamená, že existují přirozená čísla q_a, q_b tak, že platí
$$a = q_a \cdot d \quad \text{a} \quad b = q_b \cdot d$$
Protože $a = q \cdot b + r$, platí tedy $q_a \cdot d = q \cdot q_b \cdot d + r$, neboli $q_a \cdot d - q \cdot q_b \cdot d = r$. Z levé strany rovnosti lze vytknout d, to znamená, že d je dělitelem čísla r.

■

Předpokládejme, že pro přirozená čísla a, b platí $a \geq b > 0$. *Eukleidův algoritmus* pro hledání největšího společného dělitele čísel a, b popíšeme takto:

Označme $b = b_0$ a dělením se zbytkem vytvořme posloupnost přirozených čísel b_1, b_2, ...:

$$\begin{aligned} a &= q_0 \cdot b_0 + b_1 \\ b_0 &= q_1 \cdot b_1 + b_2 \\ b_1 &= q_2 \cdot b_2 + b_3 \\ &\vdots \end{aligned}$$

Povšimněme si, že platí $b_1 > b_2 > b_3 > \ldots$. Jde totiž o zbytky při dělení. Proto existuje n takové, že $b_n = 0$. Tvorbu posloupnosti $b_1, b_2, \ldots$ v tomto okamžiku zastavíme. Potom číslo b_{n-1} je největší společný dělitel čísel a, b — to plyne z tvrzení 2.1.12. Všimněte si, že $b_1, b_2, \ldots$ tvoří variant Eukleidova algoritmu ve smyslu odstavce 1.2.

2.1.13 Příklad Ukažme příklad běhu Eukleidova algoritmu pro $a = 427$, $b = 133$.

$$\begin{aligned} 427 &= 3 \cdot 133 + 28 \\ 133 &= 4 \cdot 28 + 21 \\ 28 &= 1 \cdot 21 + 7 \\ 21 &= 3 \cdot 7 + 0 \end{aligned}$$

Proto platí, že $\gcd(427, 133) = 7$.

Důsledkem Eukleidova algoritmu je následující tvrzení, které budeme v dalším potřebovat.

2.1.14 Důsledek (Bezoutova rovnost)
Ať a a b jsou přirozená čísla. Potom existují celá čísla α, β tak, že platí rovnost

$$\gcd(a, b) = \alpha \cdot a + \beta \cdot b.$$

Bezoutovu rovnost nebudeme dokazovat, ukážeme na příkladu, jak příslušné koeficienty α a β nalézt.

2.1.15 Příklad V příkladu 2.1.13 jsme pomocí Eukleidova algoritmu ukázali, že $\gcd(427, 133) = 7$. Rovnosti, které jsme při běhu Eukleidova algoritmu vytvořili, nyní využijeme k nalezení celých čísel α, β, pro která platí

$$7 = \alpha \cdot 427 + \beta \cdot 133$$

Protože číslo 7 (největší společný dělitel) se vyskytlo v rovnicích příkladu 2.1.13 jako zbytek (předposlední rovnost), můžeme psát

$$7 = 1 \cdot \underline{28} - \underline{21}$$

kde čísla 28 a 21 jsou opět zbytky po dělení (v první a druhé rovnici). Všechny zbytky v dalších výpočtech jsou podtržené, čísla 427 a 133 jsou v rámečku, o nepodtržených číslech budeme uvažovat jako o skalárech, s jejichž pomocí vytvoříme hledaná čísla α a β. Postupným využíváním rovnic z příkladu 2.1.13 tak dostaneme

$$\begin{aligned} 7 &= 1 \cdot \underline{28} - \underline{21} \\ &= 1 \cdot (\boxed{427} - 3 \cdot \boxed{133}) - (\boxed{133} - 4 \cdot \underline{28}) \\ &= 1 \cdot (\boxed{427} - 3 \cdot \boxed{133}) - (\boxed{133} - 4 \cdot (\boxed{427} - 3 \cdot \boxed{133})) \\ &= 5 \cdot \boxed{427} - 16 \cdot \boxed{133} \end{aligned}$$

Hledaná čísla jsou: $\alpha = 5$, $\beta = -16$.

Se zvyšujícím se počtem řádků v Eukleidově algoritmu se výše uvedený postup se stává zdlouhavým a těžkopádným. Proto zformulujeme:

2.1.16 Rozšířený Eukleidův algoritmus
Vstup: přirozená čísla a, b, kde $a \geq b \geq 0$.
Výstup: $d = \gcd(a, b)$ a celá čísla α, β, splňující $d = \alpha \cdot a + \beta \cdot b$.

1. Je-li $b = 0$, položte $d := a$, $\alpha := 1$, $\beta := 0$ a skončete.
2. Položte $\alpha_2 := 1$, $\alpha_1 := 0$, $\beta_2 := 0$, $\beta_1 := 1$.
3. Dokud $b > 0$ dělejte následující:
   3.1 Spočtěte q a r tak, že $a = q \cdot b + r$, $0 \leq r < b$.

3.2 Položte $\alpha := \alpha_2 - q \cdot \alpha_1$, $\beta := \beta_2 - q \cdot \beta_1$.

3.3 Položte $a := b$, $b := r$.

3.4 Položte $\alpha_2 := \alpha_1$, $\alpha_1 := \alpha$, $\beta_2 := \beta_1$, $\beta_1 := \beta$.

4. Položte $d := a$, $\alpha := \alpha_2$, $\beta := \beta_2$ a skončete.

V krocích 3.1 a 3.3 jste určitě rozpoznali klasický Eukleidův algoritmus. Zbývá vysvětlit kroky 3.2 a 3.4. Necháme si to na později a spustíme rozšířený Eukleidův algoritmus na nalezení $\gcd(427, 133)$. Naše výpočty uspořádáme do tabulky:

a	b	q	r	α	β	α_2	α_1	β_2	β_1
427	133					1	0	0	1
427	133	3	28	1	-3	0	1	1	-3
133	28	4	21	-4	13	1	-4	-3	13
28	21	1	7	5	-16	-4	5	13	-16
21	7	3	0	-19	61	5	-19	-16	61
7	0			5	-16				

Proto tvrdíme, že $\gcd(427, 133) = 7$ a že Bezoutova rovnost má tvar $7 = 5 \cdot 427 + (-16) \cdot 133$.

2.1.17 Poznámka Naznačíme, proč je rozšířený Eukleidův algoritmus korektní. Půjde o ukázku užitečné techniky, která se pro rekursivní algoritmy často používá, o *invariant*. (Připomeňme, že v odstavci 1.2 jsme využili pojmu *variant* k důkazu terminace rekursivního algoritmu.) Pojem invariantu se využívá k důkazu (parciální) korektnosti rekursivního algoritmu. Zhruba řečeno, invariant je formule, která platí na začátku a zůstává pravdivou „při průchodu algoritmem“. (Srovnejte s invariancí vlastnosti V v principu strukturální indukce ve větě 1.4.5.) Proto invariant platí po skončení rekurse a proto vhodný invariant nám říká, že algoritmus počítá to, co chceme. Je zapotřebí poznamenat, že invariant není v žádném případě určen jednoznačně (např. $1+1 = 2$ je invariantem vždy) a že hledání dobrého invariantu není algoritmicky řešitelná úloha.

Invariant rozšířeného Eukleidova algoritmu vypadá takto:

1. Původní hodnoty a a b uložíme do A a B, neboli algoritmus 2.1.16 obohatíme o řádek

   0. Položte $A := a$, $B := b$.

2. Pokud se vykoná řádek 1., pak je d vskutku největší společný dělitel a a b. Protože algoritmus končí, je korektnost v tomto případě dokázána.

3. Pokud se řádek 1. nevykoná, povšimněte si, že po vykonání řádku 2. platí rovnosti

$$b = \alpha_1 \cdot A + \beta_1 \cdot B \quad \text{a} \quad a = \alpha_2 \cdot A + \beta_2 \cdot B$$

   Tato formule je hledaným invariantem a my právě dokázali, že platí na začátku rekurse.

4. Abychom ukázali, že výše uvedené rovnosti tvoří invariant, předpokládejme, že platí $b = \alpha_1 \cdot A + \beta_1 \cdot B$ a $a = \alpha_2 \cdot A + \beta_2 \cdot B$ a že se má vykonat tělo cyklu while, tj., že platí $b > 0$. Musíme ukázat, že platí

$$r = (\alpha_2 - q \cdot \alpha_1) \cdot A + (\beta_2 - q \cdot \beta_1) \cdot B \quad \text{a} \quad b = \alpha_1 \cdot A + \beta_1 \cdot B$$

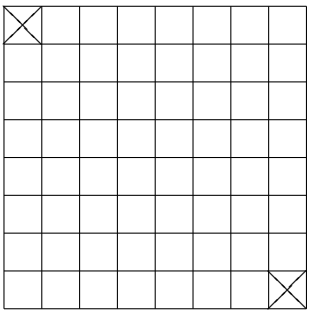
   (což je invariant po průchodu cyklem). Druhá rovnost je triviální a pro důkaz první rovnosti spočtěme

$$\begin{aligned} r &= a - q \cdot b = (\alpha_2 \cdot A + \beta_2 \cdot B) - q \cdot (\alpha_1 \cdot A + \beta_1 \cdot B) = \\ &= (\alpha_2 - q \cdot \alpha_1) \cdot A + (\beta_2 - q \cdot \beta_1) \cdot B \end{aligned}$$

5. Proto invariant platí po skončení cyklu while. To nastane, když $b = 0$ a protože platí $a = \alpha_2 \cdot A + \beta_2 \cdot B$, dává řádek 4. hledanou rovnost $d = \alpha \cdot A + \beta \cdot B$.

2.1.18 Příklad V tomto příkladu ukážeme *použití invariantu* na vyřešení další úlohy o *parketáži* (viz také cvičení 1.6.20).

Místností rozměru n, kde $n \geq 1$, budeme rozumět šachovnici rozměru $2n \times 2n$, ze které jsou vyjmuty *protilehlá* rohová pole. Toto je příklad místnosti rozměru 4:



Domino je parketa následujícího tvaru:

Ukážeme tvrzení:

Žádnou místnost rozměru $n \geq 1$ nelze *vyparketovat dominy.*

Budeme postupovat následovně:

1. Jednotlivá pole místnosti rozměru n obarvíme střídavě černě a bíle tak, jak je tomu na běžné šachovnici.

   Povšimněme si, že díky předpokladům musí mít vyjmutá protilehlá rohová pole *stejnou* barvu. Bez újmy na obecnosti budeme předpokládat, že *oba vyjmuté rohy jsou černé*.

   Proto pro naši místnost platí rovnost
   $$\#B + 2 = \#W \tag{2.1}$$
   kde $\#B$ značí počet černých polí a $\#W$ značí počet bílých polí v místnosti.

2. Jakýkoli parketovací algoritmus musí pokládat (podle nějaké strategie S) domina tak dlouho, dokud není místnost vyparketována.

   Pokud před položením jednoho domina pro dosud nevyparketovanou část místnosti platila rovnost (2.1), bude po položení jednoho domina rovnost (2.1) pro dosud nevyparketovanou část místnosti platit opět. Tato vlastnost zjevně na strategii S vůbec nezávisí.

3. Předpokládejme, že existuje terminující parketovací algoritmus. Podle předchozího musí po ukončení algoritmu platit rovnost (2.1) pro *prázdnou* místnost. V předchozím bodě jsme totiž ukázali, že rovnost (2.1) je *invariantem*.

   To je ovšem spor: prázdná místnost má 0 černých a 0 bílých polí a rovnost (2.1) pro ni neplatí. Úspěšný parketovací algoritmus tudíž nemůže existovat.

2.1.19 Poznámka Algoritmus 2.1.16 nám dodal *konstruktivní* způsob nalezení největšího společného dělitele. Zajímavé použití principu dobrého uspořádání nám dovolí dokázat existenci $\gcd(a, b)$ *nekonstruktivně*. Důkaz probíhá následovně (a, b jsou kladná přirozená čísla):

Definujte $S = \{n \cdot a + m \cdot b \mid m \text{ a } n \text{ jsou celá čísla}\}$. Množina S je neprázdná a obsahuje alespoň jedno kladné přirozené číslo. (Celá množina S samozřejmě *není* podmnožinou $\mathbb{N}$.)

Podle principu dobrého uspořádání má množina $S \cap \{n \mid n > 0\}$ nejmenší prvek, řekněme d. Pak, pro nějaká celá čísla α, β, platí rovnost $d = \alpha \cdot a + \beta \cdot b$. Navíc je $d \leq a$ (protože platí $a = 1 \cdot a + 0 \cdot b$) a $d \leq b$ (protože $b = 0 \cdot a + 1 \cdot b$). Podle věty o dělení se zbytkem můžeme psát:

$$\begin{aligned} a &= d \cdot q_a + r_a,\ 0 \leq r_a < d \\ b &= d \cdot q_b + r_b,\ 0 \leq r_b < d \end{aligned}$$

Pak platí rovnost $r_a = (1 - \alpha \cdot q_a) \cdot a + (-\beta \cdot q_a) \cdot b$ a tudíž je $r_a = 0$, protože d je nejmenší kladné přirozené číslo v množině S. Proto platí $d \mid a$. Podobné úvahy nás dovedou k tomu, že platí $d \mid b$. Ukázali jsme, že d je společným dělitelem čísel a a b. Samozřejmě, pokud je c jiný společný dělitel čísel a a b, pak platí $c \mid d$ (využijte rovnost $d = \alpha \cdot a + \beta \cdot b$). Tudíž jsme ukázali, že $d = \gcd(a, b)$.

2.1.20 Poznámka Úvahy předchozího důkazu byly typickou ukázkou *nekonstruktivního* existenčního důkazu v matematice. Po skončení takového důkazu jsme přesvědčeni o existenci hledaného objektu nade vší pochybnost, není však zadán žádný způsob, jak onen objekt sestrojit. Krásnou ukázkou nekonstruktivního důkazu je důkaz tvrzení

Existují dvě kladná iracionální čísla a, b taková, že mocnina a^b je racionální číslo.

Při důkazu využijeme faktu, že nastane právě jeden ze dvou případů:

1. Mocnina $\sqrt{2}^{\sqrt{2}}$ je racionální číslo. V tomto případě lze důkaz ukončit, položte $a = \sqrt{2}$, $b = \sqrt{2}$. Pak a^b je racionální číslo.

2. Mocnina $\sqrt{2}^{\sqrt{2}}$ je iracionální číslo. V tomto případě položte $a = \sqrt{2}^{\sqrt{2}}$, $b = \sqrt{2}$. Pak $a^b = 2$.

Vtip je v tom, že nepotřebujeme vědět, která z alternativ nastává. V každém z obou případů totiž umíme důkaz úspěšně dokončit.

Víme, že Eukleidův algortimus vždy skončí. Můžeme se však zeptat, *jak dlouho* výpočet $\gcd(a, b)$ trvá. Odpověď souvisí s Fibonacciho[4] posloupností $\{F(n)\}_{n=1}^{\infty}$, zavedenou ve cvičení 1.6.22.

2.1.21 Tvrzení *Ať $a > b$ a ať n je počet cifer čísla b v decimálním rozvoji. Potom Eukleidův algoritmus při hledání* $\gcd(a, b)$ *vykoná méně než $5n$ dělení.*

Důkaz. Označme jako k počet po sobě jdoucích dělení v Eukleidově algoritmu, spuštěném na a a b. Chceme dokázat nerovnost $k < 5n$.

Jako $b = b_0 > b_1 > b_2 > \ldots b_k \geq 1$ označíme posloupnost zbytků po dělení, tj. běh Eukleidova algoritmu má následující tvar:

$$\begin{aligned} a &= q_0 \cdot b_0 + b_1 \\ b_0 &= q_1 \cdot b_1 + b_2 \\ b_1 &= q_2 \cdot b_2 + b_3 \\ &\vdots \\ b_{k-2} &= q_{k-1} \cdot b_{k-1} + b_k \\ b_{k-1} &= q_k \cdot b_k + 0 \end{aligned}$$

Dokážeme nerovnost

$$b_{k-i} \geq F(i+2), \quad 0 \leq i \leq k \tag{2.2}$$

kde $F(i+2)$ je $(i+2)$-té Fibonacciho číslo.

1. Pro $i = 0$ platí nerovnost $b_k \geq F(2)$, protože $F(2) = 1$ a $b_k \geq 1$.

2. Pro $i = 1$ platí nerovnost $b_{k-1} \geq F(3)$, protože $F(3) = 2$ a $b_{k-1} > b_k \geq 1$.

[4] Leonardo Pisano (1170–1250), známější pod předzdívkou Fibonacci, je mimo jiné autorem díla *Liber Abaci*. Opisováním této knihy se po Evropě rozšířil poziční decimální zápis arabskými číslicemi. Arabské číslice jsou ovšem původem indické, ale do povědomí se dostaly díky arabským pracem. Viz také stranu 70. Stojí za zmínku, že v obchodním styku bylo používání dekadického zápisu kvůli možným podvodům zakázáno a až do 16. století se proto v Evropě používal zápis římskými číslicemi. Zápis římskými číslicemi je totiž na možné připsání číslice odolnější než poziční zápis.

3. Pro $i \geq 2$ se podívejme na rovnost

$$b_{k-i} = q_{k-i+1} \cdot b_{k-i+1} + b_{k-i+2}$$

kterou nám dodá spuštění Eukleidova algoritmu. Pak platí

$$b_{k-i} = q_{k-i+1} \cdot b_{k-i+1} + b_{k-i+2} \geq b_{k-i+1} + b_{k-i+2} \geq F(i) + F(i+1) = F(i+2)$$

(použili jsme zřejmé nerovnosti $q_{k-i+1} \geq 1$).

S využitím nerovnosti (2.2) dostáváme

$$b = b_0 \geq F(k+2)$$

Podle explicitní formule pro Fibonacciho čísla (1.17) víme, že platí

$$F(k+2) = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{k+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+2}\right)$$

a proto platí i

$$b \geq \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{k+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+2}\right) > \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^{k+2} - 1\right) \tag{2.3}$$

protože platí $\left|\frac{1-\sqrt{5}}{2}\right|^{k+2} < 1$. Přepsáním (2.3) dostaneme následující nerovnosti

$$\left(\frac{1+\sqrt{5}}{2}\right)^{k+2} < b\sqrt{5} + 1 < \sqrt{5}(b+1) < \left(\frac{1+\sqrt{5}}{2}\right)^{2}(b+1) \tag{2.4}$$

a, po vydělení obou stran, nerovnost

$$\left(\frac{1+\sqrt{5}}{2}\right)^{k} < b+1 \tag{2.5}$$

Protože platí $10 < (\frac{1+\sqrt{5}}{2})^5$, platí podle (2.5) nerovnost

$$10^k < \left(\frac{1+\sqrt{5}}{2}\right)^{5k} < (b+1)^5 \leq 10^{5n} \tag{2.6}$$

(zde používáme nerovnost $b < 10^n$, která plyne z toho, že b má n cifer v decimálním rozvoji, a proto platí $b+1 \leq 10^n$).

Nerovnost $k < 5n$ plyne okamžitě z (2.6). ■

Pro další analýzu Eukleidova algoritmu viz cvičení 2.4.9 a pro jiný důkaz tvrzení 2.1.21 viz cvičení 2.4.10.

2.2 Přerušení výkladu — binární relace a relace ekvivalence

V tomto odstavci přerušíme tok výkladu: v odstavci 2.3 budeme totiž chtít „slepovat“ celá čísla a vybudovat tak teorii počítání modulo. Než vysvětlíme, co přesně slepováním myslíme, zavedeme pojem *binární relace na množině*.

2.2.1 Definice *Binární relace R na množině A* je podmnožina $R \subseteq A \times A$. Namísto $(x,y) \in R$ budeme psát $x\ R\ y$ a budeme říkat, že *x je v relaci R s y*.

Zadat binární relaci na množině A tedy znamená zadat seznam R uspořádaných dvojic prvků množiny A. Na onom seznamu R se vyskytují přesně ty dvojice (x,y), pro které platí $x\ R\ y$.

2.2.2 Příklad Ať $A = \{a,b,c\}$. Příklady binárních relací na A jsou:

1. $R = \{(a,b),(c,a)\}$. Platí $a\ R\ b$ a také $c\ R\ a$.

2. $\Delta_A = \{(a,a),(b,b),(c,c)\}$. Všimněme si, že platí $x\ \Delta_A\ y$ právě, když $x = y$. Této relaci budeme říkat *diagonála na A* nebo *identita na A*.

3. $A \times A$ je binární relace na A. Je to „největší“ možná binární relace na množině A. Platí $x\ (A \times A)\ y$ právě tehdy, když $x, y \in A$.

4. $\emptyset$ je binární relace na A. Je to „nejmenší“ možná binární relace na množině A. Pro žádnou dvojici (x,y) neplatí $x\ \emptyset\ y$.

Binární relaci R na množině A je vskutku užitečné si představovat jako seznam dvojic, ve kterých má první složka nějaký vztah ke druhé. V dalším budeme chtít takovou relaci R chápat dvěma různými způsoby:

1. Jako seznam dvojic (x,y), kdy se x má „slepit“ s y. Takovým relacím R se říká *relace ekvivalence*. Samozřejmě, že ne každou relaci můžeme chápat jako návod ke slepování: relace ekvivalence musí mít speciální vlastnosti, viz definici 2.2.3.

2. Jako seznam dvojic (x,y), kdy x je „menší nebo rovno“ y. Takovým relacím R se říká *relace uspořádání*. Samozřejmě, že ne každou relaci můžeme chápat jako uspořádání: relace uspořádání musí mít speciální vlastnosti, viz definici 2.2.3.

Speciální vlastnosti binárních relací, které nás budou zajímat, jsou následující:

2.2.3 Definice Řekneme, že binární relace R na množině A je:

1. *Reflexivní*, když pro všechna $x \in A$ platí: $x\ R\ x$.

2. *Symetrická*, když pro všechna $x, y \in A$ platí: jestliže $x\ R\ y$, pak $y\ R\ x$.

3. *Transitivní*, když pro všechna $x, y, z \in A$ platí: jestliže $x\ R\ y$ a současně $y\ R\ z$, pak $x\ R\ z$.

4. *Antisymetrická*, když pro všechna $x, y \in A$ platí: jestliže $x\ R\ y$ a současně $y\ R\ x$, pak $x = y$.

5. *Relace ekvivalence*, pokud je reflexivní, symetrická a transitivní současně.

6. *Relace uspořádání*, pokud je reflexivní, antisymetrická a transitivní současně.

2.2.4 Poznámka Jedné speciální relaci uspořádání (sice dělitelnosti na množině přirozených čísel) je věnováno cvičení 2.4.1. Uspořádáním se budeme dále věnovat v kapitole 6 (speciálně v odstavci 6.1). Přesto nyní budeme definici relace uspořádání analyzovat podrobněji: zadanou binární relaci uspořádání na množině A budeme pro větší názornost v této poznámce značit $\sqsubseteq$. To je ostatně značení v aplikacích zcela běžné. Množina $\sqsubseteq$ je tedy množina uspořádaných dvojic prvků z A, která je navíc reflexivní, antisymetrická a transitivní. Podívejme se, co jednotlivé požadavky znamenají:

1. Reflexivita relace $\sqsubseteq$ znamená, že pro všechna $x \in A$ platí $x \sqsubseteq x$. Neboli: každý prvek $x \in A$ je „menší nebo roven“ sám sobě.

2. Antisymetrie relace $\sqsubseteq$ znamená, že pro všechna $x, y \in A$ platí: jestliže $x \sqsubseteq y$ a současně $y \sqsubseteq x$, pak $x = y$. Neboli: jestliže x je „menší nebo roven“ y a současně y je „menší nebo roven“ x, pak $x = y$.

3. Transitivita relace $\sqsubseteq$ znamená, že pro všechna $x, y, z \in A$ platí: jestliže $x \sqsubseteq y$ a současně $y \sqsubseteq z$, pak $x \sqsubseteq z$. Neboli: jestliže x je „menší nebo roven“ y a současně y je „menší nebo roven“ z, pak x je „menší nebo roven“ z.

Je vidět, že tři vlastnosti relace uspořádání jsou přirozenými požadavky na vztah „být menší nebo roven“. Více v odstavci 6.1.

2.2.5 Poznámka Ukážeme, že slepování prvků množiny X se děje obecně pomocí relace ekvivalence (tj. reflexivní, symetrické a transitivní binární relace na množině X, viz definice 2.2.3). Analyzujme opět tyto tři vlastnosti a čtěme přitom $x\ R\ y$ jako x se má „slepit“ s y.

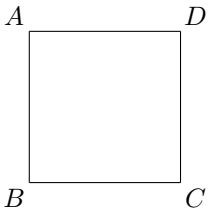
1. Reflexivita relace R znamená, že pro všechna $x \in A$ platí $x \; R \; x$. Neboli: každý prvek $x \in A$ se má „slepit“ sám se sebou.

2. Symetrie relace R znamená, že pro všechna $x, y \in A$ platí: jestliže $x \; R \; y$, pak $y \; R \; x$. Neboli: jestliže x se má „slepit“ s y, pak se y má „slepit“ s x.

3. Transitivita relace R znamená, že pro všechna $x, y, z \in A$ platí: jestliže $x \; R \; y$ a současně $y \; R \; z$, pak $x \; R \; z$. Neboli: jestliže x se má „slepit“ s y a současně y se má „slepit“ se z, pak se x má „slepit“ se z.

Předvedeme to na několika geometrických příkladech.

Vezměme si následující čtverec $\mathbb{S}$ v rovině:

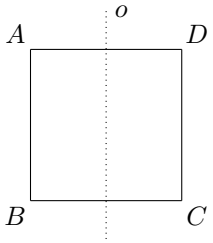


1. Zaveďme na $\mathbb{S}$ binární relaci R takto: dva body P_1, P_2 čtverce $\mathbb{S}$ jsou v relaci R právě tehdy, když platí buď $P_1 = P_2$ nebo P_1 i P_2 leží na hranici.

   Je snadné ukázat, že R je relace ekvivalence na množině $\mathbb{S}$ (ukažte to). Relace R je tedy návod, jak slepit body čtverce: slepte všechny body hranice do jednoho a uvnitř čtverce neslepujte nic (tj. slepte pouze bod sám se sebou). Výsledná *faktorová množina* $\mathbb{S}/R$ (neboli množina $\mathbb{S}$ slepená podle návodu R) je evidentně povrch koule, neboli *sféra*.

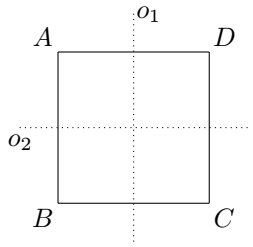
   To je možné si uvědomit „procházkou“ po $\mathbb{S}$ s optikou danou relací R: chodíme-li uvnitř čtverce, potkáváme různé body, dorazíme-li na hranici, pak se smíme vynořit z jakéhokoli dalšího bodu hranice. Na obvodu čtverce $\mathbb{S}$ totiž „vidíme špatně“. Taková procházka je samozřejmě přesně procházkou po povrchu koule.

2. Zavedeme-li na $\mathbb{S}$ binární relaci R tak, že chceme slepit pouze body na úsečce AB s odpovídajícími body na úsečce CD podle symetrie dané osou o



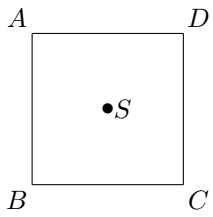
   dostaneme *válcovou plochu*. (Popište příslušnou ekvivalenci přesně a popište procházku na čtverci.)

3. Zavedeme-li na $\mathbb{S}$ binární relaci R tak, že chceme slepit pouze body na úsečce AB s odpovídajícími body na úsečce CD podle symetrie dané osou o_1 a body na úsečce BC s odpovídajícími body na úsečce DA podle symetrie dané osou o_2 (tj. děláme válcovou plochu na válcové ploše)



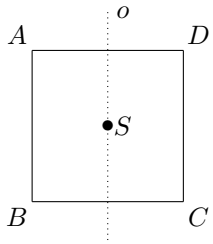
   dostaneme *povrch toru*. (Popište příslušnou ekvivalenci přesně a popište procházku na čtverci.)

4. Zavedeme-li na $\mathbb{S}$ binární relaci R tak, že chceme slepit pouze body na úsečce AB s odpovídajícími body na úsečce CD podle symetrie dané bodem S



dostaneme *Möbiův list*. (Popište příslušnou ekvivalenci přesně a popište procházku na čtverci.)

5. Zavedeme-li na $\mathbb{S}$ binární relaci R tak, že chceme slepit pouze body na úsečce AB s odpovídajícími body na úsečce CD podle symetrie dané osou o a body úsečky BC s odpovídajícími body úsečky DA podle symetrie dané bodem S (tj. děláme Möbiův list na válcové ploše)



dostaneme *Kleinovu láhev*. (Popište příslušnou ekvivalenci přesně a popište procházku na čtverci.[5])

Obecně tedy můžeme říci následující:

> Pokud je zadána množina X a relace ekvivalence R na X, je rozumné na tuto situaci pohlížet následovně: relace R je *návod*, jak se mají slepovat prvky množiny X.
>
> Po dvojici $\langle X, R\rangle$ se můžeme „procházet“, přičemž relace R nám „pokazila zrak“: některé dvojice bodů začneme považovat za bod jediný. To vede k následujícímu pojmu *faktorové množiny*: faktorová množina X/R je množina, kde jsme předepsané dvojice bodů *skutečně slepili*.
>
> Jak vypadají body (prvky) množiny X/R? Slepme každý bod $x \in X$ se všemi body x', se kterými nám přikazuje relace R bod x slepit. To znamená, že vytvoříme množiny
>
> $$[x]_R = \{x' \in X \mid x\ R\ x'\}$$
>
> Každé množině $[x]_R$ se říká *třída ekvivalence R representovaná prvkem x*. Proto platí
>
> $$X/R = \{[x]_R \mid x \in X\}$$

Později budeme slepovat prvky složitějších věcí, než jsou množiny, viz větu 5.5.7 a poznámku 5.5.6.

Poznamenejme ještě, že vlastnosti z definice 2.2.3 se dají vyjádřit i jinak. K tomu ovšem musíme zavést pojmy *skládání relací* a *opačné relace*.

2.2.6 Definice Ať R a S jsou binární relace na množině A.

1. *Opačná relace* k relaci R je binární relace značená R^{op} s vlastností $x\ R^{op}\ y$ právě tehdy, když $y\ R\ x$.
2. *Složení relací R a S* je binární relace značená $R;S$ s vlastností $x\ R;S\ y$ právě tehdy, když existuje $z \in A$ takové, že $x\ R\ z$ a současně $z\ S\ y$. Prvku z říkáme *prostředník* vztahu $x\ R;S\ y$.

2.2.7 Poznámka Poznamenejme, že značení pro opačnou relaci a pro skládání relací se *liší* od značení ve skriptu

[5] Kleinova láhev není třídimensionální objekt, přesto se po něm můžete procházet — její povrch je totiž dvoudimensionální!

☞ M. Demlová a B. Pondělíček, *Matematická logika*, skriptum FEL ČVUT, Praha, 1997

V tomto skriptu se místo R^{op} píše R^{-1} a místo opačná relace se říká relace inverzní. Dále se místo našeho značení $R;S$ pro složení relací používá $R \circ S$. Značení a terminologie, které jsme použili my, jsou obvyklé v computer science.

2.2.8 Příklad Ať $A = \{a, b, c\}$. Pro příklady binárních relací na A z příkladu 2.2.2 máme:

1. $R = \{(a, b), (c, a)\}$. Platí $R^{op} = \{(b, a), (a, c)\}$. Relace R^{op} je totiž seznam R, kde jsme prohodili pořadí prvních a druhých složek.

   Platí $R;R = \{(c, b)\}$. To je totiž jediná dvojice, která má komplikovaný vztah $R;R$ (zprostředkován prostřednictvím prvku a): platí $c\ R\ a$ a současně $a\ R\ b$. Proto platí $c\ R;R\ b$.

2. $\Delta_A = \{(a, a), (b, b), (c, c)\}$. Potom platí $\Delta_A^{op} = \Delta_A$ a $\Delta_A;\Delta_A = \Delta_A$.

2.2.9 Poznámka Prostředníků zprostředkujících vztah složených relací může být více než jeden. Ať $A = \{a, b, c\}$, $R = \{(a, a), (a, b), (b, c)\}$, $S = \{(c, a), (a, a), (b, a)\}$. Potom platí $R;S = \{(a, a), (b, a)\}$. Dvojice (a, a) je zprostředkována prvkem a (nebo prvkem b), dvojice (b, a) je zprostředkována prvkem c.

Nyní vyslovíme alternativní kritéria pro reflexivitu, symetrii, transitivitu a antisymetrii binární relace. Připomeňme, že Δ_A značí diagonální relaci na množině A (tj. $x\ \Delta_A\ y$ platí, právě když platí $x = y$).

2.2.10 Tvrzení *Ať R je binární relace na množině A. Pak platí:*

1. *Relace R je reflexivní právě tehdy, když platí $\Delta_A \subseteq R$.*
2. *Relace R je symetrická právě tehdy, když platí $R = R^{op}$.*
3. *Relace R je transitivní právě tehdy, když platí $R;R \subseteq R$.*
4. *Relace R je antisymetrická právě tehdy, když platí $R \cap R^{op} \subseteq \Delta_A$.*

Důkaz.

1. Seznam Δ_A obsahuje pouze dvojice (x, x) pro $x \in A$. Inkluse $\Delta_A \subseteq R$ říká, že každou dvojici ze seznamu Δ_A nalezneme na seznamu R. To je ale přesně reflexivita relace R.
2. Rovnost $R = R^{op}$ říká, že seznamy dvojic R a R^{op} jsou shodné. To je přesně totéž jako říci, že v seznamu R nezáleží na pořadí položek: relace R je symetrická.
3. Inkluse $R;R \subseteq R$ říká, že každou dvojici (x, y) ze seznamu $R;R$ nalezneme na seznamu R. Dvojice (x, y) je na seznamu $R;R$ přesně tehdy, když existuje prostředník z s vlastností $x\ R\ z$ a současně $z\ R\ y$. Neboli inkluse $R;R \subseteq R$ říká, že relace R je transitivní.
4. Inkluse $R \cap R^{op} \subseteq \Delta_A$ znamená, že pro každou dvojici (x, y) ze seznamu $R \cap R^{op}$ platí $x = y$. Na seznamu $R \cap R^{op}$ jsou ovšem přesně ty dvojice (x, y), pro které platí $x\ R\ y$ a současně $y\ R\ x$. Proto inkluse $R \cap R^{op} \subseteq \Delta_A$ znamená, že relace R je antisymetrická.

■

2.2.11 Příklad Ať A je jakákoli množina. Ukážeme, že relace $A \times A$ je vždy relace ekvivalence.
Označme $A \times A$ jako R. Použijeme tvrzení 2.2.10:

1. Jistě platí $\Delta_A \subseteq R$, proto je relace R reflexivní.
2. Platí i rovnost $R = R^{op}$, proto je relace R symetrická.
3. Platí $R;R \subseteq R$, proto je relace R transitivní.

Relace R je tedy relace ekvivalence.

Ať je množina A navíc neprázdná. Spočítáme třídu ekvivalence R representovanou nějakým prvkem $a \in A$:

$$[a]_R = \{a' \in A \mid a' \; R \; a\} = A \times A$$

S prvkem a se tedy mají slepit všechny prvky množiny A. Příslušná faktorová množina A/R má tedy *jediný* prvek — všechny prvky množiny A se slepí v jediný.

2.2.12 Příklad Tvrzení 2.2.10 nám umožní zjistit, že existují relace, které jsou *symetrické* a *antisymetrické* současně. Příkladem je diagonála Δ_A na množině A:

1. Relace Δ_A je jistě symetrická, protože platí $\Delta_A = \Delta_A^{op}$.
2. Relace Δ_A je antisymetrická. Protože $\Delta_A = \Delta_A^{op}$, platí $\Delta_A \cap \Delta_A^{op} = \Delta_A$. Tím spíš platí inkluse $\Delta_A \cap \Delta_A^{op} \subseteq \Delta_A$.

Více cvičení 2.4.11 a 2.4.12.

2.3 Kongruence modulo m

2.3.1 Definice Ať $m > 1$ je pevné přirozené číslo. Řekneme, že *celá čísla a a b jsou kongruentní modulo m*, (značíme $a \equiv b \pmod m$), pokud existuje celé číslo k takové, že $a - b = k \cdot m$.

Povšimněme si, že vztah $a \equiv b \pmod m$ platí právě tehdy, když čísla a a b mají stejný zbytek po dělení číslem m.

2.3.2 Příklad V příkladu 2.1.15 jsme ukázali, že $7 = 5 \cdot 427 - 16 \cdot 133$. Platí tedy

$$7 \equiv -16 \cdot 133 \pmod{427}$$

V dalším budeme chtít tento fakt vyjádřit tak, že čísla 7 a $-16 \cdot 133$ se „chovají stejně“, pokud na ně pohlížíme jako na zbytky po dělení číslem 427.

Čísla kongruentní modulo m budeme chtít ztotožnit. K tomu musíme ukázat, že kongruence modulo m se „chová podobně jako rovnost“. Tento fakt je zformulován přesně v prvních třech podmínkách následujícího tvrzení, která (dohromady) říkají, že kongruence modulo m je *relace ekvivalence*. Poslední dvě podmínky pak říkají, že zbyky po dělení m „smíme sčítat a násobit“, což budeme potřebovat pro vybudování aritmetiky zbytků po dělení.

2.3.3 Tvrzení *Ať $m > 1$ je pevné přirozené číslo. Potom platí:*

1. *Kongruence modulo m je* reflexivní relace, *tj. pro každé celé číslo a platí $a \equiv a \pmod m$.*
2. *Kongruence modulo m je* symetrická relace, *tj. pro všechna celá čísla a, b platí:*
   jestliže platí $a \equiv b \pmod m$, pak platí $b \equiv a \pmod m$.
3. *Kongruence modulo m je* transitivní relace, *tj. pro všechna celá čísla a, b, c platí:*
   jestliže platí $a \equiv b \pmod m$ a současně $b \equiv c \pmod m$, pak platí $a \equiv c \pmod m$.
4. *Kongruence modulo m* respektuje operaci sčítání, *tj. pro všechna celá čísla a, b, a', b' platí:*
   jestliže platí $a \equiv b \pmod m$ a současně $a' \equiv b' \pmod m$, pak platí $a + a' \equiv b + b' \pmod m$.
5. *Kongruence modulo m* respektuje operaci násobení, *tj. pro všechna celá čísla a, b, a', b' platí:*
   jestliže platí $a \equiv b \pmod m$ a současně $a' \equiv b' \pmod m$, pak platí $a \cdot a' \equiv b \cdot b' \pmod m$.

Důkaz. Důkazy všech tvrzení jsou velmi jednoduché. Přesto je uvedeme.

1. Zřejmě $a - a = 0 \cdot m$ pro libovolné celé číslo a.

2. Ať platí $a \equiv b \pmod m$. Existuje tedy $k \in \mathbb{Z}$ tak, že $a - b = k \cdot m$. Potom $b - a = (-k) \cdot m$, a tedy $b \equiv a \pmod m$ (uvědomme si, že $-k$ je celé číslo).

3. Ať platí $a \equiv b \pmod m$ a současně $b \equiv c \pmod m$. Existují tedy $k \in \mathbb{Z}$ a $l \in \mathbb{Z}$ tak, že $a - b = k \cdot m$ a $b - c = l \cdot m$. Potom $a - c = (a - b) + (b - c) = (k + l) \cdot m$. Protože $k + l$ je celé číslo, platí $a \equiv c \pmod m$.

4. Ať platí $a \equiv b \pmod m$ a současně $a' \equiv b' \pmod m$. Existují tedy $k \in \mathbb{Z}$ a $k' \in \mathbb{Z}$ tak, že $a - b = k \cdot m$ a $a' - b' = k' \cdot m$. Potom $(a+a') - (b+b') = (k+k') \cdot m$. Protože $k+k'$ je celé číslo, platí $a + a' \equiv b + b' \pmod m$.

5. Ať platí $a \equiv b \pmod m$ a současně $a' \equiv b' \pmod m$. Existují tedy $k \in \mathbb{Z}$ a $k' \in \mathbb{Z}$ tak, že $a - b = k \cdot m$ a $a' - b' = k' \cdot m$. Potom $(a \cdot a') - (b \cdot b') = (a - b) \cdot a' + (a' - b') \cdot b = (a' \cdot k + b \cdot k') \cdot m$. Protože $a' \cdot k + b \cdot k'$ je celé číslo, platí $a \cdot a' \equiv b \cdot b' \pmod m$.

■

Připomeňme znovu, že první tři vlastnosti v tvrzení 2.3.3 ukazují, že relace „být kongruentní modulo m“ je relace ekvivalence na množině $\mathbb{Z}$. Tato ekvivalence nám dovoluje při nahlížení na množinu $\mathbb{Z}$ „změnit optiku“ ve smyslu poznámky 2.2.5 a chápat čísla kongruentní modulo m jako totožná. Zavedeme nyní označení pro třídy této ekvivalence.

2.3.4 Definice Ať $m > 1$ je pevné přirozené číslo. Pro libovolné celé číslo c definujeme $[c]_m$ jako množinu všech čísel kongruentních s c modulo m. Přesněji:

$$[c]_m = \{a \in \mathbb{Z} \mid c \equiv a \pmod m\}.$$

Množinu $[c]_m$ nazýváme *třídou kongruence čísla c modulo m*. Libovolný prvek množiny $[c]_m$ nazveme *representantem třídy* $[c]_m$.

Označme $\mathbb{Z}_m = \{[a]_m \mid a \in \mathbb{Z}\}$.

2.3.5 Příklad Pro $m = 6$ je například

$$[1]_6 = \{\ldots, -11, -5, 1, 7, 13, \ldots\}$$

Modulo 6 jsou tedy například čísla -11 a 7 „stejná“: dávají stejný zbytek po dělení číslem 6 a spadají tudíž do stejné třídy kongruence.

Fakt, že čísla -11 a 7 jsou „stejná“ modulo 6 můžeme vyjádřit i takto: všimněme si, že platí

$$[-11]_6 = \{\ldots, -11, -5, 1, 7, 13, \ldots\} = [7]_6$$

Obecně, pro libovolné $m > 1$, platí

$$[a]_m = [b]_m \quad \text{právě tehdy, když} \quad a \equiv b \pmod m$$

2.3.6 Příklad Připomeňme, že kongruence modulo m je relace ekvivalence, která respektuje sčítání a násobení (tvrzení 2.3.3). V tomto příkladu ukážeme jistou *minimální vlastnost* kongruence modulo m. Pro jednoduchost zvolíme $m = 6$, snadno ale zjistíte, že se důkaz dá provést pro obecné přirozené číslo $m > 1$.

Kongruenci modulo 6 jsme zavedli, abychom si vynutili platnost „rovnice“ $0 = 6$, neboli, abychom mohli slepit 0 a 6. Víme, že slepování prvků množin se děje pomocí relace ekvivalence. Pojďme se podívat, co všechno bychom požadovali od relace R, která slepuje 0 a 6 na množině $\mathbb{Z}$ a která se chová „dobře“ vzhledem ke sčítání a násobení.

1. Musí platit $0 \; R \; 6$, protože chceme slepit 0 a 6.

2. Relace R musí být relace ekvivalence, tj. R musí mít následující tři vlastnosti:

   (a) *reflexivita*: pro každé $a \in \mathbb{Z}$ platí $a \; R \; a$. Každý bod a slepíme s ním samým.

   (b) *symetrie*: pro každé $a, b \in \mathbb{Z}$ platí: jestliže $a \; R \; b$, pak $b \; R \; a$. Kdykoli slepíme a s b, pak slepíme i b s a.

(c) *transitivita*: pro každé $a, b, c \in \mathbb{Z}$ platí: jestliže $a \ R \ b$ a současně $b \ R \ c$, pak platí $a \ R \ c$. Kdykoli slepíme a s b a b s c, pak slepíme i a s c.

3. Relace R musí:

(a) *respektovat sčítání*: pro každé $a, b, a', b' \in \mathbb{Z}$ platí: jestliže $a \ R \ b$ a současně $a' \ R \ b'$, pak $a + a' \ R \ b + b'$. Neboli součty slepených bodů zustanou slepeny.

(b) *respektovat násobení*: pro každé $a, b, a', b' \in \mathbb{Z}$ platí: jestliže $a \ R \ b$ a současně $a' \ R \ b'$, pak $a \cdot a' \ R \ b \cdot b'$. Neboli součiny slepených bodů zustanou slepeny.

Samořejmě: víme, že relace kongruence modulo 6 má všechny výše uvedené vlastnosti. Ukážeme nyní, že jde o *nejmenší relaci* na $\mathbb{Z}$ s těmito vlastnostmi.

Zvolme tedy *jakoukoli* relaci R na $\mathbb{Z}$, která všechny výše uvedené vlastnosti má. Chceme ukázat tvrzení:

Pro každé $a, b \in \mathbb{Z}$ platí: jestliže platí $a \equiv b \pmod 6$, pak platí $a \ R \ b$.

Zvolme tedy libovolná a, b tak, že platí $a \equiv b \pmod 6$. To znamená, že existuje celé číslo k tak, že platí rovnost $a - b = k \cdot 6$. Mohou nastat dva případy: $k \geq 0$ a $k < 0$. Bez újmy na obecnosti budeme předpokládat, že nastává ten první. Kdyby totiž $k < 0$, pak můžeme rovnost $a - b = k \cdot 6$ přepsat na rovnost $b - a = (-k) \cdot 6$, kde platí $-k > 0$.

Máme tedy rovnost $a - b = k \cdot 6$, kde $k \geq 0$, a nyní indukcí podle k ukážeme, že platí $a \ R \ b$.

1. Základní krok: $k = 0$. Pak platí $a = b$ a my chceme ukázat, že platí $a \ R \ a$. To platí, protože R je reflexivní relace.

2. Indukční předpoklad zformulujeme takto: pro všechna a', b' plyne z rovnosti $a' - b' = k \cdot 6$ platnost $a' \ R \ b'$.

   Víme, že platí rovnost $a - b = (k + 1) \cdot 6$, neboli $(a - 6) - b = k \cdot 6$. Podle indukčního předpokladu platí $(a - 6) \ R \ b$. Protože platí $6 \ R \ 0$ (předpokládali jsme $0 \ R \ 6$, ale R je symetrická relace) a protože R respektuje sčítání, platí $((a - 6) + 6) \ R \ (b + 0)$. To ale znamená, že platí $a \ R \ b$, a to jsme chtěli dokázat.

Tvrzení nyní plyne podle slabého principu indukce.

Porovnejte předchozí výpočty s poznámkou 5.5.9: ukázali jsme, že relace $a \equiv b \pmod 6$ je *nejmenší kongruence* na okruhu $\mathbb{Z}$, která si vynucuje rovnost $0 = 6$.

2.3.7 Lemma *Ať $m > 1$ je pevné přirozené číslo. Potom množina $\mathbb{Z}_m$ má přesně m různých prvků.*

Důkaz. Ukážeme, že $\mathbb{Z}_m = \{[0]_m, [1]_m, \ldots, [m-1]_m\}$. Označme množinu $\{[0]_m, [1]_m, \ldots, [m-1]_m\}$ jako P. Množina P má zřejmě přesně m různých prvků.

Zřejmě platí inkluse $P \subseteq \mathbb{Z}_m$. Abychom ukázali, že $\mathbb{Z}_m \subseteq P$, zvolme libovolný prvek $[a]_m$ v $\mathbb{Z}_m$. Celé číslo a vydělíme číslem m se zbytkem, tj. najdeme celé číslo q a přirozené číslo r takové, že $a = q \cdot m + r$, kde $0 \leq r < m$. Potom zřejmě platí $a \equiv z \pmod m$, neboli $[a]_m = [r]_m$. Protože $0 \leq r < m$, ukázali jsme, že $[a]_m \in P$. ■

2.3.8 Definice Ať $m > 1$ je pevné přirozené číslo. Třídy kongruence $[0]_m$, $[1]_m$, …, $[m-1]_m$ nazveme *standardními tvary prvků $\mathbb{Z}_m$*.

2.3.9 Poznámka Kongruenci modulo m (definice 2.3.1) a $\mathbb{Z}_m$ (definice 2.3.4) jsme zavedli pro $m > 1$. Co by znamenala kongruence modulo 1? Vztah $a \equiv b \pmod 1$ znamená, že platí $1 \mid (a - b)$. Poslední relace ale platí vždy. Relace $- \equiv - \pmod 1$ je tedy relace $\mathbb{Z} \times \mathbb{Z}$, *každou* dvojici celých čísel najdeme na seznamu $- \equiv - \pmod 1$. Relace $- \equiv - \pmod 1$ je tedy zřejmě relace ekvivalence (viz příklad 2.2.11) a faktorová množina $\mathbb{Z}_1$ má tedy jediný prvek.

Rovnost $a = b$ v $\mathbb{Z}_1$ tedy znamená pouhé konstatování faktu, že jak a, tak b, jsou celá čísla.

2.3.10 Příklad Podle předchozích tvrzení má množina Z_6 přesně šest různých prvků. Standardní tvary těchto prvků jsou třídy ekvivalence $[0]_6$, $[1]_6$, $[2]_6$, $[3]_6$, $[4]_6$, $[5]_6$.

Ke konstrukci $\mathbb{Z}_6$ jsme zatím použili pouze faktu, že kongruence modulo 6 je relace ekvivalence. Relace „být kongruentní modulo 6“ má však i další dvě vlastnosti (viz tvrzení 2.3.3):

1. Respektuje sčítání: pro všechna celá čísla a, b, a', b' platí:
   jestliže platí $a \equiv b \pmod 6$ a současně $a' \equiv b' \pmod 6$, pak platí $a + a' \equiv b + b' \pmod 6$.

2. Respektuje násobení: pro všechna celá čísla a, b, a', b' platí:
   jestliže platí $a \equiv b \pmod 6$ a současně $a' \equiv b' \pmod 6$, pak platí $a \cdot a' \equiv b \cdot b' \pmod 6$.

Tyto dvě vlastnosti umožňují se na prvky množiny $\mathbb{Z}_6$ dívat jako na nová „čísla“ a zavést pro ně operace „sčítání“ $\oplus_6$ a „násobení“ $\odot_6$ následujícím způsobem:

1. Součtem „čísel“ $[3]_6$ a $[4]_6$ je „číslo“

$$[3]_6 \oplus_6 [4]_6 = [3+4]_6 = [7]_6 = [1]_6$$

2. Součinem „čísel“ $[3]_6$ a $[4]_6$ je „číslo“

$$[3]_6 \odot_6 [4]_6 = [3 \cdot 4]_6 = [12]_6 = [0]_6$$

Napíšeme nyní tabulku pro sčítání v $\mathbb{Z}_6$:

$\oplus_6$	$[0]_6$	$[1]_6$	$[2]_6$	$[3]_6$	$[4]_6$	$[5]_6$
$[0]_6$	$[0]_6$	$[1]_6$	$[2]_6$	$[3]_6$	$[4]_6$	$[5]_6$
$[1]_6$	$[1]_6$	$[2]_6$	$[3]_6$	$[4]_6$	$[5]_6$	$[0]_6$
$[2]_6$	$[2]_6$	$[3]_6$	$[4]_6$	$[5]_6$	$[0]_6$	$[1]_6$
$[3]_6$	$[3]_6$	$[4]_6$	$[5]_6$	$[0]_6$	$[1]_6$	$[2]_6$
$[4]_6$	$[4]_6$	$[5]_6$	$[0]_6$	$[1]_6$	$[2]_6$	$[3]_6$
$[5]_6$	$[5]_6$	$[0]_6$	$[1]_6$	$[2]_6$	$[3]_6$	$[4]_6$

Tabulka násobení v $\mathbb{Z}_6$ vypadá takto:

$\odot_6$	$[0]_6$	$[1]_6$	$[2]_6$	$[3]_6$	$[4]_6$	$[5]_6$
$[0]_6$	$[0]_6$	$[0]_6$	$[0]_6$	$[0]_6$	$[0]_6$	$[0]_6$
$[1]_6$	$[0]_6$	$[1]_6$	$[2]_6$	$[3]_6$	$[4]_6$	$[5]_6$
$[2]_6$	$[0]_6$	$[2]_6$	$[4]_6$	$[0]_6$	$[2]_6$	$[4]_6$
$[3]_6$	$[0]_6$	$[3]_6$	$[0]_6$	$[3]_6$	$[0]_6$	$[3]_6$
$[4]_6$	$[0]_6$	$[4]_6$	$[2]_6$	$[0]_6$	$[4]_6$	$[2]_6$
$[5]_6$	$[0]_6$	$[5]_6$	$[4]_6$	$[3]_6$	$[2]_6$	$[1]_6$

V dalším budeme často používat zjednodušené značení a namísto málo přehledného $[3]_6 \odot [4]_6 = [0]_6$ budeme psát

$$3 \cdot 4 = 0 \text{ v } \mathbb{Z}_6$$

Definujme nyní sčítání a násobení obecně:

2.3.11 Definice Ať $m > 1$ je pevné přirozené číslo. Na množině $\mathbb{Z}_m$ definujeme binární operace $\oplus_m$ a $\odot_m$ následujícím způsobem:

$$[a]_m \oplus_m [b]_m = [a+b]_m$$
$$[a]_m \odot_m [b]_m = [a \cdot b]_m$$

Připomeňme, že v definici 2.3.11 jsme na množině $\mathbb{Z}_m$ definovali binární operace $\oplus$ a $\odot$. Ve větě 2.3.13 ukážeme, že množina $\mathbb{Z}_m$ spolu s těmito operacemi se chová velmi podobně jako například množina všech celých čísel spolu s obvyklými operacemi sčítání a násobení. Nejprve však řekneme, které vlastnosti operací sčítání a násobení celých čísel nás zajímají.

2.3.12 Příklad Dobře známé operace $+$ (sčítání) a $\cdot$ (násobení) na množině celých čísel $\mathbb{Z}$ mají následující vlastnosti:

1. Sčítání i násobení jsou *komutativní* operace, tj. platí

$$a+b=b+a \quad \text{a} \quad a\cdot b=b\cdot a$$

pro všechna $a,b\in\mathbb{Z}$.

2. Sčítání i násobení jsou *asociativní* operace, tj. platí

$$a+(b+c)=(a+b)+c \quad \text{a} \quad a\cdot(b\cdot c)=(a\cdot b)\cdot c$$

pro všechna $a,b,c\in\mathbb{Z}$.

3. Obě operace mají *neutrální prvek*, tj. existují reálná čísla r_s a r_n taková, že platí

$$r_s+a=a \quad \text{a} \quad r_n\cdot a=a$$

pro všechna $a\in\mathbb{Z}$. (Samozřejmě: $r_s=0$ a $r_n=1$.)

4. Operace sčítání i násobení jsou navzájem svázány *distributivním zákonem*, tj. platí

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

pro všechna $a,b,c\in\mathbb{Z}$.

5. Každé celé číslo má *inversi vzhledem ke sčítání*, tj. pro každé reálné číslo a existuje (právě jedno) reálné číslo b takové, že

$$a+b=0.$$

(Samozřejmě: číslo b je tzv. *opačné číslo k číslu* a, tj. $b=-a$.)

Nyní již můžeme vyslovit větu, která říká, že sčítání a násobení v $\mathbb{Z}_m$ mají stejné vlastnosti jako sčítání a násobení v $\mathbb{Z}$. Přesněji: ukážeme, že $\mathbb{Z}_m$ má vlastnosti, vypsané v příkladu 2.3.12.

2.3.13 Věta *Ať $m>1$ je pevné přirozené číslo. Potom pro operace $\oplus$ a $\odot$ na množině $\mathbb{Z}_m$ platí následující:*

1. *Operace $\oplus$ je komutativní, asociativní, má neutrální prvek $[0]_m$ a každý prvek $[a]_m$ v $\mathbb{Z}_m$ má inversi $[-a]_m$ vzhledem k $\oplus$. To znamená, že pro libovolné prvky $[a]_m$, $[b]_m$, $[c]_m$ v $\mathbb{Z}_m$ platí:*

$$\begin{aligned}
[a]_m\oplus[b]_m &= [b]_m\oplus[a]_m\\
[a]_m\oplus([b]_m\oplus[c]_m) &= ([a]_m\oplus[b]_m)\oplus[c]_m\\
[a]_m\oplus[0]_m &= [a]_m\\
[a]_m\oplus[-a]_m &= [0]_m
\end{aligned}$$

2. *Operace $\odot$ je komutativní, asociativní a má neutrální prvek $[1]_m$. To znamená, že pro libovolné prvky $[a]_m$, $[b]_m$, $[c]_m$ v $\mathbb{Z}_m$ platí:*

$$\begin{aligned}
[a]_m\odot[b]_m &= [b]_m\odot[a]_m\\
[a]_m\odot([b]_m\odot[c]_m) &= ([a]_m\odot[b]_m)\odot[c]_m\\
[a]_m\odot[1]_m &= [a]_m
\end{aligned}$$

3. *Operace $\oplus$ a $\odot$ jsou svázány distributivním zákonem. To znamená, že pro libovolné prvky $[a]_m$, $[b]_m$, $[c]_m$ v $\mathbb{Z}_m$ platí:*

$$[a]_m\odot([b]_m\oplus[c]_m) = ([a]_m\odot[b]_m)\oplus([a]_m\odot[c]_m)$$

Důkaz. Důkaz rozdělíme na několik částí.

1. Pro libovolné prvky $[a]_m$, $[b]_m$, $[c]_m$ v $\mathbb{Z}_m$ platí:

$$\begin{aligned}
[a]_m \oplus [b]_m &= [a+b]_m = [b+a]_m = [b]_m \oplus [a]_m \\
[a]_m \oplus ([b]_m \oplus [c]_m) &= [a]_m \oplus [b+c]_m = [a+(b+c)]_m = \\
&= [(a+b)+c]_m = [a+b]_m \oplus [c]_m = \\
&= ([a]_m \oplus [b]_m) \oplus [c]_m \\
[a]_m \oplus [0]_m &= [a+0]_m = [a]_m \\
[a]_m \oplus [-a]_m &= [a-a]_m = [0]_m
\end{aligned}$$

2. Pro libovolné prvky $[a]_m$, $[b]_m$, $[c]_m$ v $\mathbb{Z}_m$ platí:

$$\begin{aligned}
[a]_m \odot [b]_m &= [a \cdot b]_m = [b \cdot a]_m = [b]_m \odot [a]_m \\
[a]_m \odot ([b]_m \odot [c]_m) &= [a]_m \odot [b \cdot c]_m = [a \cdot (b \cdot c)]_m = \\
&= [(a \cdot b) \cdot c]_m = [a \cdot b]_m \odot [c]_m = \\
&= ([a]_m \odot [b]_m) \odot [c]_m \\
[a]_m \odot [1]_m &= [a \cdot 1]_m = [a]_m
\end{aligned}$$

3. Pro libovolné prvky $[a]_m$, $[b]_m$, $[c]_m$ v $\mathbb{Z}_m$ platí:

$$\begin{aligned}
[a]_m \odot ([b]_m \oplus [c]_m) &= [a]_m \odot [b+c]_m = [a \cdot (b+c)]_m = \\
&= [(a \cdot b) + (a \cdot c)]_m = [a \cdot b]_m \oplus [a \cdot c]_m = \\
&= ([a]_m \odot [b]_m) \oplus ([a]_m \odot [c]_m)
\end{aligned}$$

■

Následující definice je abstrakcí námi uvedených vlastností operací $\oplus$ a $\odot$ na množině $\mathbb{Z}_m$ z věty 2.3.13.

2.3.14 Definice Ať K je neprázdná množina, na které jsou definovány dvě binární operace, $+$ a $\cdot$, tj. funkce $+ : K \times K \longrightarrow K$ a $\cdot : K \times K \longrightarrow K$. Namísto zápisu $+(a,b)$ pro hodnotu operace $+$ v argumentech a a b budeme psát obvyklejší *infixní zápis* $a+b$ (podobně pro operaci $\cdot$).

Uspořádanou trojici $\mathbb{K} = \langle K, +, \cdot \rangle$ nazveme *okruhem*, pokud platí následující tři podmínky:

1. Operace $+$ je komutativní, asociativní, má neutrální prvek a existují inversní prvky vzhledem k $+$, tj. platí:

   (K+) Operace $+$ je *komutativní*, tj. platí $a+b = b+a$, pro všechna $a, b \in K$.

   (A+) Operace $+$ je *asociativní*, tj. platí $a+(b+c) = (a+b)+c$, pro všechna $a, b, c \in K$.

   (N+) Operace $+$ má *neutrální prvek*, tj. existuje prvek $0 \in K$ tak, že platí $0+a = a$, pro všechna $a \in K$.

   (I+) Každý prvek K má *inversi vzhledem k* $+$, tj. pro každé $a \in K$ existuje právě jeden prvek $b \in K$ tak, že $a+b = 0$. Tento prvek budeme značit $-a$.

2. Operace $\cdot$ je *asociativní*, tj. platí:

   (A$\cdot$) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$, pro všechna $a, b, c \in K$.

3. Operace $\cdot$ je *distributivní vzhledem k operaci* $+$, tj. platí:

   (LD) *Levý distributivní zákon*: platí $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$ pro všechna $a, b, c \in K$.

   (PD) *Pravý distributivní zákon*: platí $(b+c) \cdot a = (b \cdot a) + (c \cdot a)$ pro všechna $a, b, c \in K$.

Okruh $\mathbb{K} = \langle K, +, \cdot \rangle$ nazveme *komutativním okruhem*, pokud i operace $\cdot$ je *komutativní*, tj. pokud platí:

(K$\cdot$) $a \cdot b = b \cdot a$, pro všechna $a, b \in K$.

Okruh $\mathbb{K} = \langle K, +, \cdot \rangle$ nazveme *okruh s jednotkou*, pokud i operace $\cdot$ má *neutrální prvek*, tj. pokud platí:

(N$\cdot$) Existuje prvek $1 \in K$ tak, že $1 \cdot a = a \cdot 1 = a$, pro všechna $a \in K$.

(U okruhu s jednotkou tedy obecně nevyžadujeme komutativitu operace $\cdot$.)

Budeme-li chtít neutrální prvky zdůraznit, budeme okruh s jednotkou psát jako uspořádanou pětici $\mathbb{K} = \langle K, +, \cdot, 0, 1 \rangle$. Dále v tomto kontextu budeme obě operace i oba neutrální prvky nazývat obvyklými jmény, tj. $+$ čteme *sčítání*, $\cdot$ čteme *násobení*, prvku 0 říkáme *nula*, prvku 1 říkáme *jednotka*.

2.3.15 Poznámka Pro počítání v okruhu zavedeme nyní některé konvence:

1. Budeme vždy předpokládat, že násobení *váže silněji* než sčítání, tj. například namísto $(b \cdot a) + (c \cdot a)$ budeme psát $b \cdot a + c \cdot a$.
2. Díky asociativitě sčítání a násobení budeme často *vynechávat závorky*, tj. například místo $a + (b + c)$ budeme psát $a + b + c$.
3. Opakované sčítání budeme značit jako *násobení kladným přirozeným číslem*, tj. například $3 \cdot a$ je zkratka za $a + a + a$.
4. Opakované násobení budeme značit jako *mocnění na kladné přirozené číslo*, tj. například a^3 je zkratka za $a \cdot a \cdot a$. Má-li daný okruh jednotku, budeme výrazem a^0 značit 1.

2.3.16 Příklady Uveďme nyní několik příkladů okruhů:

1. Množina $\mathbb{R}$ všech reálných čísel spolu s obyklými operacemi sčítání a násobení tvoří komutativní okruh s jednotkou.
2. Množina $\mathbb{Q}$ všech racionálních čísel spolu s obyklými operacemi sčítání a násobení tvoří komutativní okruh s jednotkou.
3. Množina $\mathbb{Z}$ všech celých čísel spolu s obyklými operacemi sčítání a násobení tvoří komutativní okruh s jednotkou.
4. Množina $\mathbb{Z}_m$ spolu s operacemi $\oplus$ a $\odot$ tvoří komutativní okruh s jednotkou. To je obsahem věty 2.3.13.
5. Označme jako $\mathsf{Mat}_{n\times n}(\mathbb{R})$ množinu všech čtvercových matic rozměrů $n \times n$ s reálnými položkami. Potom množina $\mathsf{Mat}_{n\times n}(\mathbb{R})$ spolu s obyklými operacemi sčítání a násobení matic tvoří okruh.

   Je-li $n \geq 2$, pak tento okruh *není* komutativní: násobení matic $n \times n$ není komutativní pro $n \geq 2$. Tudíž, pro $n \geq 2$, není *žádný* z okruhů $\mathsf{Mat}_{n\times n}(\mathbb{R})$ komutativní. Protože matice 1×1 jsou reálná čísla, okruh $\mathsf{Mat}_{1\times 1}(\mathbb{R})$ *je* komutativní.

   Každý okruh $\mathsf{Mat}_{n\times n}(\mathbb{R})$ je však okruh s jednotkou: jednotková matice rozměrů $n \times n$ je neutrálním prvkem vzhledem k násobení matic.
6. Označme jako $\mathbb{R}[x]$ množinu všech polynomů s reálnými koeficienty v neurčité x.[6] Potom množina $\mathbb{R}[x]$ spolu s obyklými operacemi sčítání a násobení polynomů tvoří komutativní okruh s jednotkou.
7. Označme jako F množinu všech funkcí z $\mathbb{R}$ do $\mathbb{R}$, které mají konečnou první derivaci ve všech bodech. Potom množina F spolu s obvyklými operacemi sčítání funkcí a skládání funkcí tvoří okruh. (Vyjmenujte věty z matematické analýzy, které se za tímto tvrzením skrývají.) Nejde však o komutativní okruh: skládání funkcí není komutativní.

   Jde však o okruh s jednotkou: funkce $f(x) = x$ má zřejmě konečnou první derivaci a je neutrálním prvkem vzhledem ke skládání funkcí.

[6] V matematické analýze jste byli zvyklí říkat znaku x proměnná. Jak uvidíme později v kapitole 4, budeme muset rozlišovat mezi polynomem jako funkcí (tam je termín *proměnná* v pořádku) a polynomem jako výrazem (tam budeme používat termín *neurčitá*, anglicky *indeterminate*).

Některé z okruhů v příkladech 2.3.16 mají další speciální vlastnost: každý nenulový prvek má inversi vzhledem k operaci násobení. (Podrobněji viz cvičení 2.4.21.) Dejme této speciální vlastnosti jméno.

2.3.17 Definice Komutativní okruh s jednotkou $\mathbb{K} = \langle K, +, \cdot, 0, 1\rangle$ nazveme *těleso*, pokud každý nenulový prvek v K má inversi vzhledem k operaci $\cdot$, tj. pokud pro každé $a \in K$, $a \neq 0$ existuje právě jedno $b \in K$ takové, že $a \cdot b = 1$. Tomuto číslu b budeme říkat *inverse k* a a budeme jej značit a^{-1}.

Například množina reálných čísel $\mathbb{R}$ (spolu s obvyklým sčítáním a násobením) tvoří těleso. Později uvidíme, že toto je vlastnost reálných čísel, která umožňuje vybudovat teorii klasické lineární algebry: množina skalárů je $\mathbb{R}$. Později budeme v aplikacích (například v kódování) chtít $\mathbb{R}$ nahradit jinou „množinou čísel“. Když jsme zavedli $\mathbb{Z}_m$ jako množinu čísel, vzniká otázka, zda je každé $\mathbb{Z}_m$ těleso. Jak ukazuje následující příklad, není tomu tak vždy.

2.3.18 Příklad Ukážeme, že v komutativním okruhu $\langle \mathbb{Z}_4, \oplus, \odot\rangle$ nemá prvek $[2]_4$ inversi vzhledem k operaci $\odot$. Protože množina $\mathbb{Z}_4$ obsahuje právě čtyři různé prvky, musel by hledanou inversí být některý z prvků $[0]_4$, $[1]_4$, $[2]_4$, $[3]_4$. Lze však vyzkoušet, že žádný z těchto prvků nevyhovuje podmínce $[2]_4 \odot [x]_4 = [1]_4$:

$$\begin{aligned} [2]_4 \odot [0]_4 &= [0]_4 \\ [2]_4 \odot [1]_4 &= [2]_4 \\ [2]_4 \odot [2]_4 &= [0]_4 \\ [2]_4 \odot [3]_4 &= [2]_4 \end{aligned}$$

Některé nenulové prvky množiny $\mathbb{Z}_4$ však inversi vzhledem k operaci $\odot$ mají, například $[1]_4 \odot [1]_4 = [1]_4$, tedy $[1]_4^{-1} = [1]_4$. Podobně $[3]_4 \odot [3]_4 = [1]_4$, tedy $[3]_4^{-1} = [3]_4$.

Otázkám existence inverse zhledem k operaci $\odot$ v $\mathbb{Z}_m$ bude věnována další kapitola.

2.3.19 Příklad Tělesa jsme definovali jako *komutativní* okruh, kde každý nenulový prvek má inversi. Existují i důležité příklady „nekomutativních těles“. Nejznámějším je okruh *kvaternionů*, který zobecňuje komplexní čísla. Obecný kvaternion je zápis

$$a + bi + cj + dk$$

kde a, b, c, d jsou reálná čísla a i, j, k jsou „odmocniny z -1“, tj. platí rovnosti[7]

$$i^2 = j^2 = k^2 = ijk = -1$$

Pomocí výše uvedených rovností je snadné v okruhu kvaternionů sčítat a násobit, inspirujte se komplexními čísly. Násobení však není komutativní: platí například rovnost $ij = -ji$ (dokažte to). Dokažte také, že každý nenulový kvaternion má inversní prvek vzhledem k násobení.

Kvaterniony mají velmi důležité postavení v moderní fyzice, viz například knihu

☞ R. Penrose, *The Road to Reality — A Complete Guide to the Laws of the Universe*, Jonathan Cape, Londýn, 2004

2.4 Cvičení

2.4.1 Cvičení Ukažte, že pro dělitelnost v oboru přirozených čísel platí:

1. $|$ je binární relace na množině $\mathbb{N}$.
2. $|$ je reflexivní relace na množině $\mathbb{N}$, tj. platí:

[7] Tyto rovnosti vyryl 16. října 1843 jejich objevitel William Rowan Hamilton (1805–1865) do kamene na mostě Brougham Bridge v Dublinu. Hamilton o zobecnění komplexních čísel dlouho přemýšlel a řešení ho prý napadlo při procházce s manželkou podél Royal Canal.

pro všechna $a \in \mathbb{N}$ platí: $a \mid a$.

3. $\mid$ je transitivní relace na množině $\mathbb{N}$, tj. platí:

   pro všechna $a, b, c \in \mathbb{N}$ platí: jestliže $a \mid b$ a současně $b \mid c$, potom $a \mid c$.

4. $\mid$ je antisymetrická relace na množině $\mathbb{N}$, tj. platí:

   pro všechna $a, b \in \mathbb{N}$ platí: jestliže $a \mid b$ a současně $b \mid a$, potom $a = b$.

5. $\mid$ je relace uspořádání na množině $\mathbb{N}$.

Které z těchto vlastností se poruší, budete-li studovat dělitelnost v oboru celých čísel?

2.4.2 Cvičení Ukažte, že pro prvočíslo p a libovolná přirozená čísla a, b platí:

Jestliže $p \mid (a \cdot b)$, potom $p \mid a$ nebo $p \mid b$.

Návod: použijte větu 2.1.5.

2.4.3 Cvičení Ať $b > 1$ je pevné přirozené číslo. *Rozvoj přirozeného čísla x o základu b* je zápis

$$x = \sum_{i=0}^{n} x_i \cdot b^i$$

kde n je nějaké přirozené číslo a všechna čísla x_i jsou v rozmezí od 0 do $b - 1$.

Často se používá značení

$$(x)_b = (x_n, \ldots, x_0)$$

a v tomto kontextu se číslům $x_n, \ldots, x_0$ říká *cifry b-rozvoje čísla x*. Například

$$(37\,265)_{10} = (3, 7, 2, 6, 5) \qquad (29)_2 = (1, 1, 1, 0, 1)$$

Vysvětlete, jak použít tvrzení 2.1.10 k důkazu existence a jednoznačnosti rozvoje o základu b pro každé přirozené číslo x.

Vymyslete (rekursivní) algoritmus, který pro libovolné x spočte b-rozvoj čísla x. Nalezněte variant a invariant tohoto algoritmu a dokažte jeho totální korektnost.

2.4.4 Cvičení Ukažte, že pro libovolná přirozená čísla a, b, c platí:

$$\gcd(a, \gcd(b, c)) = \gcd(\gcd(a, b), c)$$

Výsledek využijte k (rekursivní) definici největšího společného dělitele libovolného (konečného) počtu přirozených čísel.

2.4.5 Cvičení Využijte cvičení 2.4.4 k návrhu algoritmu počítajícího největšího společného dělitele libovolného (konečného) počtu přirozených čísel. Dokažte totální korektnost tohoto algoritmu.

Platí analogie Bezoutovy rovnosti? Pokud ano, zformulujte ji a dokažte.

2.4.6 Cvičení Ukažte, že přirozené číslo $p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r}$, kde $r \geq 1$ je přirozené číslo, $p_1, p_2, \ldots, p_r$ jsou navzájem různá prvočísla a $n_1, n_2, \ldots, n_r$ jsou kladná přirozená čísla má celkem

$$(n_1 + 1) \cdot (n_2 + 1) \cdot \ldots \cdot (n_r + 1)$$

různých dělitelů.

2.4.7 Cvičení Trojici a, b, c kladných přirozených čísel budeme říkat *pythagorejská*, pokud je splněna rovnost $a^2 + b^2 = c^2$. Asi nejznámější pythagorejskou trojicí je trojice $a = 3$, $b = 4$, $c = 5$. Každou pythagorejskou trojici a, b, c si můžeme představit jako délky stran pravoúhlého trojúhelníka, protože rovnost $a^2 + b^2 = c^2$ je pak známá *Pythagorova věta* (viz příklad 3.4.9).

Ukažte následující:

1. Pro každou pythagorejskou trojici a, b, c platí rovnost $\gcd(a,b) = \gcd(b,c) = \gcd(a,c)$.

2. Řekneme, že pythagorejská trojice a, b, c je *primitivní*, když platí $\gcd(a,b) = \gcd(b,c) = \gcd(a,c) = 1$. Příkladem primitivní pythagorejské trojice je $a = 3$, $b = 4$, $c = 5$.

   Ukažte, že v primitivní pythagorejské trojici a, b, c musí být jedno z čísel a, b sudé a druhé liché. Návod: postupujte sporem a využijte počítání v $\mathbb{Z}_2$ a v $\mathbb{Z}_4$.

3. Ať jsou dána kladná čísla u, v, která nejsou obě lichá, a pro která platí $u > v$ a $\gcd(u,v) = 1$. Ukažte, že trojice čísel
   $$a = u^2 - v^2 \qquad b = 2uv \qquad c = u^2 + v^2$$
   je primitivní pythagorejská trojice. (Uvedené vzorce dokonce charakterizují *všechny* primitivní pythagorejské trojice. Důkaz není těžký, je však poměrně zdlouhavý.)

2.4.8 Cvičení Pro prvočíslo p a přirozené číslo a zaveďte značení

$p^n \preceq a$ právě tehdy, když platí $p^n \mid a$ a neplatí $p^{n+1} \mid a$.

Rozhodněte, zda platí:

1. Jestliže $p^n \preceq a$ a $p^m \preceq b$, potom $p^{n+m} \preceq (a \cdot b)$.

2. Jestliže $p^n \preceq a$ a $p^n \preceq b$, potom $p^n \preceq (a + b)$.

2.4.9 Cvičení Dokažte, že pro posloupnost Fibonacciho čísel (viz cvičení 1.6.22) $\{F(n)\}_{n=1}^{\infty}$ platí rovnost

$$\gcd(F(n), F(n+1)) = \gcd(F(n+1), F(n+2))$$

pro všechna $n \geq 1$. Tohoto faktu využijte k důkazu tvrzení, že každé dvě po sobě jdoucí Fibonacciho čísla jsou nesoudělná.

Vysvětlete, proč z toho plyne, že největšího počtu kroků Eukleidova algorimu je dosaženo pro $a = F(n+1)$, $b = F(n)$. (Vyzkoušejte běh algoritmu pro $a = F(20) = 6765$ a $b = F(19) = 4181$.)

2.4.10 Cvičení Označte jako $N(a,b)$ počet kroků, které Eukleidův algoritmus vykoná při hledání $\gcd(a,b)$ přirozených čísel $a \geq b$. Z tvrzení 2.1.21 víme, že platí nerovnost $N(a,b) < 5n$, kde n je počet cifer v decimálním rozvoji čísla b.

Zde je jiný způsob, jak dokázat tvrzení 2.1.21:

1. Dokažte: pokud $a > b$ a $a' = a + k \cdot b$ pro nějaké přirozené číslo k, pak platí $N(a,b) = N(a',b)$.

2. Dokažte: $N(a,b) \leq 3$, jestliže $b < 8$ a $a > b$.

3. Dokažte *Lamého větu*:

   Ať $b < F(n)$. Potom platí $N(a,b) \leq n - 3 < n - 2 = N(F(n+1), F(n))$.

   (Návod: použijte princip indukce.)

4. Dokažte, že pro všechna $n \geq 5$ platí nerovnost $\dfrac{F(n+5)}{F(n)} \geq 10$.

5. Dokažte, že $F(5n+2)$ má alespoň $n+1$ cifer.

6. Z Lamého věty odvoďte tvrzení 2.1.21.

2.4.11 Cvičení Ať R je binární relace na množině A. Ukažte, že je ekvivalentní:

1. Platí $R \subseteq \Delta_A$.

2. Relace R je symetrická a antisymetrická současně.

Návod: využijte tvrzení 2.2.10.

2.4.12 Cvičení Ať R je binární relace na množině A. Ukažte, že je ekvivalentní:

1. Platí $R = \Delta_A$.
2. Relace R je reflexivní, symetrická a antisymetrická současně.
3. Relace R je reflexivní, symetrická, transitivní a antisymetrická současně.

Návod: využijte tvrzení 2.2.10.

2.4.13 Cvičení Ať $A = \{a, b, c\}$. Ukažte, že

1. Relace $\{(a,a),(b,b),(a,b),(b,c),(b,a)\}$ je reflexivní, není symetrická, není transitivní, není antisymetrická.
2. Relace $\{(a,b),(b,a),(b,c),(c,b)\}$ není reflexivní, je symetrická, není transitivní, není antisymetrická.
3. Relace $\{(a,b),(b,c),(a,c),(b,a),(a,a)\}$ není reflexivní, není symetrická, je transitivní, není antisymetrická.
4. Relace $\{(a,b),(b,c)\}$ není reflexivní, není symetrická, není transitivní, je antisymetrická.

Dejte příklady relací na množině A, které mají vždy přesně dvě vlastnosti z vlastností reflexivita, symetrie, transitivita a antisymetrie.

2.4.14 Cvičení Řekneme, že relace R je *totální*, když pro všechna $x, y \in A$ platí: $x \; R \; y$ a současně $y \; R \; x$. Zodpovězte následující otázky:

1. Je každá symetrická relace totální?
2. Je každá totální relace symetrická?

Popište všechny totální relace.

2.4.15 Cvičení Indukcí dokažte:

1. $10^k \equiv 1 \pmod 3$ pro všechna $k \geq 0$.
2. $10^k \equiv 1 \pmod 9$ pro všechna $k \geq 0$.
3. $k^5 - k \equiv 0 \pmod{30}$ pro všechna $k \geq 0$.

2.4.16 Cvičení Každé přirozené číslo a lze psát jednoznačně v decimálním rozvoji jako $\sum_{k=0}^{n} a_k \cdot 10^k$, kde všechna čísla a_k jsou přirozená čísla v rozmezí od 0 do 9. Z předchozího cvičení víte, že platí $10^k \equiv 1 \pmod 3$ pro všechna $k \geq 0$. Využijte toho k důkazu, že platí

$$\sum_{k=0}^{n} a_k \cdot 10^k \equiv \sum_{k=0}^{n} a_k \pmod 3$$

a odvoďte tedy kriterium dělitelnosti třemi:

Přirozené číslo je dělitelné třemi právě tehdy, když je třemi dělitelný jeho ciferný součet.

2.4.17 Cvičení Odvoďte kriterium dělitelnosti devíti. Návod: využijte cvičení 2.4.15.

2.4.18 Cvičení Navrhněte kriterium dělitelnosti jedenácti. Návod: zjistěte, s čím jsou kongruentní mocniny 10 modulo 11.

2.4.19 Cvičení Označte $x = \sum_{k=0}^{128} k!$ v $\mathbb{Z}_7$. Spočtěte x. Návod: využijte vlastností kongruence modulo 7.

2.4.20 Cvičení Ať $\mathbb{K}$ je (libovolný) okruh. Definujte $\mathbb{K}[x]$ jako množinu všech polynomů v neurčité x majících koeficienty v $\mathbb{K}$. Pro tyto polynomy definujte sčítání a násobení. Dokažte, že $\mathbb{K}[x]$ je okruh. Je $\mathbb{K}[x]$ vždy komutativní okruh? (V této obecnosti budeme polynomy studovat v kapitole 4.)

2.4.21 Cvičení Které příklady okruhů v příkladech 2.3.16 jsou tělesa?

Revize kapitoly

Dozvěděli jsme se:

- ✔ Prvočísla tvoří základní stavební kameny přirozených čísel — to tvrdí *základní věta elementární teorie čísel*. Najít prvočíselný rozklad přirozeného čísla je výpočetně náročné.
- ✔ V oboru celých čísel platí *věta o dělení se zbytkem*, která je základem *(rozšířeného) Eukleidova algoritmu* pro hledání *největšího společného dělitele* dvou čísel.
- ✔ *Kongruence modulo číslo* umožňuje „změnit optiku“ a *sčítat a násobit* zbytky po dělení. Vlastnosti sčítání a násobení, které známe z celých čísel, zůstávají zachovány.
- ✔ Zajímavé a důležité vlastnosti sčítání a násobení jsme vystihli abstraktně v definicích *okruhu* a *tělesa*.

Pro přípravu na zkoušku zkuste zodpovědět následující otázky:

- ✔ Šlo by dokázat větu o dělení se zbytkem i v oboru přirozených čísel? V oboru reálných čísel?
- ✔ Podrobně analyzujte chování rozšířeného Eukleidova algoritmu: jeho terminaci, parciální korektnost, výpočetní složitost (dobu zpracování dat).
- ✔ Podrobně popište všechna fakta z lineární algebry, která nám umožňují říci, že čtvercové matice (pevných rozměrů) tvoří okruh s jednotkou. Tvoří matice rozměrů 2×3 okruh?

Doplňující literatura

O zavedení počítání modulo a o pojmu okruh se lze dočíst například v kapitole II, odstavcích 4 a 5 a kapitole IV, odstavci 1 knihy

- ☞ S. MacLane a G. Birkhoff, *Algebra*, Alfa, Bratislava, 1974

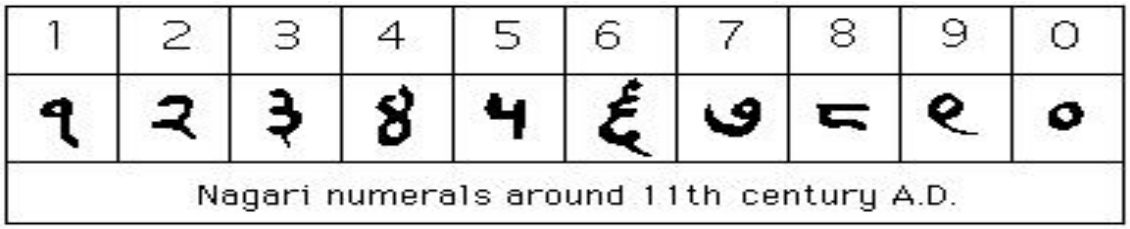
Základní vlastnosti těchto pojmů jsou vysvětleny i v kapitole III.9 knihy

- ☞ J. Adámek, *Kódování*, SNTL, Praha, 1989

obšírnější informace lze ovšem nalézt v angličtině, například v knihách

- ☞ L. Childs, *A Concrete Introduction to Higher Algebra*, Springer, New York, 1995
- ☞ J. Adámek, *Foundations of Coding*, John Wiley & Sons, New York, 1991

Ukázka indických číslic, ze kterých se vyvinuly dnešní číslice:



Převzato z

- ☞ http://www-history.mcs.st-andrews.ac.uk/HistTopics/Indian_numerals.html

Kapitola 3

Počítání modulo — část 2

Mezi změtí písmen jich jedenáct bylo vyrovnáno do úhledné řádky. Dávala dvě slova: „ČTYŘICET DVA.“
„Grrrugh guh guh“, vysvětlil domorodec.
Douglas Adams, *Restaurant na konci vesmíru*

V této kapitole vybudujeme základy *lineární algebry* nad $\mathbb{Z}_m$, kterou potřebuje teorie *lineárních kódů*. Dále se seznámíme s dalšími jednoduchými, ale velmi užitečnými výsledky počítání modulo — s *čínskou větou o zbytcích* (věta 3.4.1) a *Eulerovou větou* (věta 3.4.13), které budeme aplikovat na *šifrovací protokol RSA*.

Začneme systematicky používat zjednodušené značení. Budeme tak například namísto $[3]_4 \odot_4 [2]_4 = [2]_4$ psát

$$3 \cdot 2 = 2 \quad \text{v } \mathbb{Z}_4$$

Podobně budeme například místo $\mathbb{Z}_4 = \{[0]_4, [1]_4, [2]_4, [3]_4\}$ psát

$$\mathbb{Z}_4 = \{0, 1, 2, 3\}$$

3.1 Lineární rovnice v $\mathbb{Z}_m$

Základem lineární algebry nad reálnými čísly je analýza řešení lineárních rovnic. Předvedeme, že situace se komplikuje, pokud místo reálných čísel použijeme čísla ze $\mathbb{Z}_m$.

3.1.1 Příklad Uvažujme nejprve o lineární rovnici

$$6x = 9 \text{ v } \mathbb{R}$$

a ukažme podrobně, jak ji řešíme: především si uvědomíme, že $6 \neq 0$, existuje tedy reálné číslo 6^{-1} (*inverse* čísla 6) s vlastností $6 \cdot 6^{-1} = 6^{-1} \cdot 6 = 1$ (toto číslo je pochopitelně zlomek $\frac{1}{6}$). Výše uvedenou rovnici pak číslem 6^{-1} vynásobíme zleva a dostaneme tak ekvivalentní problém:

$$6^{-1} \cdot (6x) = 6^{-1} \cdot 9 \text{ v } \mathbb{R}$$

Využijeme nyní asociativity násobení a přepíšeme rovnici na problém

$$(6^{-1} \cdot 6) \cdot x = 6^{-1} \cdot 9 \text{ v } \mathbb{R}$$

a konečně, s využitím $6 \cdot 6^{-1} = 6^{-1} \cdot 6 = 1$ píšeme

$$x = 6^{-1} \cdot 9 = \frac{9}{6} \text{ v } \mathbb{R}$$

Co se stane, když se zaměříme na podobný problém, tentokrát v jistém $\mathbb{Z}_m$? Například: chceme vyřešit

$$6x = 12 \text{ v } \mathbb{Z}_{13}$$

Existuje číslo 6^{-1} v $\mathbb{Z}_{13}$ (*inverse* čísla 6) s vlastností $6 \cdot 6^{-1} = 6^{-1} \cdot 6 = 1$? Ano, jde o číslo 11, protože v $\mathbb{Z}_{13}$ platí $6 \cdot 11 = 11 \cdot 6 = 66 = 1$. (Číslo 6^{-1} jsme nalezli „hrubou silou“, tj. vyzkoušením všech třinácti možných kandidátů. Algoritmus pro hledání inverse předvedeme v příkladu 3.1.5.)

Výše uvedenou rovnici pak číslem 6^{-1} vynásobíme zleva a dostaneme tak ekvivalentní problém:

$$6^{-1} \cdot (6x) = 6^{-1} \cdot 12 \text{ v } \mathbb{Z}_{13}$$

Využijeme nyní asociativity násobení a přepíšeme rovnici na problém

$$(6^{-1} \cdot 6) \cdot x = 6^{-1} \cdot 12 \text{ v } \mathbb{Z}_{13}$$

a konečně, s využitím $6 \cdot 6^{-1} = 6^{-1} \cdot 6 = 1$ píšeme

$$x = 6^{-1} \cdot 12 = 11 \cdot 12 = 132 = 2 \text{ v } \mathbb{Z}_{13}$$

Zdánlivě je tedy vše stejné jako v reálných číslech. Je to proto, že $\mathbb{Z}_{13}$ má podobné vlastnosti jako $\mathbb{R}$ — obě struktury jsou *tělesa* (viz definice 2.3.17 a důsledek 3.1.6).

V případech, kdy příslušné $\mathbb{Z}_m$ těleso není, mohou se dít nezvyklé věci:

Lineární rovnice může mít v $\mathbb{Z}_m$ více než jedno řešení:

$$6x = 12 \text{ v } \mathbb{Z}_{15}$$

Množina $\mathbb{Z}_{15}$ těleso není, číslo $6 \neq 0$ *nemá* v $\mathbb{Z}_{15}$ inversi. (Ověřte to vyzkoušením všech kandidátů.) Daná rovnice má *přesně tři* různá řešení: $x_1 = 2$, $x_2 = 7$ a $x_3 = 12$. (Zatím jsme nepodali žádný algoritmus pro nalezení řešení, řešili jsme rozborem všech možných situací. Algoritmus popíšeme ve větě 3.1.2, viz také příklad 3.1.3.)

Poznamenejme ještě, že lineární rovnice v $\mathbb{Z}_m$ nemusí mít *žádné řešení* (to je analogické situaci v reálných číslech):

$$0x = 6 \text{ v } \mathbb{Z}_9$$

Zjevně žádné x s vlastností $0x = 6$ v $\mathbb{Z}_9$ neexistuje.

Podrobnou klasifikaci existence a počtu řešení lineární rovnice v $\mathbb{Z}_m$ podává následující věta.

3.1.2 Věta *Ať a a m jsou přirozená čísla, $m \geq 2$. Předpokládejme, že $\gcd(a, m) = d > 0$. Potom lineární rovnice*

$$ax = b \quad \text{v } \mathbb{Z}_m$$

má řešení právě tehdy, když $d \mid b$.

Navíc, jestliže $d \mid b$, má tato rovnice v $\mathbb{Z}_m$ právě d různých řešení.

Důkaz. Nejprve ukážeme, že pokud řešení rovnice $ax = b$ v $\mathbb{Z}_m$ existuje, pak $d \mid b$. Když totiž $au = b$ v $\mathbb{Z}_m$, existuje celé číslo k tak, že v $\mathbb{Z}$ platí $au - b = k \cdot m$, neboli $au - km = b$. Protože levá strana je dělitelná číslem d, musí být číslem d dělitelná i pravá strana, což jsme chtěli ukázat.

Předpokládejme nyní, že $d \mid b$. Chceme ukázat, že lineární rovnice $ax = b$ má v $\mathbb{Z}_m$ právě d různých řešení.

1. Předpokládejme nejprve, že $d = 1$. Nejprve ukážeme, že rovnice $ax = b$ má v $\mathbb{Z}_m$ řešení. To ale ihned plyne z Bezoutovy rovnosti: existují totiž celá čísla α, β taková, že

   $$a \cdot \alpha + m \cdot \beta = 1$$

   Potom je $a \cdot \alpha \cdot b + m \cdot \beta \cdot b = b$ v $\mathbb{Z}$ a tudíž $a \cdot (\alpha \cdot b) = b$ v $\mathbb{Z}_m$. Řešením je $x = \alpha \cdot b$.

   Zbývá ukázat, že toto řešení je v $\mathbb{Z}_m$ jediné. Předpokládejme, že v $\mathbb{Z}_m$ platí $au = b$ a $av = b$. Potom $a(u - v) = 0$ v $\mathbb{Z}_m$ a protože $\gcd(a, m) = 1$, platí $u - v = 0$, čili $u = v$ v $\mathbb{Z}_m$.

2. Předpokládejme nyní, že $d > 1$. Rovnici $ax = b$ v $\mathbb{Z}_m$ můžeme upravit na rovnici

$$\frac{a}{d}x = \frac{b}{d} \quad \text{v } \mathbb{Z}_{\frac{m}{d}}$$

Protože $\gcd(\frac{a}{d}, \frac{m}{d}) = 1$, má tato rovnice podle předchozího jediné řešení v $\mathbb{Z}_{\frac{m}{d}}$. Označme toto řešení jako u. Toto u je ale také řešením původní rovnice $ax = b$ v $\mathbb{Z}_m$. Ukázali jsme, že řešení existuje.

Abychom ukázali, že rovnice $ax = b$ má v $\mathbb{Z}_m$ přesně d různých řešení, uvědomme si, že jestliže $au = b$ a $av = b$ v $\mathbb{Z}_m$, potom $u = v$ v $\mathbb{Z}_{\frac{m}{d}}$.

Každé řešení původní rovnice lze tedy napsat ve tvaru

$$u_k = u + k \cdot \frac{m}{d}, \quad \text{kde } k \in \{0, 1, \ldots, d-1\}$$

kde u je jedno pevně zvolené řešení. Zvolme různá čísla $k, l \in \{0, 1, \ldots, d-1\}$ a předpokládejme, že $u_k = u_l$ v $\mathbb{Z}_m$. Potom existuje celé číslo r takové, že v $\mathbb{Z}$ platí

$$(k - l) \cdot \frac{m}{d} = rm$$

neboli $m(k - l) = rmd$. Po zkrácení číslem m tedy musí v $\mathbb{Z}$ platit rovnost $k - l = rd$, čili $|k - l| = |r| \cdot d$. Protože $k, l \in \{0, 1, \ldots, d-1\}$, platí $|k - l| \leq d - 1$ a tudíž z rovnosti $|k - l| = |r| \cdot d$ plyne, že $k = l$. To je spor, my jsme předpokládali, že $k \neq l$.

Prvky u_k jsou tedy v $\mathbb{Z}_m$ navzájem různé a řeší rovnici $ax = b$ v $\mathbb{Z}_m$.

■

3.1.3 Příklad Předchozí věta vysvětluje, jak vyřešit lineární rovnici

$$6x = 12 \text{ v } \mathbb{Z}_{15}$$

z příkladu 3.1.1. Nejprve spočteme $\gcd(6, 15) = 3$. Protože $3 \mid 12$, řešení existuje. Podle druhé části věty 3.1.2 existují přesně 3 různá řešení dané rovnice. Nejprve zadanou rovnici redukujeme na rovnici

$$2x = 4 \text{ v } \mathbb{Z}_5$$

s jediným řešením

$$x = 2^{-1} \cdot 4 = 3 \cdot 4 = 12 = 2 \text{ v } \mathbb{Z}_5$$

protože $2^{-1} = 3$ v $\mathbb{Z}_5$ (to jsme opět našli vyzkoušením všech možností, algoritmus pro nalezení inverse je popsán v příkladu 3.1.5). V $\mathbb{Z}$ tedy platí, že $x = 2 + 5k$, kde $k = \ldots, -2, -1, 0, 1, 2, \ldots$ Volbou $k = 0$, $k = 1$, $k = 2$ dostáváme tři různá řešení $x_1 = 2$, $x_2 = 2 + 5 = 7$, $x_3 = 2 + 2 \cdot 5 = 12$ v $\mathbb{Z}_{15}$.

Najít inversi k prvku a v $\mathbb{Z}_m$ znamená umět vyřešit v $\mathbb{Z}_m$ lineární rovnici $ax = 1$ *jednoznačně*. Dostáváme tedy následující důsledek:

3.1.4 Důsledek *Prvek a v $\mathbb{Z}_m$ má inversi právě tehdy, když platí* $\gcd(a, m) = 1$.

Inversi navíc můžeme spočítat pomocí (rozšířeného) Eukleidova algoritmu a Bezoutovy rovnosti.

3.1.5 Příklad Nalezněte inversi (pokud existuje) k 11 v $\mathbb{Z}_{19}$.

Protože $\gcd(11, 19) = 1$, víme, že 11^{-1} v $\mathbb{Z}_{19}$ existuje. Nalezení inverse je bezprostřední aplikací Bezoutovy rovnosti. Podle ní existují celá čísla α, β tak, že v $\mathbb{Z}$ platí:

$$11\alpha + 19\beta = 1$$

Použitím (rozšířeného) Eukleidova algoritmu snadno zjistíme, že $\alpha = 7$ a $\beta = -4$. V $\mathbb{Z}$ tedy platí

$$11 \cdot 7 + 19 \cdot (-4) = 1$$

Přečtením poslední rovnosti modulo 19 dostáváme

$$11 \cdot 7 = 1 \quad \text{v } \mathbb{Z}_{19}$$

a proto platí $11^{-1} = 7$ v $\mathbb{Z}_{19}$.

Připomeňme, že v tělese požadujeme, aby každý nenulový prvek měl inversi. Z důsledku 3.1.4 tedy dále plyne:

3.1.6 Důsledek *$\mathbb{Z}_m$ je těleso právě tehdy, když m je prvočíslo.*

S řešením lineárních rovnic v $\mathbb{Z}_m$ úzce souvisí i takzvané *diofantické rovnice*.

3.1.7 Příklad Máme dvoje přesýpací hodiny: na 11 minut a na 5 minut. Lze pomocí této dvojice přesýpacích hodin naměřit interval 7 minut? Pokud ano, jak? Lze popsat všechny způsoby, kterými interval 7 minut naměříme?

Danou úlohu lze zformulovat takto: nalezněte v $\mathbb{Z}$ všechna řešení následující rovnice

$$11x + 5y = 7$$

Předpokládejme totiž, že jsme nějakou dvojici (x_0, y_0), která výše uvedenou rovnici řeší, nalezli. To lze interpretovat následovně:

1. Jsou-li obě čísla x_0, y_0 kladná, potom máme x_0-krát použít 11-minutové hodiny a vzápětí použít y_0-krát 5-minutové hodiny.
2. Je-li například číslo x_0 kladné a y_0 číslo záporné, potom máme oboje hodiny spustit současně, x_0-krát použít 11-minutové hodiny a $|y_0|$-krát použít 5-minutové hodiny. Interval 7 minut bude naměřen poté, co přestaneme používat 5-minutové hodiny.

   Podobně lze uvažovat, je-li číslo x_0 záporné a číslo y_0 kladné.

Problému

$$11x + 5y = 7 \quad \text{v } \mathbb{Z}$$

říkáme *diofantická rovnice*. Protože jde o lineární problém, můžeme jej vyřešit nalezením obecného řešení homogenní rovnice a nalezením partikulárního řešení nehomogenní rovnice.

1. Homogenní rovnici $11x+5y = 0$ vyřešíme snadno: jejím řešením jsou všechny dvojice (x_h, y_h), celých čísel popsané následovně:

   $$(x_h, y_h) = (5t, -11t) \quad \text{kde } t \in \mathbb{Z}$$

   Skutečně, každá dvojice tvaru $(5t, -11t)$, kde t je celé číslo, řeší danou homogenní rovnici.

   Obráceně: pokud dvojice (x_h, y_h) řeší homogenní rovnici, potom $11x_h \equiv 0 \pmod 5$, čili $x_h \equiv 0 \pmod 5$, protože 11 a 5 jsou nesoudělná čísla. Analogicky dostáváme $y_h \equiv 0 \pmod{11}$. Víme tedy, že platí

   $$\begin{aligned} x_h &= k_x \cdot 5, \text{ pro nějaké } k_x \in \mathbb{Z} \\ y_h &= k_y \cdot 11, \text{ pro nějaké } k_y \in \mathbb{Z} \end{aligned}$$

   Dosadíme-li do homogenní rovnice, potom platí

   $$0 = 11 \cdot x_h + 5 \cdot y_h = 55 \cdot k_x + 55 \cdot k_y = 55 \cdot (k_x + k_y)$$

   Proto platí $k_x = -k_y$. Dvojice (x_h, y_h) je tedy tvaru $(5t, -11t)$, kde t je celé číslo.

   Shrnuto: obecným řešením homogenní rovnice jsou dvojice

   $$(x_h, y_h) = t \cdot (5, -11), \text{ kde } t \in \mathbb{Z}$$

2. Partikulární řešení (tj. *jakékoli* řešení) nehomogenní rovnice $11x + 5y = 7$ nalezneme pomocí Bezoutovy věty: protože $\gcd(11, 5) = 1$, existují totiž celá čísla α a β tak, že platí

   $$11\alpha + 5\beta = 1$$

   a tudíž po vynásobení celé rovnosti číslem 7 dostáváme

   $$11 \cdot (7 \cdot \alpha) + 5 \cdot (7 \cdot \beta) = 7$$

   Použitím Eukleidova algoritmu zjistíme, že $\alpha = 1$ a $\beta = -2$, tudíž partikulární řešení je dvojice

   $$(x_p, y_p) = (7, -14)$$

Celkové řešení[1] dané diofantické rovnice tedy je:

$$(x, y) = (7, -14) + t \cdot (5, -11) \quad \text{kde } t \in \mathbb{Z}$$

Položíme-li $t = 0$, dostaneme jako řešení dvojici $(7, -14)$. Toto řešení dává návod, jak změřit interval 7 minut: spusťte oboje hodiny současně a 11-ti minutové použijte sedmkrát a 5-ti minutové čtrnáckrát. Interval 7 minut pak je čas od doby, kdy jsme přestali používat 5-ti minutové hodiny. Jiný myslitelný způsob měření dává použití $t = 1$, atd.

Více o diofantických rovnicích viz cvičení 3.7.3.

3.2 Lineární algebra

V tomto odstavci se budeme věnovat především řešení soustav lineárních rovnic v $\mathbb{Z}_m$. Uvidíme, že všechny věty známé z „klasické“ lineární algebry, kdy skaláry jsou reálná čísla, se automaticky přenáší do situace, kdy skaláry jsou čísla ze $\mathbb{Z}_p$, kde p je prvočíslo. Důvodem je to, že jak $\mathbb{R}$ tak $\mathbb{Z}_p$ jsou tělesa.

Hlavním nástrojem při řešení soustav lineárních rovnic je v klasické lineární algebře Gaussova eliminace matic (GEM). Gaussova eliminace matice $\mathbb{A}$ je posloupnost elementárních řádkových úprav matice $\mathbb{A}$ na takzvaný *horní blokový tvar matice*.

Při jednotlivé elementární řádkové úpravě je dovoleno použít jedno z následujících:

1. Prohodit dva řádky matice.
2. Vynásobit řádek matice nenulovým reálným číslem.
3. Přičíst k danému řádku lineární kombinaci ostatních řádků.

V $\mathbb{Z}_p$, kde p je prvočíslo, jsou elementární řádkové úpravy definovány analogicky. Nebudeme nyní věty týkající se Gaussovy eliminace a řešení soustav lineárních rovnic v $\mathbb{Z}_p$ formulovat obecně, ale uvedeme je na příkladech.

3.2.1 Příklad V $\mathbb{Z}_5$ vyřešte soustavu rovnic:

$$\begin{array}{rcrcrcrcl} x & + & 2y & - & z & + & 2w & = & 3 \\ 4x & + & & & z & + & w & = & 2 \\ 4x & + & 4y & + & z & + & 2w & = & 2 \end{array}$$

Zapíšeme nyní *rozšířenou matici soustavy*:

$$\left(\begin{array}{cccc|c} 1 & 2 & -1 & 2 & 3 \\ 4 & 0 & 1 & 1 & 2 \\ 4 & 4 & 1 & 2 & 2 \end{array}\right)$$

a tuto matici budeme upravovat Gaussovou eliminací. To je v pořádku, protože $\mathbb{Z}_5$ je těleso. Nejprve vynulujeme první sloupec: ke druhému řádku přičteme první řádek a ke třetímu také. Dostaneme tak matici:

$$\left(\begin{array}{cccc|c} 1 & 2 & -1 & 2 & 3 \\ 0 & 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 4 & 0 \end{array}\right)$$

Nyní ke třetímu řádku přičteme dvojnásobek druhého řádku:

$$\left(\begin{array}{cccc|c} 1 & 2 & -1 & 2 & 3 \\ 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Gaussova eliminace skončila, protože jsme obdrželi horní blokový tvar matice. Nyní použijeme *Frobeniovu větu*:

> *Řešení soustavy $\mathbb{A}x = b$ existuje právě tehdy, když hodnost matice $\mathbb{A}$ je rovna hodnosti rozšířené matice soustavy.*

[1] Povšimněte si geometrie řešení: jde o „přímku“.

V našem případě je hodnost matice soustavy a hodnost rozšířené matice soustavy rovna 2. Hodnost matice je totiž definována jako počet nenulových řádků po skončení Gaussovy eliminace.

Řešení zadané soustavy tedy existuje. Protože jde o lineární problém, stačí najít partikulární řešení nehomogenní soustavy a obecné řešení homogenní soustavy. Celkové řešení je pak součtem těchto řešení.[2]

1. Partikulární řešení je *jakékoli* řešení nehomogenní soustavy:

$$\left(\begin{array}{cccc|c} 1 & 2 & -1 & 2 & 3 \\ 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

   Za partikulární řešení můžeme zvolit čtveřici $(3,0,0,0)$. Partikulární řešení je totiž jakékoli řešení, zvolme v něm proto tolik nul, kolik můžeme, a zbytek dopočítáme.

2. Obecné řešení homogenní soustavy je lineární kombinací *fundamentálního systému* soustavy. Fundamentální systém je tvořen lineárně nezávislými řešeními homogenní rovnice. Počet prvků fundamentálního systému pro obecnou soustavu $\mathbb{A}x = b$ se řídí vzorcem

$$\text{počet sloupců matice } \mathbb{A} - \text{hodnost matice } \mathbb{A}$$

   V našem případě má matice $\mathbb{A}$ čtyři sloupce a její hodnost je 2. Hledáme tedy $4-2=2$ lineárně nezávislá řešení homogenní soustavy

$$\left(\begin{array}{cccc|c} 1 & 2 & -1 & 2 & 0 \\ 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

   Můžeme postupovat metodou „organizovaného hádání“: lineární nezávislost zajistíme volbou nul a jedniček:

$$(\_,\_,1,0)$$
$$(\_,\_,0,1)$$

   a neobsazená místa dopočítáme tak, aby oba vektory řešily homogenní soustavu. Dostaneme vektory

$$(1,0,1,0)$$
$$(1,1,0,1)$$

   Obecné řešení homogenní soustavy je lieární kombinací těchto dvou vektorů

$$r\cdot(1,0,1,0) + s\cdot(1,1,0,1) \quad \text{kde } r,s\in\mathbb{Z}_5$$

Celkové řešení[3] soustavy je

$$(3,0,0,0) + r\cdot(1,0,1,0) + s\cdot(1,1,0,1) \quad \text{kde } r,s\in\mathbb{Z}_5$$

V teorii lineárních kódů (odstavec 3.3 a cvičení 3.7.7) je zapotřebí umět v $\mathbb{Z}_p$ řešit následující problém:

Pro danou matici $\mathbb{H}$ nalezněte matici $\mathbb{G}$ s maximální hodností a takovou, že součin $\mathbb{H}\cdot\mathbb{G}^\top$ je nulová matice.

Ukážeme na příkladu, že problém lze převést na řešení soustavy lineárních rovnic.

3.2.2 Příklad V $\mathbb{Z}_{11}$ je dána následující matice $\mathbb{H}$:

$$\left(\begin{array}{ccccc} 7 & 4 & 2 & 1 & 2 \\ 0 & 1 & 3 & 1 & 6 \\ 0 & 0 & 9 & 2 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

[2] Porovnejte tento postup se způsobem řešení lineárních rekurentních rovnic z odstavce 1.3.
[3] Povšimněte si geometrie řešení: jde o „rovinu“.

(Pro matici, která není v horním blokovém tvaru, bychom nejprve provedli Gaussovu eliminaci.)

Hledáme matici $\mathbb{G}$, která má maximální možnou hodnost, a takovou, že součin $\mathbb{H}\cdot\mathbb{G}^\top$ je roven nulové matici. Uvědomme si, že požadavek

$$\mathbb{H}\cdot\mathbb{G}^\top = 0$$

říká přesně to, že každý řádek matice $\mathbb{G}$ musí být řešením homogenní soustavy rovnic $\mathbb{H}x = 0$. Když tedy do jednotlivých řádků matice $\mathbb{G}$ napíšeme prvky fundamentálního systému, bude úloha vyřešena, protože potom matice $\mathbb{G}$ bude mít maximální možnou hodnost.

V našem případě má fundamentální systém dva prvky (protože $5-3=2$), například:

$$(8,7,1,1,0)$$
$$(2,1,5,0,1)$$

Proto lze jako matici $\mathbb{G}$ zvolit

$$\begin{pmatrix} 5 & 7 & 1 & 1 & 0 \\ 2 & 3 & 8 & 0 & 1 \end{pmatrix}$$

Uvědomme si, že matice $\mathbb{G}$ není určena jednoznačně, protože ani fundamentální systém není určen jednoznačně. Jednoznačně určená je pouze maximální možná hodnost matice $\mathbb{G}$.

3.2.3 Poznámka V případě, kdy m není prvočíslo, můžeme při Gaussově eliminaci v $\mathbb{Z}_m$ narazit na nečekané problémy. Následující příklad ukazuje, že v takovém $\mathbb{Z}_m$ obecně *neplatí* věta

Hodnost matice je rovna hodnosti transponované matice.

Důvodem je to, že v klasické lineární algebře se tento výsledek opírá o Gaussovu eliminační metodu.

Proto pro složená čísla m nebudeme v $\mathbb{Z}_m$ soustavy s více než jednou rovnicí řešit.

3.2.4 Příklad Ukážeme, že v $\mathbb{Z}_{30}$ má matice

$$\begin{pmatrix} 1 & 1 & -1 \\ 0 & 2 & 3 \end{pmatrix}$$

lineárně nezávislé řádky, ale každé dva sloupce jsou lineárně závislé.

1. Předpokládejme, že pro $\alpha,\beta\in\mathbb{Z}_{30}$ platí

   $$\alpha\cdot(1,1,-1)+\beta\cdot(0,2,3)=(0,0,0)$$

   V $\mathbb{Z}_{30}$ tedy musí platit soustava rovností

   $$\begin{aligned} \alpha &= 0\\ \alpha+2\beta &= 0\\ -\alpha+3\beta &= 0 \end{aligned}$$

   Protože $\alpha=0$ v $\mathbb{Z}_{30}$, zbývá v $\mathbb{Z}_{30}$ vyřešit soustavu

   $$\begin{aligned} 2\beta &= 0\\ 3\beta &= 0 \end{aligned}$$

   Podle věty 3.1.2 má první rovnice v $\mathbb{Z}_{30}$ právě dvě různá řešení, a sice čísla 0 a 15. Druhá rovnice má právě tři různá řešení: čísla 0, 10 a 20. Řešením celé soustavy je tedy pouze číslo $\beta=0$.

   Protože pak i $\alpha=0$, ukázali jsme, že řádky jsou lineárně nezávislé.

2. Napíšeme nyní pro každou dvojici sloupců netriviální lineární kombinaci, která dává nulový vektor:

   $$\begin{aligned} (0,0) &= 15\cdot(1,0) + 15\cdot(1,2)\\ (0,0) &= 10\cdot(1,0) + 10\cdot(-1,3)\\ (0,0) &= 6\cdot(1,2) + 6\cdot(-1,3) \end{aligned}$$

Další důležitou partií klasické lineární algebry je *aritmetika matic*: násobení, sčítání matic, výpočet determinantu a otázka existence inversní matice.

Násobení a sčítání matic i *rekursivní* definici determinantu (tj. rozvoj podle řádku či sloupce) lze převzít z klasické lineární algebry a pro obecné $\mathbb{Z}_m$ pro ně platí obdobná tvrzení jako v klasické lineární algebře. Jediný rozdíl je v otázce existence inversní matice. V obecném $\mathbb{Z}_m$ platí následující věta (povšimněme si ale, že pokud je $\mathbb{Z}_m$ těleso, jde o přesnou kopii klasické věty nad $\mathbb{R}$):

3.2.5 Věta *Čtvercová matice $\mathbb{A}$ nad $\mathbb{Z}_m$ má inversi právě tehdy, když $\det \mathbb{A}$ má inversi v $\mathbb{Z}_m$. Inversní matici lze potom spočítat podle následujícího vzorce*

$$\mathbb{A}^{-1} = (\det \mathbb{A})^{-1} \cdot \mathbb{D}^\top$$

kde $\mathbb{D}$ je matice algebraických doplňků matice $\mathbb{A}$.

3.2.6 Příklad Rozhodněte, zda v $\mathbb{Z}_{26}$ existuje inverse k matici

$$\mathbb{A} = \begin{pmatrix} 2 & 3 & 5 \\ 5 & 11 & 2 \\ 1 & 2 & 2 \end{pmatrix}$$

V kladném případě tuto inversi nalezněte.

1. Nejprve spočítáme determinant matice $\mathbb{A}$. Můžeme ho spočítat buď rozvojem podle některého řádku či sloupce nebo Sarrusovým pravidlem. (Pozor, matici *nemůžeme* nejprve upravit Gaussovou eliminací, protože 26 není prvočíslo!) Vyjde
$$\det \mathbb{A} = 7 \quad \text{v } \mathbb{Z}_{26}$$
Protože 7^{-1} v $\mathbb{Z}_{26}$ existuje (a sice $7^{-1} = 15$), existuje i inverse k matici $\mathbb{A}$.

2. Matici $\mathbb{A}^{-1}$ spočítáme pomocí věty 3.2.5. K tomu nejprve potřebujeme znát matici $\mathbb{D}$ algebraických doplňků.

   Matice $\mathbb{D}$ má na místě (i, j) algebraický doplněk A_{ij} prvku a_{ij} matice $\mathbb{A}$. Platí
   $$A_{ij} = (-1)^{i+j} \cdot \det \mathbb{A}_{ij}$$
   kde matice $\mathbb{A}_{ij}$ vznikla z matice $\mathbb{A}$ vynecháním i-tého řádku a j-tého sloupce.

   Pro naši matici $\mathbb{A}$ tedy jednotlivé algebraické doplňky jsou:

$$\begin{aligned}
A_{11} &= (-1)^{1+1} \cdot \begin{vmatrix} 11 & 2 \\ 2 & 2 \end{vmatrix} = 18 \\
A_{12} &= (-1)^{1+2} \cdot \begin{vmatrix} 5 & 2 \\ 1 & 2 \end{vmatrix} = 18 \\
A_{13} &= (-1)^{1+3} \cdot \begin{vmatrix} 5 & 11 \\ 1 & 2 \end{vmatrix} = 25 \\
A_{21} &= (-1)^{2+1} \cdot \begin{vmatrix} 3 & 5 \\ 2 & 2 \end{vmatrix} = 4 \\
A_{22} &= (-1)^{2+2} \cdot \begin{vmatrix} 2 & 5 \\ 1 & 2 \end{vmatrix} = 25 \\
A_{23} &= (-1)^{2+3} \cdot \begin{vmatrix} 2 & 3 \\ 1 & 2 \end{vmatrix} = 25 \\
A_{31} &= (-1)^{3+1} \cdot \begin{vmatrix} 3 & 5 \\ 11 & 2 \end{vmatrix} = 3 \\
A_{32} &= (-1)^{3+2} \cdot \begin{vmatrix} 2 & 5 \\ 5 & 2 \end{vmatrix} = 21 \\
A_{33} &= (-1)^{3+3} \cdot \begin{vmatrix} 2 & 3 \\ 5 & 11 \end{vmatrix} = 7
\end{aligned}$$

Matice $\mathbb{D}$ je tedy

$$\begin{pmatrix} 18 & 18 & 25 \\ 4 & 25 & 25 \\ 3 & 21 & 7 \end{pmatrix} = \begin{pmatrix} -8 & -8 & -1 \\ 4 & -1 & -1 \\ 3 & -5 & 7 \end{pmatrix}$$

Matici $\mathbb{D}^\top$ budeme násobit skalárem, proto jsme $\mathbb{D}$ upravili tak, že obsahuje čísla s nejmenší absolutní hodnotou.

Podle vzorce

$$\mathbb{A}^{-1} = (\det \mathbb{A})^{-1} \cdot \mathbb{D}^\top$$

je tedy

$$\mathbb{A}^{-1} = 15 \cdot \begin{pmatrix} -8 & -8 & -1 \\ 4 & -1 & -1 \\ 3 & -5 & 7 \end{pmatrix}^\top = 15 \cdot \begin{pmatrix} -8 & 4 & 3 \\ -8 & -1 & -5 \\ -1 & -1 & 7 \end{pmatrix} = \begin{pmatrix} 10 & 8 & 19 \\ 10 & 11 & 3 \\ 11 & 11 & 1 \end{pmatrix}$$

3.3 Aplikace — lineární kódy

V tomto odstavci naznačíme aplikaci lineární algebry nad $\mathbb{Z}_p$ v teorii lineárních kódů, které jsou schopny opravovat chyby.[4] Zprávy (půjde o prvky konečně dimenzionálního vektorového prostoru, tedy o uspořádané n-tice) hodláme posílat kanálem, který může do zpráv zanést chyby. Problémem je rekonstrukce původní zprávy.

Připomeňme, že množina $(\mathbb{Z}_p)^n$ všech uspořádaných n-tic prvků $\mathbb{Z}_p$ tvoří vektorový prostor dimense n nad tělesem $\mathbb{Z}_p$. Vektorový podprostor V dimense k nazveme *lineárním p-kódem dimense k a délky n*. Prvkům V budeme říkat *kódová slova*.

Základní vlastnosti lineárního kódu jsou:

1. $(0, \ldots, 0)$ (nulový vektor) je vždy kódové slovo.
2. Libovolná lineární kombinace kódových slov je kódové slovo.
3. Existuje k vektorů $g_1, \ldots, g_k$ s vlastností: *každé* kódové slovo lze napsat jednoznačně jako lineární kombinaci vektorů $g_1, \ldots, g_k$:
$$u = a_1 \cdot g_1 + a_2 \cdot g_2 + \ldots + a_k \cdot g_k \tag{3.1}$$

Výše uvedená fakta jsou pochopitelně jen přeformulováním toho, že V je vektorový podprostor dimense k, systém $g_1, \ldots, g_k$ je výběr báze prostoru V. Pokud tuto bázi napíšeme do řádků matice $\mathbb{G}$, vytvoříme *generující matici*, která má k řádků, n sloupců a hodnost k. Generující matici využíváme k *přeměně* zprávy (nesoucí informaci) v kódové slovo (nesoucí redundantní informaci, která nám dovoluje detekovat a opravovat chyby).

3.3.1 Poznámka Pro pochopení základních pojmů teorie lineárních kódů může pomoci následující *geometrická* představa.

Zvolme v $\mathbb{R}^3$ rovinu danou rovnicí $x + y + z = 0$. Dobře víme, že tato rovina prochází počátkem souřadnic a tvoří vektorový podprostor V dimense 2 v $\mathbb{R}^3$.

1. Známe-li *bázi* g_1, g_2 prostoru V, můžeme *generovat* vektory v $\mathbb{R}^3$, které *leží* ve V, následujícím způsobem: zvolíme si souřadnice (a_1, a_2) vzhledem ke g_1, g_2 a spočítáme $a_1 \cdot g_1 + a_2 \cdot g_2$.

   Zvolme jako bázi roviny $x + y + z = 0$ vektory $g_1 = (2, 2, -4)$ a $g_2 = (2, -1, -1)$. Potom pro souřadnice $(a_1, a_2) = (3, 4)$ *vygenerujeme* vektor $3 \cdot (2, 2, -4) + 4 \cdot (2, -1, -1) = (14, 2, -16)$, o kterém bezpečně víme, že leží v podprostoru V.

   Povšimněme si ještě, že generování lze zapsat maticově:

$$\begin{pmatrix} 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 2 & 2 & -4 \\ 2 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 14 & 2 & -16 \end{pmatrix}$$

   a že tudíž můžeme chápat vektor souřadnic $(3, 4)$ jakožto *informaci*, kterou jsme *zakódovali* pomocí *generující matice* na *kódové slovo* $(14, 2, -16)$, které nese *redundantní informaci*.

   Všimněme si ještě, jak generující matice vypadá: v řádcích má napsán *fundamentální systém* rovnice $x + y + z = 0$. Srovnejte s příkladem 3.2.2 a cvičením 3.7.7.

[4] Zájemce o podrobnosti odkazujeme např. na knihu J. Adámek, *Kódování*, SNTL, Praha, 1989.

2. Máme-li jakýkoli vektor v z $\mathbb{R}^3$, je snadné otestovat, zda $v \in V$. Stačí totiž tento vektor dosadit do výrazu $x + y + z$ a testovat, zda vyjde 0.

   Předpokládejme, že chceme testovat, zda $(2, 3, 1) \in V$. Dosazení $(2, 3, 1)$ do $x + y + z$ lze zapsat maticově:

$$\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} = 6$$

   Výsledné matici (v tomto případě rozměrů 1×1, čili číslu) říkáme *syndrom* slova $(2, 3, 1)$.

   Zjevně platí, že slovo leží ve V právě tehdy, když má nulový syndrom. Protože matici $(1, 1, 1)$ využíváme ke kontrole toho, zda slovo je kódovým slovem, říkáme jí *kontrolní matice*.

   Ještě si povšimněme, že kontrolní matice je *normálový vektor* roviny $x + y + z = 0$.

V dalším budeme provádět totožné úvahy. Prostor $\mathbb{R}^3$ však nahradíme prostorem $(\mathbb{Z}_p)^n$. Přesto jsou výše uvedené úvahy typické:

1. Generující matice $\mathbb{G}$ kódu V (V je podprostor $(\mathbb{Z}_p)^n$ dimense k) má v řádcích vektory báze prostoru V. To znamená, že $\mathbb{G}$ má k řádků a n sloupců a hodnost k.

2. Kontrolní matice $\mathbb{H}$ kódu V (V je podprostor $(\mathbb{Z}_p)^n$ dimense k) má v řádcích fundamentální systém rovnice $\mathbb{G} \cdot x^\top = 0$. To znamená, že každý řádek matice $\mathbb{H}$ je *kolmý* na všechny řádky matice $\mathbb{G}$.

   Geometricky řečeno, matice $\mathbb{H}$ zobecňuje pojem normálového vektoru. Vektorovému prostoru, který je generován řádky matice $\mathbb{H}$, se říká *ortogonální doplněk* vektorového prostoru V (a často se značí $V^\perp$).

   Matice $\mathbb{H}$ má $n - k$ řádků a n sloupců a hodnost $n - k$.

3.3.2 Příklad V tomto příkladu ukážeme příklad dobře známého *lineárního kódu ISBN.*[5]
Kód ISBN je deseticiferné číslo $a_1 \dots a_{10}$ v soustavě o základu 11, které splňuje rovnost

$$\sum_{i=1}^{10} i \cdot a_i = 0 \quad \text{v } \mathbb{Z}_{11} \tag{3.2}$$

Cifry v soustavě o základu 11 jsou znaky 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, X.

V praxi je toto desetiferné číslo rozděleno pomlčkami do čtyř skupin: jazyk knihy, číslo nakladatele, číslo knihy a *kontrolní znak*.

Kontrolní znak využíváme k tomu, abychom *zkontrolovali*, zda máme před sebou platný kód ISBN.

Například číslo

$$0\text{-}74X2\text{-}7560\text{-}8$$

není platný kód ISBN, protože platí

$$1 \cdot 0 + 2 \cdot 7 + 3 \cdot 4 + 4 \cdot 10 + 5 \cdot 2 + 6 \cdot 7 + 7 \cdot 5 + 8 \cdot 6 + 9 \cdot 0 + 10 \cdot 8 = 5 \neq 0 \quad \text{v } \mathbb{Z}_{11}$$

Podobnou kontrolu lze provést pro každé deseticiferné číslo v $\mathbb{Z}_{11}$. Zapsáno maticově, testujeme nulovost *syndromu*

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \end{pmatrix} = s \tag{3.3}$$

[5] ISBN znamená *International Standard Book Number*. V současnosti se rozmáhá nový standard: 13-ISBN.

čísla $a_1 \dots a_{10}$. Matici $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ říkáme *kontrolní matice* kódu ISBN, protože ji využíváme ke kontrole. Kontrolní matici budeme značit $\mathbb{H}$.

Jak spočítat „kontrolní bit“ (tj. poslední cifru) kódu ISBN, pokud máme k dispozici údaje o jazyku, nakladateli a čísle knihy? Z rovnosti (3.2) dostáváme

$$a_{10} = \sum_{i=1}^{9} i \cdot a_i \quad \text{v } \mathbb{Z}_{11}$$

protože $-10 = 1$ v $\mathbb{Z}_{11}$. Tvorbu celého ISBN můžeme zapsat maticově takto:

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 9 \end{pmatrix} =$$

$$= \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \end{pmatrix}$$

Matici rozměrů 9×10, která v této rovnici vystupuje, budeme říkat *generující matice* kódu ISBN a budeme ji značit $\mathbb{G}$.

Kód ISBN je tedy lineární 11-kód dimense 9 a délky 10: pracujeme totiž v $\mathbb{Z}_{11}$, kódujeme vektory délky 9 a výsledný kód je vektor délky 10.

Podívejme se, jak moc je kód ISBN schopen *detekovat* a *opravovat* chyby. Vypůjčíme si postavy Alice a Boba z odstavce 3.5 a budeme si představovat následující scénář:

1. Alice v knihkupectví opsala na kus papíru ISBN nějaké knihy.
2. Alice tento kus papíru předala Bobovi.
3. Bob má zjistit (detekovat), zda Alice neudělala při opisování ISBN chybu nebo chyby. Pokud ano, má se tyto chyby pokusit opravit.

Bob postupuje takto:

1. Bob spočítá syndrom s slova $a_1 \dots a_{10}$ podle vzorce (3.3).
2. Jestliže vyjde $s = 0$, je na papíře napsáno platné ISBN.

   Pozor! To neznamená, že Alice neudělala při opisování chybu: mohla jich udělat tolik, že to ze syndromu nejde poznat. Pokud například Alice v platném kódu ISBN 0-349-10571-5 udělá chyby na druhém a čtvrtém místě a napíše na papír 0-14X-10571-5, vznikne opět platný kód ISBN a Bob tyto chyby nepozná.
3. Jestliže vyjde $s \neq 0$, došlo při opisování alespoň k jedné chybě.

   Pokud je Alice spolehlivá a při opisování udělá vždy *nanejvýš jednu chybu*, může Bob chybu *detekovat*. Detekce však znamená pouze zjištění, že došlo k chybě.

   Lze zjistit, na které pozici k chybě došlo?

   Pokud k chybě došlo na j-té pozici, musí být na papíře zapsáno číslo

   $$(a_1, \dots, a_{j-1}, a_j + b, a_{j+1}, \dots, a_{10})$$

   tj., k původnímu kódu ISBN $a = (a_1, \dots, a_{10})$ byl (v $\mathbb{Z}_{11}$) přičteno *chybové slovo* $e = (0, \dots, 0, b, 0 \dots, 0)$, mající na j-tém místě b a jinde nuly. Syndrom tedy musí být

   $$s = \mathbb{H} \cdot (a + e)^\top = \mathbb{H} \cdot a^\top + \mathbb{H} \cdot e^\top = \mathbb{H} \cdot e^\top = j \cdot b \quad \text{v } \mathbb{Z}_{11}$$

Umí Bob ze znalosti s zrekonstruovat j a b? Pokud ano, znamenalo by to, že je schopen zjistit pozici j, na které k chybě došlo, a odečtením b na této pozici pak kód opravit.

Taková rekonstrukce ale obecně není možná: pro syndrom 6 může být $j = 2$ a $b = 3$ nebo $j = 3$ a $b = 2$.

Kód ISBN tedy nemá příliš dobré vlastnosti detekce a opravování chyb: zhruba řečeno je to proto, že má příliš dlouhou informační část (devět „informačních bitů“) a příliš krátkou kontrolní část (jeden „kontrolní bit“). Viz také cvičení 3.7.6.

Úvahy o kódu ISBN byly typické příklady problémů v teorii lineárních kódů. My se v dalším omezíme na *binární kódy*, tj. budeme pracovat nad $\mathbb{Z}_2$. Použití generující matice ukážeme na příkladu:

3.3.3 Příklad Předpokládejme, že V je podprostor $(\mathbb{Z}_2)^7$ dimense 4, který má bázi

$$g_1 = (1,0,0,0,0,1,1) \quad g_2 = (0,1,0,0,1,0,1) \quad g_3 = (0,0,1,0,1,1,0) \quad g_4 = (0,0,0,1,1,1,1)$$

Generující matice je tudíž matice

$$\mathbb{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Protože každé kódové slovo lze jednoznačně vyjádřit jako lineární kombinaci

$$u = a_1 \cdot g_1 + a_2 \cdot g_2 + a_3 \cdot g_3 + a_4 \cdot g_4$$

můžeme si čtveřici (a_1, a_2, a_3, a_4) představit jako *informaci*, kterou chceme poslat a příslušnou lineární kombinaci $a_1 \cdot g_1 + a_2 \cdot g_2 + a_3 \cdot g_3 + a_4 \cdot g_4$ jako *zprávu*, kterou skutečně pošleme. Povšimněte si, že díky našemu výběru báze má zpráva následující tvar:

$$(a_1, a_2, a_3, a_4, a_2 + a_3 + a_4, a_1 + a_3 + a_4, a_1 + a_2 + a_4)$$

V této sedmici tedy o prvních čtyřech položkách uvažujeme jako o *informačních bitech*, zbytek jsou *kontrolní bity*, které informaci chrání.

Generující matice z předchozího příkladu měla tvar $\mathbb{G} = (\mathbb{E} \mid \mathbb{B})$, kde $\mathbb{E}$ je jednotková matice rozměrů 4×4. Lineárním kódům s generující maticí tvaru $\mathbb{G} = (\mathbb{E} \mid \mathbb{B})$ se říká *systematické*. V systematickém kódu můžeme okamžitě odlišit informační bity od kontrolních. Pro další výhodu systematických kódů viz tvrzení 3.3.4.

Jak jsme viděli, je matice $\mathbb{G}$ užitečná na začátku přenosu. Při příjmu však čelíme otázce, zda přijaté slovo je totéž slovo, které bylo odesláno. K tomu je vhodná matice $\mathbb{H}$ (říkáme jí *kontrolní matice*) s následující vlastností:

u je prvkem V právě tehdy, když platí $Hu^\top = 0$.

Pro obecný vztah matic $\mathbb{G}$ a $\mathbb{H}$ odkazujeme na příklad 3.2.2 a cvičení 3.7.7. V případě systematického kódu je však tento vztah velmi jednoduchý.

3.3.4 Tvrzení *Ať p je prvočíslo a ať jsou všechny následující matice nad $\mathbb{Z}_p$. Má-li matice $\mathbb{G}$ tvar $(\mathbb{E} \mid \mathbb{B})$, pak $\mathbb{H}$ má tvar $(-\mathbb{B}^\top \mid \mathbb{E}')$, kde $\mathbb{E}'$ je jednotková matice příslušných rozměrů.*

Důkaz. Tvrzení je jednoduchým důsledkem násobení blokových matic:

$$(-\mathbb{B}^\top \mid \mathbb{E}') \cdot (\mathbb{E} \mid \mathbb{B})^\top = -\mathbb{B}^\top + \mathbb{B}^\top = 0$$

■

3.3.5 Příklad Kontrolní matice kódu z příkladu 3.3.3 je (například) následující matice:

$$\mathbb{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Protože šlo o systematický kód, mohli jsme samozřejmě použít tvrzení 3.3.4. To je opět ukázka toho, že ani matice $\mathbb{H}$ není určena jednoznačně.

Ukážeme, že náš kód je schopen opravit jednu chybu. Jde proto o ideální kód, pokud předpokládáme, že zprávy posíláme kanálem, který v poslané zprávě poruší maximálně jednu pozici.

Předpokládejme, že odesíláme zprávu u a přijmeme v. Rozdílu $e = v - u$ říkáme *chybové slovo*. Umíme-li chybové slovo spočítat, provedeme korekci $u = v - e$ a tak získáme původní zprávu. Pro zjištění chybového slova nejprve spočteme *syndrom* $\mathbb{H} \cdot v^\top$ přijatého slova v:

$$\mathbb{H} \cdot v^\top = \mathbb{H} \cdot (u+e)^\top = \mathbb{H} \cdot u^\top + \mathbb{H} \cdot e^\top = \mathbb{H} \cdot e^\top$$

Poslední z výše uvedených rovností plyne z toho, že u je kódové slovo a proto platí rovnost $\mathbb{H} \cdot u^\top = 0$. Víme-li navíc, že k chybě došlo nanejvýš v jednom bitu, obsahuje chybové slovo e nanejvýš jeden symbol 1, zbytek chybového slova musí být tvořen znaky 0. Pokud ovšem takovým chybovým slovem vynásobíme matici $\mathbb{H}$, vybereme tak i-tý sloupec matice $\mathbb{H}$ právě tehdy, když je symbol 1 v i-té souřadnici slova e. Příklad

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Protože sloupce matice $\mathbb{H}$ jsou navzájem různé, můžeme při použití tohoto kódu opravit jednu chybu.

Pro ilustraci předpokládejme, že jsme odeslali $u = (1,1,0,1,1,1,1)$ a přijali $v = (1,1,1,1,1,1,1)$. Syndrom slova v je:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

což je třetí sloupec matice $\mathbb{H}$. Abychom zrestaurovali původní zprávu, musíme opravit třetí bit slova v.

3.4 Hlubší výsledky z počítání modulo

V tomto odstavci zformulujeme některé hlubší poznatky o $\mathbb{Z}_m$, které se využívají například v algoritmech pro zpracování velkých čísel nebo v teorii šifrování, viz například cvičení 3.7.12 a odstavec 3.5.

3.4.1 Věta (Čínská věta o zbytcích)[6]

Předpokládejme, že $m_1, m_2, \ldots, m_r$ jsou navzájem nesoudělná přirozená čísla, $m_i \geq 2$ pro $i = 1, \ldots, r$. Potom každá soustava rovnic

$$\begin{aligned} x &= a_1 \quad \text{v } \mathbb{Z}_{m_1} \\ x &= a_2 \quad \text{v } \mathbb{Z}_{m_2} \\ &\vdots \\ x &= a_r \quad \text{v } \mathbb{Z}_{m_r} \end{aligned} \tag{3.4}$$

má řešení a toto řešení je určeno jednoznačně v $\mathbb{Z}_M$, kde $M = m_1 \cdot m_2 \cdot \ldots \cdot m_r$.

[6] Nejstarší zmínkou o této větě je problém 26 z knihy *Sun Tzu Suan Ching*, kterou ve třetím století našeho letopočtu napsal *Sun Zi*. Problému 26 se též říká *problém Mistra Suna* a Vy jej máte možnost vychutnat ve cvičení 3.7.9.

Důkaz. Nejprve ukážeme jednoznačnost řešení. Předpokládejme, že čísla x' a x'' řeší danou soustavu a označme $x = x' - x''$. Chceme ukázat, že $x = 0$ v $\mathbb{Z}_M$. To ale plyne z toho, že $x = 0$ v $\mathbb{Z}_{m_i}$ pro všechna i a z toho, že čísla $m_1, m_2, \ldots, m_r$ jsou navzájem nesoudělná.

Nyní ukážeme, jak řešení nalézt. Pro každé $j = 1, \ldots, r$ označme jako M_j podíl M/m_j. Číslo M_j je tedy součinem všech čísel $m_1, \ldots, m_r$ *kromě* čísla m_j a proto platí

$$\gcd(m_j, M_j) = 1$$

Označme jako N_j inversi čísla M_j v $\mathbb{Z}_{m_j}$ a definujme

$$x = \sum_{i=1}^{r} a_i M_i N_i$$

Potom v $\mathbb{Z}_{m_j}$ platí následující rovnosti

$$\begin{aligned} x &= \sum_{i=1}^{r} a_i M_i N_i \\ &= a_j M_j N_j \quad \text{(protože } m_j \mid M_i \text{ pro } i \neq j\text{)} \\ &= a_j \end{aligned}$$

a to jsme chtěli dokázat. ■

3.4.2 Příklad Důkaz čínské věty o zbytcích dává návod, jak příslušné soustavy řešit. Ukážeme na příkladu, že tento důkaz není tak komplikovaný, jak se na první pohled zdá.

Vyřešme soustavu

$$x = 2 \text{ v } \mathbb{Z}_5 \qquad x = 4 \text{ v } \mathbb{Z}_7 \qquad x = 3 \text{ v } \mathbb{Z}_{11} \tag{3.5}$$

čínskou větou o zbytcích. Předpokládejme, že řešení je ve tvaru

$$x = 2 \cdot \boxed{} + 4 \cdot \boxed{} + 3 \cdot \boxed{}$$

kde zbývá doplnit jednotlivé obdélníky. Má-li být splněna soustava (3.5), musí být

1. První obdélník roven 1 v $\mathbb{Z}_5$ a roven 0 v $\mathbb{Z}_7$ a v $\mathbb{Z}_{11}$.
2. Druhý obdélník roven 1 v $\mathbb{Z}_7$ a roven 0 v $\mathbb{Z}_5$ a v $\mathbb{Z}_{11}$.
3. Třetí obdélník roven 1 v $\mathbb{Z}_{11}$ a roven 0 v $\mathbb{Z}_5$ a v $\mathbb{Z}_7$.

Nejprve zařídíme nulovost obdélníků:

$$x = 2 \cdot \boxed{7 \cdot 11} + 4 \cdot \boxed{5 \cdot 11} + 3 \cdot \boxed{5 \cdot 7}$$

(Všimněte si, že zatím jsou v obdélnících vepsána — ve značení důkazu věty 3.4.1 — čísla M_1, M_2 a M_3.) Ovšem první obdélník je v $\mathbb{Z}_5$ roven číslu $7 \cdot 11 = 2$, druhý obdélník je v $\mathbb{Z}_7$ roven číslu $5 \cdot 11 = 6$ a třetí obdélník je v $\mathbb{Z}_{11}$ roven číslu $5 \cdot 7 = 2$.

To, aby obdélníky byly rovny jedničce, zařídíme vynásobením příslušnými inversemi: do prvního obdélníka přinásobíme číslo $2^{-1} = 3$ v $\mathbb{Z}_5$, do druhého obdélníka přinásobíme číslo $6^{-1} = 6$ v $\mathbb{Z}_7$ a do třetího obdélníka přinásobíme číslo $2^{-1} = 6$ v $\mathbb{Z}_{11}$:

$$x = 2 \cdot \boxed{7 \cdot 11 \cdot 3} + 4 \cdot \boxed{5 \cdot 11 \cdot 6} + 3 \cdot \boxed{5 \cdot 7 \cdot 6}$$

(Ve značení důkazu věty 3.4.1 jsme do obdélníků přinásobili čísla N_1, N_2 a N_3.)

Nyní jen zbývá přečíst x v $\mathbb{Z}_{5 \cdot 7 \cdot 11}$:

$$x = 2 \cdot 7 \cdot 11 \cdot 3 + 4 \cdot 5 \cdot 11 \cdot 6 + 3 \cdot 5 \cdot 7 \cdot 6 = 2\,412 = 102 \quad \text{v } \mathbb{Z}_{385}$$

Pro jiný důkaz (a tím i jiný postup řešení) čínské věty o zbytcích viz cvičení 3.7.11.

Bezprostřední aplikací čínské věty o zbytcích je algoritmus pro rychlé výpočty s velkými čísly. Ukážeme jej na příkladu.[7]

3.4.3 Příklad Pro realističtější podobu následujících úvah odkazujeme na cvičení 3.7.12.

V dalším použijeme stejné značení jako v čínské větě o zbytcích. Zvolme $m_1 = 3$, $m_2 = 5$, $m_3 = 7$, $m_4 = 11$. Potom $M = m_1 \cdot m_2 \cdot m_3 \cdot m_4 = 1155$. Pro číslo A označme po řadě A_1, A_2, A_3, A_4 zbytky po dělení čísla A po řadě čísly m_1, m_2, m_3 a m_4. Z čínské věty o zbytcích okamžitě vyplývá, že je-li číslo A v rozmezí od 0 do $M - 1 = 1154$, je korespondence

$$A \mapsto (A_1, A_2, A_3, A_4)$$

vzájemně jednoznačná. Pokud tedy výsledky násobení a sčítání čísel budou v rozmezí od 0 do $M - 1 = 1154$, dává čínská věta o zbytcích možnost rychlého výpočtu součtů a součinů. Předvedeme to na příkladu:

1. Pro naši volbu čísel m_1, m_2, m_3 a m_4 spočítáme

$$M_1 = M/m_1 = 1155/3 = 385 \quad M_2 = M/m_2 = 1155/5 = 231$$
$$M_3 = M/m_3 = 1155/7 = 165 \quad M_4 = M/m_4 = 1155/11 = 105$$

   a jako N_j označíme inversi čísla M_j v $\mathbb{Z}_{m_j}$:

$$N_1 = 1 \quad N_2 = 1 \quad N_3 = 2 \quad N_4 = 2$$

2. Chceme vynásobit čísla $X = 35$ a $Y = 27$. Nejprve obě čísla zakódujeme jako čtveřice zbytků po dělení:

$$X = 35 \quad \mapsto \quad (X_1, X_2, X_3, X_4) = (2, 0, 0, 2)$$
$$Y = 27 \quad \mapsto \quad (Y_1, Y_2, Y_3, Y_4) = (0, 2, 6, 5)$$

   Nyní složky vektorů (X_1, X_2, X_3, X_4) a (Y_1, Y_2, Y_3, Y_4) vynásobíme v příslušných $\mathbb{Z}_{m_j}$. Dostaneme tak vektor

$$(Z_1, Z_2, Z_3, Z_4) = (0, 0, 0, 10)$$

   Podle čínské věty o zbytcích tomuto vektoru odpovídá číslo

$$Z = \sum_{i=1}^{4} Z_i M_i N_i = 10 \cdot M_4 \cdot N_4 = 10 \cdot 105 \cdot 2 = 2100 = 945 \quad \text{v } \mathbb{Z}_{1155}$$

   Součinem čísel X a Y tedy je číslo 945.

Pro jinou aplikaci čínské věty o zbytcích viz cvičení 3.7.13.

V některých aplikacích (například při útoku na RSA při stejném malém veřejném exponentu 3.6.6) je zapotřebí řešit soustavy tvaru (3.4) v případě, kdy čísla $m_1, \ldots, m_r$ *nejsou* navzájem nesoudělná. Čínskou větu o zbytcích tedy zobecníme pro řešitelnost obecné soustavy tvaru (3.4). Výsledný algoritmus bude rekursivní, viz příklad 3.4.6.

Nejprve zformulujeme, jak řešit soustavy dvou rovnic.

3.4.4 Lemma *Předpokládejme, že m_1 a m_2 jsou libovolná přirozená čísla, $m_i \geq 2$ pro $i = 1, 2$. Označme $d = \gcd(m_1, m_2)$. Pak jsou následující dvě podmínky ekvivalentní:*

1. *Soustava*

$$\begin{aligned} x &= a_1 \quad \text{v } \mathbb{Z}_{m_1} \\ x &= a_2 \quad \text{v } \mathbb{Z}_{m_2} \end{aligned} \tag{3.6}$$

   má řešení.

2. *Platí $d \mid (a_2 - a_1)$.*

[7] Podrobněji se lze o rychlosti tohoto algoritmu dočíst například v knize L. Kučera a J. Nešetřil, *Algebraické metody diskrétní matematiky*, SNTL, Praha, 1989.

Jestliže platí $d \mid (a_2 - a_1)$, *je řešení určeno jednoznačně v* $\mathbb{Z}_M$, *kde* $M = \operatorname{lcm}(m_1, m_2)$. [8]

Důkaz. Podmínky 1. a 2. jsou ekvivalentní: Soustava (3.6) má řešení u právě tehdy, když v $\mathbb{Z}$ platí rovnosti $u = a_1 + m_1 k$ a $u = a_2 + m_2 l$ pro nějaká celá čísla k, l. To nastává právě tehdy, když platí v $\mathbb{Z}$ rovnost $m_1 k + m_2(-l) = a_2 - a_1$, neboli právě tehdy, když má diofantická rovnice $m_1 x + m_2 y = a_2 - a_1$ řešení. Podle cvičení 3.7.3 má tato diofantická rovnice řešení právě tehdy, když platí $d \mid (a_2 - a_1)$.

Předpokládejme, že platí podmínka $d \mid (a_2 - a_1)$. Každé řešení rovnice $x = a_1$ v $\mathbb{Z}_{m_1}$ je tvaru $u = a_1 + m_1 k$ pro celé číslo k. Určíme nyní hodnotu k tak, aby platila rovnice $x = a_2$, neboli $a_1 + m_1 k = a_2$ v $\mathbb{Z}_{m_2}$. Takových čísel k existuje v $\mathbb{Z}_{m_2}$ přesně d, protože lineární rovnice $m_1 k = (a_2 - a_1)$ má podle věty 3.1.2 v $\mathbb{Z}_{m_2}$ přesně d různých řešení. Tudíž lineární rovnice

$$\frac{m_1}{d} k = \frac{a_2 - a_1}{d} \quad \text{v } \mathbb{Z}_{\frac{m_2}{d}}$$

má jediné řešení k, protože $\gcd(\frac{m_1}{d}, \frac{m_2}{d}) = 1$. Toto jediné řešení je tedy tvaru $k = k_0 + l\frac{m_2}{d}$ pro celá čísla l. Dosadíme nyní toto k do rovnice pro u:

$$u = a_1 + m_1 k = a_1 + m_1(k_0 + l\frac{m_2}{d}) = a_1 + m_1 k_0 + l\frac{m_1 m_2}{d} = a_1 + m_1 k_0 + lM$$

protože $\frac{m_1 m_2}{d}$ je nejmenší společný násobek čísel m_1 a m_2. Nalezli jsme jediné řešení soustavy (3.6). ■

3.4.5 Příklad Metodou předchozího lemmatu vyřešíme soustavu

$$\begin{aligned} x &= 21 \quad \text{v } \mathbb{Z}_{33} \\ x &= 43 \quad \text{v } \mathbb{Z}_{55} \end{aligned}$$

Především $d = \gcd(33, 55) = 11$ a platí $11 \mid (43 - 21)$, podle lemmatu 3.4.4 tedy řešení u naší soustavy existuje jednoznačně v $\mathbb{Z}_M$, kde $M = \operatorname{lcm}(33, 55) = 165$.

Protože musí platit $u = 21 + 33k$ v $\mathbb{Z}$, určíme číslo k tak, aby platilo $21 + 33k = 43$ v $\mathbb{Z}_{55}$. Řešíme tedy rovnici $33k = 43 - 21 = 22$ v $\mathbb{Z}_{55}$ (ta má podle věty 3.1.2 přesně 11 různých řešení), neboli řešíme rovnici $3k = 2$ v $\mathbb{Z}_5$. Toto jediné řešení je $k = 4$.

Celkově je $u = 21 + 33 \cdot 4 = 153$ v $\mathbb{Z}_{165}$.

Nyní na příkladu vysvětlíme, jak řešit obecné soustavy tvaru (3.4) rekursivně. Pro obecnou podmínku řešitelnosti soustavy tvaru (3.4) viz cvičení 3.7.14.

3.4.6 Příklad Připomeňme, že z příkladu 3.4.5 víme, jak řešit soustavy o dvou rovnicích. Předvedeme, jak řešit soustavy o třech rovnicích, pro větší soustavy budeme postupovat analogicky.

1. Vyřešíme soustavu

$$\begin{aligned} x &= 1 \quad \text{v } \mathbb{Z}_{12} \\ x &= 7 \quad \text{v } \mathbb{Z}_{14} \\ x &= 4 \quad \text{v } \mathbb{Z}_{15} \end{aligned}$$

   Nejprve vyřešíme soustavu prvních dvou rovnic metodou příkladu 3.4.5. To lze, protože platí $\gcd(12, 14) = 2$ a $2 \mid (7 - 1)$. První dvě rovnice tedy jednoznačně určují $x = 49$ v $\mathbb{Z}_{84}$.

   Původní soustavu tedy nahradíme soustavou *dvou* rovnic

$$\begin{aligned} x &= 49 \quad \text{v } \mathbb{Z}_{84} \\ x &= 4 \quad \text{v } \mathbb{Z}_{15} \end{aligned}$$

   a tu opět vyřešíme metodou příkladu 3.4.5. To lze, protože platí $\gcd(84, 15) = 3$ a $3 \mid (4 - 49)$. Řešení je $x = 49$ v $\mathbb{Z}_{420}$.

   Číslo $x = 49$ v $\mathbb{Z}_{420}$ je také jediné řešení původní soustavy tří rovnic.

[8] $\operatorname{lcm}(a, b)$ značí *nejmenší společný násobek* celých čísel a, b, anglicky *least common multiple*.

2. Vyřešíme soustavu

$$\begin{aligned} x &= 5 \quad \text{v } \mathbb{Z}_{12} \\ x &= 17 \quad \text{v } \mathbb{Z}_{18} \\ x &= 3 \quad \text{v } \mathbb{Z}_{40} \end{aligned}$$

Nejprve vyřešíme soustavu prvních dvou rovnic metodou příkladu 3.4.5. To lze, protože platí $\gcd(12,18) = 6$ a $6 \mid (17-5)$. První dvě rovnice tedy jednoznačně určují $x = 17$ v $\mathbb{Z}_{36}$.

Původní soustavu tedy nahradíme soustavou *dvou* rovnic

$$\begin{aligned} x &= 17 \quad \text{v } \mathbb{Z}_{36} \\ x &= 3 \quad \text{v } \mathbb{Z}_{40} \end{aligned}$$

a tu opět vyřešíme metodou příkladu 3.4.5. To ale nelze, protože platí $\gcd(36,40) = 4$, ale $4 \mid (3-17)$ *neplatí*. Řešení původní soustavy *neexistuje*.

Viz také cvičení 3.7.14.

Další řada výsledků nám umožní efektivně zpracovávat velké mocniny.

3.4.7 Věta (Malá Fermatova věta) *Ať p je prvočíslo. Potom pro libovolné celé číslo a nesoudělné s p platí*

$$a^{p-1} = 1 \quad \text{v } \mathbb{Z}_p$$

Důkaz. Nejprve ukážeme, že jsou-li i a j dvě různá čísla z množiny $\{0,1,\ldots,(p-1)\}$, potom

$$ia \neq ja \quad \text{v } \mathbb{Z}_p$$

Kdyby totiž nastala rovnost, znamenalo by to, že platí $p \mid a(i-j)$ a protože předpokládáme, že $\gcd(a,p) = 1$, musí platit $p \mid (i-j)$. To je spor.

Ukázali jsme tedy, že posloupnosti

$$1a, 2a, \ldots, (p-1)a \quad \text{a} \quad 1, \ldots, (p-1)$$

musí v $\mathbb{Z}_p$ obsahovat (až na pořadí) stejná čísla. Součiny všech členů obou posloupností v $\mathbb{Z}_p$ jsou si tedy v $\mathbb{Z}_p$ rovny. To znamená, že platí

$$a^{p-1}(p-1)! = (p-1)! \quad \text{v } \mathbb{Z}_p$$

Protože p je prvočíslo, je $\gcd(p,(p-1)!) = 1$, a tudíž

$$a^{p-1} = 1 \quad \text{v } \mathbb{Z}_p$$

■

3.4.8 Poznámka Existuje ještě *velká* (také *poslední*) Fermatova věta. Jde o následující tvrzení:

Pro libovolné pevné přirozené číslo $n \geq 3$ nemá rovnice $a^n + b^n = c^n$ řešení v kladných přirozených číslech.

Srovnejte výše uvedené tvrzení se známou *Pythagorovou větou*, která zaručuje existenci *nekonečně mnoha* trojic přirozených čísel a, b, c, splňujících rovnost $a^2 + b^2 = c^2$ — viz příklad 3.4.9.

Pierre de Fermat (1601–1665) byl francouzský právník, který se matematikou zabýval pouze rekreačně. Přesto je autorem řady pozoruhodných matematických výsledků. Své důkazy však psal velmi nepořádně, většinou na prázdné okraje knih, které právě studoval. Svým kolegům matematikům pak posílal pouze znění tvrzení bez důkazů.[9] Proto se ve Fermatově pozůstalosti na okraji Diophantovy[10] knihy *Arithmetica*, pod zněním velké Fermatovy věty, zachovala známá okřídlená formulace

[9] Fermatův současník, anglický matematik John Wallis, nemluvil prý o Fermatovi jinak, než jako o „zatraceném francouzovi".

[10] Jde o Diophanta z Alexandrie, po kterém jsou pojmenovány diofantické rovnice, viz příklad 3.1.7. O Diophantovi toho moc nevíme, pravděpodobně žil kolem roku 250. Díky (jak jinak, než aritmetické) hádance na Diophantově náhrobku však víme, že zemřel v 84 letech.

Mám vskutku krásný důkaz tohoto tvrzení. Tento okraj je však příliš malý, než abych jej sem mohl napsat.

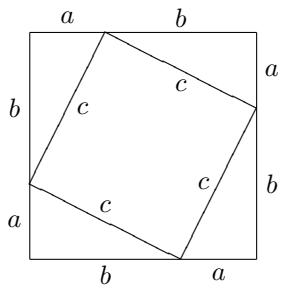
Problém odolával více než 300 let (zůstal tak posledním z nedokázaných Fermatových tvrzení, odtud pochází název věty). Po Fermatově smrti se objevilo mnoho špatných důkazů a pokusy o důkaz velké Fermatovy věty zruinovaly nejednu vědeckou kariéru.

Teprve 23. června 1993 na přednášce v Cambridge předvedl angličan Andrew Wiles důkaz, který je korektní. Intenzivně na něm pracoval sedm let.

Historii pokusů o důkaz velké Fermatovy věty (a samozřejmě Wilesův triumf) popisuje velmi čtivá kniha

☞ S. Singh, *Fermat's Last Theorem*, Fourth Estate Ltd., London, 1997

3.4.9 Příklad Následující krásný důkaz Pythagorovy věty je připisován samotnému Pythagorovi.[11] Máme pravoúhlý trojúhelník s odvěsnami délky a a b, c je délka přepony. Podívejte se na následující obrázek:

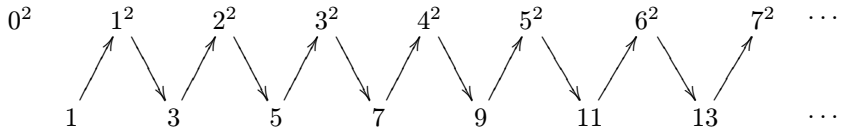


Obsah velkého čtverce je $(a+b)^2 = a^2 + 2ab + b^2$ (pro geometrický důkaz této rovnosti odkazujeme na poznámku 5.3.18). Počítáno jiným způsobem, je obsah velkého čtverce roven $c^2 + 2ab$ (obsah vnitřního čtverce sečtený s obsahem čtyř stejných pravoúhlých trojúhelníků). Tudíž platí rovnost $a^2 + 2ab + b^2 = c^2 + 2ab$. Odečtením dostaneme rovnost

$$a^2 + b^2 = c^2$$

což je Pythagorova věta.

Nyní dokážeme, že existuje nekonečně mnoho přirozených čísel a, b, c, která splňují rovnost $a^2 + b^2 = c^2$ (viz také cvičení 2.4.7). Důkaz je připisován Eukleidovi: představme si následující dvě řady čísel



Vrchní nekonečná řada je řadou čtverců přirozených čísel, spodní řada je nekonečná řada lichých čísel. Pokud ve spodní řadě začnete od čísla 1 a pojedete-li po směru šipek do nějakého lichého čísla n ve spodní řadě, pak šipka z čísla n ukazuje na součet řady lichých čísel od 1 do n (srovnejte s dopisem Leibnize pruské královně z poznámky 1.1.5). Speciálně platí, že přičtením lichého čísla k nějakému čtverci vzniká opět čtverec.

Ve spodní řadě existuje nekonečně mnoho čísel, která jsou čtverci, protože čtverec lichého čísla je liché číslo a číslo 9 je čtverec (tudíž i 9^2, $(9^2)^2$ atd. a všechna tato čísla jsou ve spodní řadě). Tudíž existuje nekonečně mnoho přirozených čísel a, b, c, pro která platí rovnost $a^2 + b^2 = c^2$.

3.4.10 Poznámka Malá Fermatova věta prvočísla necharakterisuje. Existují totiž složená čísla n, pro která platí

jestliže $\gcd(a, n) = 1$, pak v $\mathbb{Z}_n$ platí rovnost $a^{n-1} = 1$.

Takovým složeným číslům, pro která platí tvrzení malé Fermatovy věty, se říká *Carmichaelova*.[12] Carmichaelova čísla existují a je jich dokonce nekonečně mnoho, viz

[11] Pythagoras ze Samu (pravděpodobně 572–497) jako první zavedl myšlenku matematického důkazu jakožto jistoty poznání.
[12] Tato čísla na počátku 20. století objevil a studoval Robert Daniel Carmichael (1879–1967).

☞ W. R. Alford, A. Granville a C. Pomerance, There Are Infinitely Many Carmichael Numbers, *Ann. Math.* 140 (1994), 703–722

Nejmenší Carmichaelovo číslo je $561 = 3 \cdot 11 \cdot 17$.

Carmichaelova čísla nejsou jen teoretická zrůdnost: mají uplatnění například při *Millerově-Rabinově testu prvočíselnosti*, viz knihu

☞ N. Koblitz, *A Course in Number Theory and Cryptography*, Springer, New York, 1994.

nebo algoritmus B.3.8 v dodatcích.

Malou Fermatovu větu lze zobecnit. Zobecnění spočívá na pozorování, že pro prvočíslo p je $p-1$ počet všech čísel z množiny $\{0, 1, \ldots, p-1\}$, která jsou s p nesoudělná. Zavedeme nyní funkci, která dává počet čísel nesoudělných s daným číslem.

3.4.11 Definice *Eulerova*[13] *funkce* φ je definována následovně: pro přirozené číslo $n \geq 2$ je $\varphi(n)$ počet všech čísel z množiny $\{0, 1, \ldots, n-1\}$, která jsou s n nesoudělná. Dále definujeme $\varphi(1) = 1$.

3.4.12 Poznámka Víme, že pro prvočíslo p platí $\varphi(p) = p-1$. Některé další vlastnosti Eulerovy funkce z tohoto faktu okamžitě plynou:

1. Pro prvočíslo p a přirozené číslo $n \geq 1$ platí

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

Důvodem je to, že čísla od 0 do $p^n - 1$, která jsou *soudělná* s p jsou přesně ta, která jsou dělitelná p. Takových čísel je přesně p^{n-1}.

2. Jsou-li m_1 a m_2 nesoudělná čísla, potom platí

$$\varphi(m_1 m_2) = \varphi(m_1)\varphi(m_2)$$

Musíme zjistit počet čísel od 0 do $m_1 m_2 - 1$, která jsou nesoudělná s číslem $m_1 m_2$. Protože čísla m_1, m_2 jsou nesoudělná, využijeme k tomu čínskou větu o zbytcích.

Zvolme číslo x v intervalu od 0 do $m_1 m_2 - 1$. Toto číslo jednoznačně určuje dvojici čísel

$$a_1 \in 0, 1, \ldots, m_1 - 1 \quad a_2 \in 0, 1, \ldots, m_2 - 1$$

pro která platí

$$\begin{aligned} x &= a_1 \quad \text{v } \mathbb{Z}_{m_1} \\ x &= a_2 \quad \text{v } \mathbb{Z}_{m_2} \end{aligned}$$

Obráceně, podle čínské věty o zbytcích, každá dvojice čísel

$$a_1 \in 0, 1, \ldots, m_1 - 1 \quad a_2 \in 0, 1, \ldots, m_2 - 1$$

jednoznačně určuje číslo x v intervalu od 0 do $m_1 m_2 - 1$, pro které platí

$$\begin{aligned} x &= a_1 \quad \text{v } \mathbb{Z}_{m_1} \\ x &= a_2 \quad \text{v } \mathbb{Z}_{m_2} \end{aligned}$$

Této korespondence nyní využijeme. Platí totiž

$\gcd(x, m_1 m_2) = 1$ právě tehdy, když $\gcd(a_1, m_1) = 1$ a současně $\gcd(a_2, m_2) = 1$.

[13] Rodák ze švýcarské Basileje, Leonard Euler (1707–1783), strávil velkou část svého života na dvoře ruského cara Petra Velikého. Za svůj život Euler napsal obrovské množství prací zásadní důležitosti, převážně z matematické analýzy, a to přesto, že od roku 1740 byl téměř slepý. (Ztrátu zraku prý komentoval: *Budu alespoň méně rozptylován.*) Řadu známých symbolů v matematice zavedl právě Euler: značení $f(x)$ pro hodnotu funkce (1734), i pro odmocninu z -1 (1777), $\sum$ pro značení součtu (1755), a další.

Z tohoto faktu okamžitě vyplývá, že $\varphi(m_1 m_2) = \varphi(m_1)\varphi(m_2)$.

3. Z předchozí úvahy plyne rovnost

$$\varphi(p_1^{n_1} \cdot p_2^{n_2} \cdot \ldots \cdot p_r^{n_r}) = p_1^{n_1}(1 - \frac{1}{p_1})p_2^{n_2}(1 - \frac{1}{p_2}) \ldots p_r^{n_r}(1 - \frac{1}{p_r})$$

Známe-li tedy prvočíselný rozklad daného čísla, je nalezení hodnoty Eulerovy funkce snadné. Například

$$\varphi(1960) = \varphi(2^3 \cdot 5 \cdot 7^2) = 2^3(1 - \frac{1}{2}) \cdot 5(1 - \frac{1}{5}) \cdot 7^2(1 - \frac{1}{7}) = 672$$

Nyní zformulujeme jednoduché, ale důležité zobecnění malé Fermatovy věty. Povšimněte si důkazu: je přesnou analogií důkazu malé Fermatovy věty. Stejná důkazová technika nám později umožní zformulovat a dokázat ještě obecnější tvrzení, viz cvičení 5.7.12.

3.4.13 Věta (Eulerova věta) *Je-li* $\gcd(a, m) = 1$, *potom platí*

$$a^{\varphi(m)} = 1 \quad \text{v } \mathbb{Z}_m$$

Důkaz. Budeme postupovat podobně jako při důkazu malé Fermatovy věty.

Protože každé číslo od 0 do $m - 1$, které je nesoudělné s m, má inversi v $\mathbb{Z}_m$, je podle definice Eulerovy funkce v $\mathbb{Z}_m$ přesně $\varphi(m)$ různých invertibilních prvků. Označme je b_1, b_2, …, $b_{\varphi(m)}$.

Nejprve ukážeme, že jsou-li i a j dvě různá čísla z množiny $\{0, 1, \ldots, \varphi(m)\}$, potom

$$b_i \cdot a \neq b_j \cdot a \quad \text{v } \mathbb{Z}_m$$

Kdyby totiž nastala rovnost, znamenalo by to, že v $\mathbb{Z}_m$ platí rovnost $b_i = b_j$, protože předpokládáme, že $\gcd(a, m) = 1$. To je spor.

Posloupnosti

$$b_1 a, b_2 a, \ldots, b_{\varphi(m)} a \quad \text{a} \quad b_1, \ldots, b_{\varphi(m)}$$

musí v $\mathbb{Z}_m$ obsahovat (až na pořadí) stejná čísla. Součiny všech členů obou posloupností v $\mathbb{Z}_m$ jsou si tedy v $\mathbb{Z}_m$ rovny. To znamená, že platí

$$a^{\varphi(m)} \cdot b_1 \cdot \ldots \cdot b_{\varphi(m)} = b_1 \cdot \ldots \cdot b_{\varphi(m)} \quad \text{v } \mathbb{Z}_m$$

Protože součin $b_1 \cdot \ldots \cdot b_{\varphi(m)}$ má inversi v $\mathbb{Z}_m$ (viz cvičení 3.7.2), platí

$$a^{\varphi(m)} = 1 \quad \text{v } \mathbb{Z}_m$$

a to jsme chtěli ukázat. ■

3.4.14 Poznámka Číslo $\varphi(m)$ *není nejmenším exponentem* e, pro který v $\mathbb{Z}_m$ nastane rovnost $a^e = 1$. To ukazuje následující příklad — viz také cvičení 3.7.17 a 3.7.25.

Protože 31 je prvočíslo, platí $\varphi(31) = 30$ a tudíž podle Eulerovy věty je $2^{30} = 1$ v $\mathbb{Z}_{31}$. Protože v $\mathbb{Z}$ platí $2^5 = 32$, platí v $\mathbb{Z}_{31}$ i rovnost $2^5 = 1$.

Eulerovu větu lze využít pro počítání zbytků po dělení pro velké mocniny.

3.4.15 Příklad Spočtěte zbytek čísla $12^{316\,803}$ po dělení číslem 26 741.

Předvedeme dva výpočty: jeden s čistým použitím Eulerovy věty, druhý jako kombinaci Eulerovy věty a čínské věty o zbytcích. V obou případech ale nejprve potřebujeme prvočíselný rozklad čísla $26\,741 = 11^2 \cdot 13 \cdot 17$.

1. Protože $\varphi(26\,741) = \varphi(11^2) \cdot \varphi(13) \cdot \varphi(17) = (11^2 - 11) \cdot 12 \cdot 16 = 21\,120$ a protože $\gcd(12, 26\,741) = 1$, je podle Eulerovy věty $12^{21\,120} = 1$ v $\mathbb{Z}_{26\,741}$. Protože $316\,803 = 21\,120 \cdot 15 + 3$, dostáváme

$$12^{316\,803} = 12^{21\,120 \cdot 15 + 3} = (12^{21\,120})^{15} \cdot 12^3 = 12^3 = 1\,728 \quad \text{v } \mathbb{Z}_{26\,741}$$

2. Protože čísla 11^2, 13 a 17 jsou navzájem nesoudělná, je podle čínské věty o zbytcích číslo $12^{316\,803}$ v $\mathbb{Z}_{26\,741}$ určeno jednoznačně svými zbytky po dělení čísly 11^2, 13 a 17. Každý tento zbytek opět spočteme Eulerovou větou:

$$\begin{aligned} 12^{316\,803} &= 12^{110\cdot 2\,880+3} = 12^3 = 34 \quad \text{v } \mathbb{Z}_{11^2} \\ 12^{316\,803} &= 12^{12\cdot 26\,400+3} = 12^3 = 12 \quad \text{v } \mathbb{Z}_{13} \\ 12^{316\,803} &= 12^{16\cdot 19\,800+3} = 12^3 = 11 \quad \text{v } \mathbb{Z}_{17} \end{aligned}$$

Postupovali jsme podobně jako v předchozím případě a použili jsme $\varphi(11^2) = 110$, $\varphi(13) = 12$ a $\varphi(17) = 16$.

Nyní použijeme čínskou větu o zbytcích, kde $m_1 = 11^2$, $m_2 = 13$ a $m_3 = 17$. S využitím značení důkazu této věty je

$$12^{316\,803} = 34 \cdot M_1 \cdot N_1 + 12 \cdot M_2 \cdot N_2 + 11 \cdot M_3 \cdot N_3 \quad \text{v } \mathbb{Z}_{26\,741}$$

kde $M_1 = 13 \cdot 17 = 221$, $N_1 = M_1^{-1} = 100^{-1} = 23$ v $\mathbb{Z}_{11^2}$, $M_2 = 11^2 \cdot 17 = 2\,057$, $N_2 = M_2^{-1} = 3^{-1} = 9$ v $\mathbb{Z}_{13}$ a $M_3 = 11^2 \cdot 13 = 1\,573$, $N_3 = M_3^{-1} = 9^{-1} = 2$ v $\mathbb{Z}_{17}$. Tudíž

$$12^{316\,803} = 34 \cdot 221 \cdot 23 + 12 \cdot 2\,057 \cdot 9 + 11 \cdot 1\,573 \cdot 2 = 1\,728 \quad \text{v } \mathbb{Z}_{26\,741}$$

Druhý výpočet je zřejmě poněkud početně náročný. Má však své výhody: buďto když počítáme zbytky po dělení obrovským číslem (viz například cvičení 3.7.18) nebo když počítáme zbytky po dělení pevným (velkým) číslem velmi často (to nastává při provozování protokolu RSA, viz příklad 3.6.1).

Pro počítání zbytků po dělení velkých mocnin lze použít i takzvaný *algoritmus opakovaných čtverců*. Vysvětlíme ho na příkladu.

3.4.16 Příklad Pro číslo 110^{113} chceme spočítat zbytek po dělení číslem 2 803. Algoritmem opakovaných čtverců tento problém lze vyřešit následovně:

Zapišme nejprve exponent 113 ve dvojkové soustavě: $113 = 2^6 + 2^5 + 2^4 + 1$, tudíž 113 je ve dvojkové soustavě sedmiciferné číslo 1 110 001.

Nyní vytvoříme posloupnost písmen S o jedničku kratší než je počet cifer exponentu ve dvojkové soustavě. V našem případě $7 - 1 = 6$, utvoříme tedy posloupnost

$$S\ S\ S\ S\ S\ S$$

Písmena S nám vytvořila celkem 7 různých „přihrádek“. Tato místa budeme obsazovat buď písmenem X nebo je necháme prázdná. Postupujeme podle následujícího návodu:

> Čteme binární rozvoj exponentu zleva doprava a místo každé jedničky napíšeme X, je-li v binárním rozvoji nula, místo neobsazujeme.

V našem případě dostaneme posloupnost

$$X\ S\ X\ S\ X\ S\ S\ S\ S\ X$$

Písmeno X symbolizuje násobení daným číslem (v našem případě číslem 110), písmeno S symbolizuje umocnění (squaring) daného čísla na druhou. V obou případech ovšem budeme brát zbytek po dělení číslem 2 803.

Hodnotu hledané mocniny (vždy) inicializujeme číslem 1. Dostaneme tak posloupnost (všechna čísla jsou chápána v $\mathbb{Z}_{2\,803}$):

$$\begin{array}{lll} X & 110 & = 110 \cdot 1 \\ S & 110^2 & = 888 \\ X & 110^3 & = 110 \cdot 888 = -425 \\ S & 110^6 & = (-425)^2 = 1233 \\ X & 110^7 & = 110 \cdot 1233 = 1086 \\ S & 110^{14} & = (1086)^2 = -667 \\ S & 110^{28} & = (-667)^2 = -778 \\ S & 110^{56} & = (-778)^2 = 1481 \\ S & 110^{112} & = (1481)^2 = 1415 \\ X & 110^{113} & = 110 \cdot 1415 = 1485 \end{array}$$

Viz také cvičení 3.7.19.

Eulerovu větu lze také využít k alternativnímu způsobu hledání inverse prvku v $\mathbb{Z}_m$. Platí-li totiž pro nesoudělná čísla a, m v $\mathbb{Z}_m$ rovnost $a^{\varphi(m)} = 1$, potom musí platit $a^{-1} = a^{\varphi(m)-1}$. Tato metoda může někdy být výhodnější než Eukleidův algoritmus.

3.4.17 Příklad Spočtěte inversi čísla 2 v $\mathbb{Z}_{15}$ Eulerovou větou. Protože $15 = 3 \cdot 5$, platí $\varphi(15) = 8$. Protože $\gcd(15, 2) = 1$, platí podle Eulerovy věty v $\mathbb{Z}_{15}$ rovnost $2^8 = 1$, neboli $2 \cdot 2^7 = 1$. Proto je $2^{-1} = 2^7 = 2^4 \cdot 2^3 = 16 \cdot 8 = 1 \cdot 8 = 8$ v $\mathbb{Z}_{15}$ (využili jsme faktu, že v $\mathbb{Z}_{15}$ platí $16 = 1$).

Následující technické lemma je jednoduchým důsledkem Eulerovy věty a čínské věty o zbytcích. Bude se nám hodit při důkazu korektnosti protokolu RSA (věta 3.5.4).

3.4.18 Lemma *Předpokládejme, že v prvočíselném rozkladu čísla m jsou všechna prvočísla pouze v první mocnině.*[14] *Potom v $\mathbb{Z}_m$ platí pro všechna a a pro všechna přirozená čísla k rovnost $a^{1+k\cdot\varphi(m)} = a$.*

Důkaz. Pro $k = 0$ není co dokazovat: rovnost $a^{1+0\cdot\varphi(n)} = a$ platí pro všechna a v $\mathbb{Z}_m$. Pro $k \geq 1$ budeme rovnost dokazovat slabou indukcí podle $k \geq 1$. Prvočíselný rozklad čísla m označme $m = p_1 \cdot p_2 \cdot \ldots \cdot p_r$.

1. Základní krok: chceme dokázat, že platí rovnost $a^{1+\varphi(m)} = a$ pro všechna a v $\mathbb{Z}_m$. Může nastat právě jeden ze dvou případů:

   (a) $\gcd(a, m) = 1$. Potom podle Eulerovy věty platí rovnost $a^{\varphi(m)} = 1$ v $\mathbb{Z}_m$ a po vynásobení obou stran číslem a dostaneme rovnost $a^{1+\varphi(n)} = a$ v $\mathbb{Z}_m$.

   (b) $\gcd(a, m) = d > 1$. Protože $m = p_1 \cdot p_2 \cdot \ldots \cdot p_r$, kde všechna prvočísla $p_1, p_2, \ldots, p_r$ jsou navzájem různá, je d součinem některých z nich. Tudíž d je square-free číslo, stejně jako číslo $c = m/d$. Navíc čísla d a c jsou nesoudělná a platí

   $$0 = a = a^{1+\varphi(m)} \text{ v } \mathbb{Z}_d \qquad a = a^{1+\varphi(m)} \text{ v } \mathbb{Z}_c$$

   Podle čínské věty o zbytcích tedy platí

   $$a = a^{1+\varphi(m)} \text{ v } \mathbb{Z}_m$$

2. Indukční krok: předpokládejme, že rovnost $a^{1+k\cdot\varphi(m)} = a$ platí pro všechna a v $\mathbb{Z}_m$ a pro pevné $k \geq 1$.

   Potom platí $a^{1+(k+1)\cdot\varphi(m)} = a^{1+k\cdot\varphi(m)} \cdot a^{\varphi(m)}$ a podle indukčního předpokladu jde pravou stranu této rovnosti upravit na $a \cdot a^{\varphi(m)}$, neboli na $a^{1+\varphi(m)}$. Podle základního kroku je poslední výraz roven a v $\mathbb{Z}_m$.

■

3.4.19 Poznámka Lemma 3.4.18 *neplatí*, jakmile se v prvočíselném rozkladu čísla m nějaké prvočíslo vyskytuje ve větší než první mocnině (tj., jakmile m není square-free).

Ať $m = 8 = 2^3$. Potom $\varphi(m) = 2^3 - 2^2 = 4$. Zvolme $a = 2$. Potom $a^{1+\varphi(m)} = 2^5 = 32 = 0 \neq 2$ v $\mathbb{Z}_8$.

3.5 Aplikace — protokol RSA

Předpokládejme, že si dva uživatelé, A a B, chtějí vyměňovat zašifrované zprávy. (Zprávou zde budeme rozumět přirozené číslo.) Šifrovací protokol, který zde uvedeme, publikovali v roce 1978 Ronald Rivest, Adi Shamir a Leonard Adleman (odtud zkratka RSA).[15] Jak uvidíme, protokol RSA je založen na platnosti Eulerovy věty (viz také cvičení 3.7.25).

Uživatel A (v literatuře často pojmenovaný Alice) a uživatel B (Bob) si chtějí vyměňovat zprávy menší než předem dané (velké) přirozené číslo N. Postupují následovně:

[14] Takovým číslům m se anglicky říká *square-free*.

[15] Článek: R. L. Rivest, A. Shamir a L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Commun. ACM* 21 (1978), 294–299. V USA byl protokol RSA patentován 20. září 1983. Podle britské vlády byl však tento šifrovací protokol objeven již v roce 1969 Jamesem Ellisem z Government Communication Headquarters (GCHQ).

1. Alice si vybere dvě různá prvočísla p_A a q_A s vlastností $n_A = p_A q_A > N$. Potom si vybere číslo e_A nesoudělné s $\varphi(n_A) = \varphi(p_A q_A) = (p_A - 1) \cdot (q_A - 1)$. Dále Alice spočítá inversi čísla e_A v $\mathbb{Z}_{\varphi(n_A)}$ a označí ji jako d_A.

   Nakonec Alice dvojici čísel $K_{E,A} = (n_A, e_A)$ zveřejní (této dvojici se říká *veřejný klíč*) a svůj *soukromý klíč* $K_{D,A} = (n_A, d_A)$ si nechá pro sebe.

   Bob postupuje analogicky jako Alice: jeho veřejný klíč je $K_{E,B} = (n_B, e_B)$ a jeho soukromý klíč je $K_{D,B} = (n_B, d_B)$.

2. Předpokládejme, že Bob chce Alici poslat přirozené číslo z. Protože Bob Alicin veřejný klíč (n_A, e_A) zná, může spočítat v $\mathbb{Z}_{n_A}$ mocninu $x = z^{e_A}$. Číslo x je zašifrovaná zpráva.

   Alice x dešifruje následujícím způsobem: spočítá v $\mathbb{Z}_{n_A}$ mocninu $x^{d_A} = z$.

Výhodou protokolu RSA je to, že pro prolomení šifry hrubou silou je zapotřebí znát číslo $\varphi(n_A)$ pro určení čísla d_A. Jak uvidíme, znalost čísla $\varphi(n_A)$ je ekvivalentní znalosti prvočíselného rozkladu čísla n_A. Poslední úloha je však výpočetně náročný problém. Protože přechod od $\varphi(n_A)$ k rozkladu n_A a zpět se děje v polynomiálním čase, je protokol RSA relativně bezpečný.

3.5.1 Věta (O bezpečnosti RSA) *Předpokládejme, že číslo n je součinem dvou neznámých různých prvočísel p a q. Znalost těchto prvočísel je ekvivalentní znalosti čísla $\varphi(n)$.*

Důkaz. Pokud známe prvočíselný rozklad $n = p \cdot q$, spočítáme $\varphi(n) = (p-1) \cdot (q-1)$.

Obráceně, předpokládejme, že známe $\varphi(n)$. Pro n sudé je prvočíselný rozklad jednoduché nalézt: $p = 2$, $q = \dfrac{n}{2}$.

Ať je tedy n liché. Hledáme prvočísla p a q. Uvědomme si, že známe jejich součet a součin:

$$\begin{aligned} pq &= n \\ p+q &= n+1-\varphi(n) \end{aligned}$$

Proto musí být čísla p a q kořeny nějaké kvadratické rovnice. Protože n je liché, musí být číslo $p+q$ sudé — označme jej $2b$. Hledaná kvadratická rovnice pak má tvar

$$x^2 - 2bx + n = 0$$

s kořeny

$$p = \frac{2b + \sqrt{4b^2 - 4n}}{2} = b + \sqrt{b^2 - n} \quad \text{a} \quad q = \frac{2b - \sqrt{4b^2 - 4n}}{2} = b - \sqrt{b^2 - n}$$

■

3.5.2 Příklad O čísle $n = 6\,457\,037$ víme, že je součinem dvou různých prvočísel p, q. Navíc víme, že platí $\varphi(n) = 6\,451\,776$. Pomocí věty 3.5.1 nalezneme prvočísla p, q.

Víme tedy, že

$$p \cdot q = n = 6\,457\,037$$

a že platí rovnost

$$6\,451\,776 = \varphi(n) = (p-1) \cdot (q-1) = n + 1 - (p+q) = 6\,457\,038 - (p+q)$$

Známe tedy součin a součet obou prvočísel

$$p \cdot q = 6\,457\,037 \qquad p + q = 5\,262$$

Proto víme, že p a q jsou kořeny kvadratické rovnice

$$x^2 - 5\,262x + 6\,457\,037 = 0$$

a ty nalezneme známým způsobem: diskriminant dané rovnice je 1 860 496 a jeho odmocnina je 1 364. Proto pro kořeny platí

$$p = \frac{5\,262 + 1\,364}{2} = 3\,313 \qquad q = \frac{5\,262 - 1\,364}{2} = 1\,949$$

a to je hledaný prvočíselný rozklad čísla 6 457 037.

3.5.3 Poznámka *Teze sekvenčního výpočtu* (anglicky *Sequential Computation Thesis*) zhruba říká, že vztah mezi všemi sekvenciálními modely výpočtu je polynomiální. V praxi to znamená, že třída problémů řešitelných v polynomiálním čase (tj. „lehkých problémů“) je stejná pro všechny sekvenciální modely výpočtu. Obráceně: těžký problém zůstane těžkým v každém sekvenciálním modelu (viz také poznámku 1.5.5). V tom spočívá význam předchozí věty o bezpečnosti RSA. Problém rozkladu čísla n na prvočísla (zatím) patří k těžkým problémům. Proto je těžké i nalézt hodnotu $\varphi(n)$.

Sekvenciální paradigma výpočtu ovšem neumožňuje *neomezený paralelismus*. Takového paralelismu mistrně využívá nové (zatím pohříchu velmi teoretické) *kvantové paradigma*. Zájem o využití kvantové fyziky ve službách algoritmů byl značně podnícen článkem

☞ P. W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, http://fr.arxiv.org/abs/quant-ph/9508027

ve kterém je popsán faktorizační algoritmus na kvantovém počítači pracující v polynomiálním čase. To by samozřejmě byl konec relativní bezpečnosti RSA. Zatím je ovšem fyzikální implementace takového algoritmu mimo možnost naší techniky.[16] Pro první pokusy o vytvoření programovacího jazyka pro kvantové počítače odkazujeme na článek

☞ P. Selinger, Towards a Quantum Programming Language, *Math. Structures Comput. Sci.* 14(4) 2004, 527–586, http://www.mathstat.dal.ca/∼selinger/papers.html

Zájemce o teoretické podrobnosti odkazujeme na stránky Johna Preskilla z Californian Institute of Technology

☞ http://www.theory.caltech.edu/people/preskill/ph229/

kde najdete ke stažení učebnici kvantového počítání a řadu řešených problémů. Základní informace lze nalézt i v textu

☞ L. Nentvich a J. Velebil, *Kvantové počítání*, http://math.feld.cvut.cz/velebil/

Korektností šifrovacího protokolu rozumíme vzájemnou jednoznačnost procesu zašifrování a dešifrování. Korektnost protokolu RSA zaručuje následující věta:

3.5.4 Věta (O korektnosti RSA) *Pokud je* $x = z^{e_A}$ *v* $\mathbb{Z}_{n_A}$, *pak* $z = x^{d_A}$ *v* $\mathbb{Z}_{n_A}$.

Důkaz. Označme $z' = (z^{e_A})^{d_A}$. V $\mathbb{Z}_{n_A}$ chceme ukázat rovnost $z' = z$.

Protože v $\mathbb{Z}_{\varphi(n_A)}$ platí $e_A \cdot d_A = 1$, existuje přirozené číslo k tak, že $e_A \cdot d_A = 1 + k\varphi(n_A)$. Proto v $\mathbb{Z}_{n_A}$ platí rovnost

$$z' = (z^{e_A})^{d_A} = z^{1+k\varphi(n_A)} = z$$

podle lemmatu 3.4.18. ■

Pro jiné jednoduché a účinné použití prvočísel v kryptosystémech viz cvičení 3.7.30 a 3.7.31.

3.6 Elementární útoky na protokol RSA

V tomto odstavci předvedeme některé jednoduché útoky na šifrovací protokol RSA, které jsou přesto někdy velmi účinné. Podrobněji se o nejrůznějších útocích lze dočíst například v článku

☞ D. Boneh, Twenty Years of Attacks on the RSA Cryptosystem, Notices Amer. Math. Soc. (AMS), Vol. 46, No. 2, 203–213, 1999, http://crypto.stanford.edu/∼dabo/abstracts/RSAattack-survey.html

nebo v knize

☞ N. Smart, *Cryptography: An Introduction*, McGraw-Hill, 2003

[16] Kvantovým počítačem zatím umíme faktorizovat číslo 15. Ovšem číslo 15 umíme faktorizovat i pomocí psa, který je vycvičen třikrát zaštěkat. (Robert Harley, *Sci. crypt.*, 5. 12. 2001.)

Složitější útok na RSA, založený na teorii řetězových zlomků, uvádíme v dodatcích v odstavci A.3.

Základním útokem na RSA je *útok hrubou silou*. O co jde, uvidíme nejlépe na příkladu:

3.6.1 Příklad Přepokládejme, že jsme třetí strana[17] naslouchající výměně zpráv mezi Alicí a Bobem v protokolu RSA a že jsme zachytili zprávu 11 pro Alici. Známe Alicin veřejný klíč $(36\,181, 3\,989)$. Pro dešifrování zprávy potřebujeme zjistit Alicin soukromý klíč. Z věty o bezpečnosti RSA víme, že je nutné rozložit číslo 36 181 na součin prvočísel. Tento prvočíselný rozklad budeme hledat hrubou silou. Stačí vyzkoušet prvočísla $\leq \sqrt{36\,181} \leq 191$. (Vysvětlete proč! Viz také algoritmus B.4.1) Těch je přesně 42:

3	37	79	131	181
5	41	83	137	191
7	43	89	139	
11	47	97	149	
13	53	101	151	
17	59	103	157	
19	61	107	163	
23	67	109	167	
29	71	113	173	
31	73	127	179	

Postupným dělením získáme prvočíselný rozklad $36\,181 = 97 \cdot 373$. Proto je $\varphi(36\,181) = \varphi(97) \cdot \varphi(373) = 96 \cdot 372 = 35\,712$. Alicin soukromý klíč pak získáme zjištěním $3\,989^{-1}$ v $\mathbb{Z}_{35\,712}$ pomocí rozšířeného Eukleidova algoritmu:

a	b	q	r	α_2	α_1	β_2	β_1
35 712	3 989			1	0	0	1
35 712	3 989	8	3 800	0	1	1	-8
3 989	3 800	1	189	1	-1	-8	9
3 800	189	20	20	-1	21	9	-188
189	20	9	9	21	-190	-188	1 701
20	9	2	2	-190	401	1 701	-3 590
9	2	4	1	401	-1 794	-3 590	16 061
2	1	2	0				

Povšimněte si, že jsme nepsali sloupce pro α a β — ty slouží jen k uložení mezivýpočtu. Hledané koeficienty pro Bezoutovu rovnost jsou hodnoty α_1, β_1 z předposledního řádku tabulky. Bezoutova rovnost má tedy tvar

$$1 = (-1\,794) \cdot (35\,712) + (16\,061) \cdot (3\,989)$$

a proto je $3\,989^{-1} = 16\,061$ v $\mathbb{Z}_{35\,712}$. Nalezli jsme Alicin soukromý klíč $(36\,181, 16\,061)$.

Nyní již můžeme zprávu 11 dešifrovat: znamená to spočíst mocninu $11^{16\,061}$ v $\mathbb{Z}_{36\,181}$. Víme, že můžeme použít Eulerovu větu (případně v kombinaci s čínskou větou o zbytcích) nebo algoritmus opakovaných čtverců. Zkombinujeme obě tyto metody, to bývá nejvhodnější. Eulerova věta totiž exponent mocniny dramaticky sníží a pak může efektivně nastoupit algoritmus opakovaných čtverců. Použití Eulerovy věty dává:

$$\begin{aligned} 11^{16\,061} &= 11^{96\cdot 167+29} = 11^{29} \quad \text{v } \mathbb{Z}_{97} \\ 11^{16\,061} &= 11^{372\cdot 43+65} = 11^{65} \quad \text{v } \mathbb{Z}_{373} \end{aligned}$$

Nyní obě mocniny spočteme algoritmem opakovaných čtverců. Binární rozvoje exponentů jsou:

$$(29)_2 = (1,1,1,0,1) \qquad (65)_2 = (1,0,0,0,0,0,1)$$

Algoritmus opakovaných čtverců dává následující dvě tabulky (levá tabulka je počítána v $\mathbb{Z}_{97}$, pravá tabulka je počítána v $\mathbb{Z}_{373}$):

[17] V literatuře je tato třetí strana tradičně pojmenovaná *Eve* z anglického *eavesdropper* — ten, kdo tajně naslouchá.

X	$11 = 1 \cdot 11$	X	$11 = 1 \cdot 11$
S	$11^2 = 121 = 24$	S	$11^2 = 121$
X	$11^3 = 24 \cdot 11 = 264 = 70$	S	$11^4 = 121^2 = 14\,641 = 94$
S	$11^6 = 70^2 = 4\,900 = 50$	S	$11^8 = 94^2 = 8\,836 = 257$
X	$11^7 = 50 \cdot 11 = 550 = 65$	S	$11^{16} = 257^2 = 66\,049 = 28$
S	$11^{14} = 65^2 = 4\,225 = 54$	S	$11^{32} = 28^2 = 784 = 38$
S	$11^{28} = 54^2 = 2\,916 = 6$	S	$11^{64} = 38^2 = 1\,444 = 325$
X	$11^{29} = 6 \cdot 11 = 66$	X	$11^{65} = 325 \cdot 11 = 3\,575 = 218$

Víme tedy, že

$$11^{16\,061} = 66 \text{ v } \mathbb{Z}_{97} \qquad 11^{16\,061} = 218 \text{ v } \mathbb{Z}_{373}$$

Pomocí čínské věty o zbytcích nyní spočteme $11^{16\,061}$ v $\mathbb{Z}_{36\,181}$:

$$11^{16\,061} = 66 \cdot M_1 \cdot N_1 + 218 \cdot M_2 \cdot N_2$$

kde $M_1 = 373$, $N_1 = 373^{-1} = 82^{-1} = 84$ v $\mathbb{Z}_{97}$ a $M_2 = 97$, $N_2 = 97^{-1} = 50$ v $\mathbb{Z}_{373}$ (obě inverse počítáme opět rozšířeným Eukleidovým algoritmem). Tudíž v $\mathbb{Z}_{36\,181}$ platí rovnosti

$$11^{16\,061} = 66 \cdot 373 \cdot 84 + 218 \cdot 97 \cdot 50 = 3\,125\,212 = 13\,646$$

Zpráva, kterou Bob Alici odeslal, je 13 646.

Připomeňme, že věta 3.5.1 říká, že modul n protokolu RSA lze efektivně faktorizovat na prvočísla, známe-li hohnotu $\varphi(n)$ Eulerovy funkce. Následující technické lemma nám umožní faktorizovat n při znalosti veřejného a soukromého klíče (srovnejte s algoritmem B.4.2):

3.6.2 Lemma *Předpokládejme znalost veřejného klíče (n, e) a soukromého klíče (n, d) jednoho účastníka protokolu RSA. Potom lze efektivně faktorizovat číslo n na prvočísla.*

Důkaz. Označme neznámá lichá prvočísla v rozkladu n jako p a q. Víme, že $\varphi(n) = (p-1) \cdot (q-1)$ a víme, že existuje celé číslo k takové, že platí rovnost

$$e \cdot d - 1 = k \cdot \varphi(n) = k \cdot (p-1) \cdot (q-1)$$

Zvolíme libovolné přirozené číslo $x \neq 0$. Podle lemmatu 3.4.18 platí

$$x^{ed-1} = 1 \text{ v } \mathbb{Z}_n$$

Protože (nám známé!) číslo $ed-1$ je jistě sudé, můžeme spočítat „druhou odmocninu“

$$y_1 = x^{\frac{ed-1}{2}} \text{ v } \mathbb{Z}_n$$

Pak platí rovnost

$$y_1^2 - 1 = (y_1 - 1) \cdot (y_1 + 1) = 0 \text{ v } \mathbb{Z}_n$$

a tudíž (v případě, kdy neplatí $y_1 = 1$ ani $y_1 = -1$ v $\mathbb{Z}_n$) můžeme faktorizovat n nalezením $\gcd(y_1 - 1, n)$.

Předpokládejme, že platí $y_1 = -1$. V tom případě se vrátíme na začátek a zvolíme jiné počáteční číslo x.

Pokud platí $y_1 = 1$, spočítáme hodnotu

$$y_2 = x^{\frac{ed-1}{4}} \text{ v } \mathbb{Z}_n$$

(můžeme vydělit čtyřmi, protože obě prvočísla p a q jsou lichá). Dostaneme rovnost

$$y_2^2 - 1 = y_1 - 1 = 0 \text{ v } \mathbb{Z}_n$$

a proto můžeme faktorizovat n nalezením $\gcd(y_2 - 1, n)$ (v případě, kdy neplatí $y_2 = 1$ ani $y_2 = -1$ v $\mathbb{Z}_n$). V případě, kdy $y_2 = -1$ nebo $y_2 = 1$, postupujeme stejně, jako výše, dokud buď nefaktorizujeme n, nebo dokud $\frac{ed-1}{2^t}$ už není dělitelné dvěma. V tom případě volíme další číslo x a začínáme znovu. ■

Následující příklad je školní povahy. Pro realističtější situaci odkazujeme na cvičení 3.7.26.

3.6.3 Příklad Je zadán modul $n = 341$ protokolu RSA a veřejný exponent $e = 7$ a soukromý exponent $d = 43$. Předvedeme, jak faktorizovat číslo n pomocí algoritmu z lemmatu 3.6.2.

Označme $t_1 = \frac{ed-1}{2} = 150$ a zvolme $x = 2$. Protože $y_1 = 2^{t_1} = 2^{150} = 1$ v $\mathbb{Z}_{341}$ (což zjistíme algoritmem opakovaných čtverců), pokračujeme takto: $t_2 = \frac{t_1}{2} = 75$ a počítáme $y_2 = 2^{t_2} = 2^{75} = 32$ v $\mathbb{Z}_{341}$.

Nyní spočteme $\gcd(32 - 1, 341) = 31$ a to je hledaný faktor čísla 341. Celkově platí $341 = 11 \cdot 31$.

3.6.4 Útok při sdíleném modulu Předpokládejme, že k uživatelů provozuje protokol RSA se *sdíleným* modulem $n = p \cdot q$ a že i-tý uživatel má svůj veřejný klíč (n, e_i) a soukromý klíč (n, d_i). Takový provoz je snadné prolomit buď útokem *insidera* (tj. některým z k účastníků) nebo *outsidera* (tj. někým zvenčí):[18]

1. **Útok insidera** Tento útok je možné provést i v případě, kdy klíče byly účastníkům přiděleny „správcem“. To jest i v případě, kdy účastníci znají sice své soukromé klíče, ale nemají žádnou informaci o rozkladu n ani o hodnotě $\varphi(n)$.

   Předpokládejme, že insider, který chce zaútočit, je účastník číslo 1. Protože tento účastník zná oba své klíče (n, e_1) a (n, d_1), může efektivně faktorizovat n pomocí algoritmu z důkazu lemmatu 3.6.2. Jakmile je číslo n faktorizováno, je snadné nalézt $\varphi(n)$ a tudíž spočítat soukromé klíče ostatních účastníků z jejich veřejných klíčů.

2. **Útok outsidera** Tento útok je možné provést, pokud jeden z účastníků provozu posílá *tutéž zprávu* dalším dvěma účastníkům.

   Předpokládejme, že Alice posílá stejnou zprávu z dvěma účastníkům s veřejnými klíči (n, e_1) a (n, e_2). Outsider, Eve, tedy zachytí čísla c_1 a c_2, o kterých ví, že splňují rovnosti

   $$c_1 = z^{e_1} \quad \text{a} \quad c_2 = z^{e_2} \quad \text{v } \mathbb{Z}_n$$

   pro neznámé číslo z. Eve spočítá $d = \gcd(e_1, e_2)$ rozšířeným Eukleidovým algoritmem a obdrží Bezoutovu rovnost

   $$d = e_1 \cdot t_1 + e_2 \cdot t_2 \quad \text{v } \mathbb{Z}$$

   V této rovnosti musí být jedno z čísel t_1, t_2 kladné a druhé záporné. Předpokládejme, že t_1 je kladné a t_2 záporné. Bezoutovu rovnost může tedy Eve přepsat takto:

   $$e_1 \cdot t_1 = d + e_2 \cdot (-t_2) \quad \text{v } \mathbb{Z}$$

   a tudíž může sestavit rovnici

   $$c_1^{t_1} = z^d \cdot c_2^{-t_2} \quad \text{v } \mathbb{Z}_n \tag{3.7}$$

   protože $c_1^{t_1} = z^{e_1 \cdot t_1}$ a $c_2^{-t_2} = z^{e_2 \cdot (-t_2)}$ v $\mathbb{Z}_n$.

   Jde o lineární rovnici pro neznámou z^d, proto lze použít větu 3.1.2:

   (a) Pokud má rovnice (3.7) právě jedno řešení, nalezne Eve z^d.
   V případě, kdy $\gcd(e_1, e_2) = 1$ nalezne Eve zprávu z zcela bez problémů, protože $z^d = z$. V případě, kdy $d = \gcd(e_1, e_2) > 1$ musí Eve vyřešit problém diskrétní odmocniny, což může být časově náročné. Z toho plyne, že volba nesoudělných veřejných exponentů je špatná, protože útok outsidera je velmi ulehčen.

   (b) Pokud má rovnice (3.7) více než jedno řešení, může Eve efektivně faktorizovat modul n. Viz příklad 3.6.5.

Následující dva příklady útoku outsidera jsou opět školní. Pro realističtější situaci odkazujeme na cvičení 3.7.27.

3.6.5 Příklad Alice posílá tutéž zprávu z dvěma účastníkům s veřejnými klíči $(n, e_1) = (703, 11)$ a $(n, e_2) = (703, 7)$.

[18] Outsider je naše dobrá známá Eve, insider se v některé literatuře jmenuje *malicious Marvin*.

1. Eve zachytí dvě zprávy $c_1 = 694$ a $c_2 = 78$ v $\mathbb{Z}_{703}$.

   Eve spočítá $\gcd(11, 7) = 1$ a Bezoutova rovnost má tvar

   $$1 = e_1 \cdot t_1 + e_2 \cdot t_2 = 11 \cdot 2 + 7 \cdot (-3) \quad \text{v } \mathbb{Z}$$

   neboli

   $$11 \cdot 2 = 1 + 7 \cdot 3 \quad \text{v } \mathbb{Z}$$

   Dále Eve získává rovnici

   $$z^{11 \cdot 2} = z^{1+7 \cdot 3} \quad \text{v } \mathbb{Z}_{703}$$

   neboli

   $$(z^{11})^2 = z^1 \cdot (z^7)^3 \quad \text{v } \mathbb{Z}_{703}$$

   Protože $z^{11} = c_1 = 694$ a $z^7 = c_2 = 78$, přepíše Eve poslední rovnici na

   $$694^2 = z \cdot 78^3 \quad \text{v } \mathbb{Z}_{703}$$

   Protože $694^2 = 81$ a $78^3 = 27$ v $\mathbb{Z}_{703}$, musí Eve vyřešit lineární rovnici

   $$81 = z \cdot 27 \quad \text{v } \mathbb{Z}_{703}$$

   To udělá pomocí věty 3.1.2: $\gcd(27, 703) = 1$, daná rovnice má právě jedno řešení

   $$z = 81 \cdot 27^{-1} = 3 \quad \text{v } \mathbb{Z}_{703}$$

   Odeslaná zpráva je $z = 3$.

2. Eve zachytila zprávy $c_1 = 190$ a $c_2 = 133$. Budeme postupovat rychleji. Z upravené Bezoutovy rovnosti

   $$11 \cdot 2 = 1 + 7 \cdot 3 \quad \text{v } \mathbb{Z}$$

   dostáváme rovnici

   $$190^2 = z \cdot 133^3 \quad \text{v } \mathbb{Z}_{703}$$

   neboli

   $$247 = z \cdot 399 \quad \text{v } \mathbb{Z}_{703}$$

   která má podle věty 3.1.2 přesně 19 různých řešení, protože platí $\gcd(703, 399) = 19$.

   Číslo 19 je ovšem netriviální dělitel modulu 703, a Eve právě provedla (v polynomiálním čase) faktorizaci $703 = 19 \cdot 37$ a dále postupuje jako v příkladu 3.6.1: zjistí (opět v polynomiálním čase), $\varphi(703) = 18 \cdot 36 = 648$, soukromý exponent $d_1 = 11^{-1} = 59$ v $\mathbb{Z}_{648}$ a $z = c_1^{d_1} = 190^{59} = 57$.

3.6.6 Útok při stejném malém veřejném exponentu Předpokládejme, že uživatelé protokolu RSA sice mají každý různý modul RSA, sdílejí ale stejný (malý) veřejný exponent menší, než je počet účastníků provozu. Ať tedy k účastníků má veřejné klíče (n_i, e), kde $e < k$ je malé číslo. Pak stačí, aby Eve zachytila stejnou zprávu poslanou e účastníkům.

Tento útok opět předvedeme na školním příkladu. Pro realističtější situaci odkazujeme na cvičení 3.7.28 a cvičení 3.7.29.

3.6.7 Příklad Předpokládejme, že v protokolu RSA máme tři účastníky s veřejnými klíči $(n_1, e) = (253, 3)$, $(n_2, e) = (51, 3)$ a $(n_3, e) = (145, 3)$. Eve zachytí zprávy $c_1 = 86$, $c_2 = 9$ a $c_3 = 40$ pro tyto účastníky, o kterých ví, že vznikly zašifrováním stejné neznámé zprávy z.

Platí tedy soustava rovnic

$$z^3 = 86 \text{ v } \mathbb{Z}_{253} \qquad z^3 = 9 \text{ v } \mathbb{Z}_{51} \qquad z^3 = 40 \text{ v } \mathbb{Z}_{145}$$

Protože moduly $n_1 = 243$, $n_2 = 51$ a $n_3 = 145$ jsou navzájem nesoudělné, smí Eve použít čínskou větu o zbytcích 3.4.1 a spočítat řešení soustavy

$$x = 86 \text{ v } \mathbb{Z}_{253} \qquad x = 9 \text{ v } \mathbb{Z}_{51} \qquad x = 40 \text{ v } \mathbb{Z}_{145}$$

což je číslo $x = 3\,375$ v $\mathbb{Z}_{1\,870\,935}$. Protože ale platí $3\,375 = z^3 < n_1 n_2 n_3 = 1\,870\,935$, nalezne Eve zprávu z obyčejnou třetí odmocninou: $z = 15$.

3.7 Cvičení

3.7.1 Cvičení Pokud existují, nalezněte inverse:

1. Čísla 6 v $\mathbb{Z}_{14}$.
2. Čísla 6 v $\mathbb{Z}_{41}$.
3. Čísla 160 v $\mathbb{Z}_{841}$.

3.7.2 Cvičení Dokažte, že v $\mathbb{Z}_m$ platí: *Jestliže mají a a b inversi, má inversi i součin $a \cdot b$.*

3.7.3 Cvičení Rovnici $ax + by = c$, kde $a, b, c \in \mathbb{Z}$, říkáme diofantická rovnice. Označte jako d největšího společného dělitele čísel $|a|$ a $|b|$. Ukažte následující:

1. Pokud d nedělí c, daná rovnice nemá žádné řešení v $\mathbb{Z}$.
2. Pokud d dělí c, daná rovnice řešení má. (Návod: obě strany rovnice vydělte číslem d a postupujte analogicky příkladu 3.1.7.)

Vyřešte následující diofantické rovnice:

1. $3x + 7y = -3$.
2. $58x + 10y = 4$.
3. $8x - 6y = 3$.

Zobecněte na diofantické rovnice tvaru $a_1x_1 + a_2x_2 + \ldots + a_nx_n = b$. Návod: využijte cvičení 2.4.4 a 2.4.5.

3.7.4 Cvičení Gaussovou eliminací vyřešte následující soustavy rovnic.

1. V $\mathbb{Z}_7$:
$$\begin{array}{rcrcrcrcl} x & + & 2y & - & z & + & 2w & = & 3 \\ 4x & + & & & z & + & w & = & 2 \\ 4x & + & 4y & + & z & + & 2w & = & 2 \end{array}$$
2. V $\mathbb{Z}_5$:
$$\begin{array}{rcrcrcl} x & + & 3y & + & 2z & = & 1 \\ x & + & 2y & + & z & = & 2 \end{array}$$
3. V $\mathbb{Z}_{61}$:
$$\begin{array}{rcrcrcl} 42x & + & 35y & + & 28z & = & 5 \\ 19x & + & 5y & + & 56z & = & 15 \end{array}$$
4. V $\mathbb{Z}_{11}$:
$$10x + 2y - z + 2w = 6$$

3.7.5 Cvičení V $\mathbb{Z}_7$ nalezněte všechny matice $\mathbb{X}$, pro které platí rovnost $\mathbb{A} \cdot \mathbb{X} = \mathbb{X} \cdot \mathbb{A}$, kde

$$\mathbb{A} = \begin{pmatrix} 5 & 4 \\ 1 & 3 \end{pmatrix}$$

Návod: ukažte nejprve, že matice $\mathbb{X}$ musí být rozměrů 2×2, označte

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

a vyřešte příslušnou soustavu rovnic.

3.7.6 Cvičení Ukažte, že kód ISBN je schopen *detekovat prohození* dvou různých cifer.[19]

3.7.7 Cvičení V příkladu 3.2.2 jsme ukázali, jak spočítat matici $\mathbb{G}$, známe-li matici $\mathbb{H}$. V tomto cvičení ukážeme, jak vyřešit následující problém (všechny výpočty jsou v tělese):

(∗) Je zadána matice $\mathbb{G}$. Nalezněte matici $\mathbb{H}$ maximální možné hodnosti takovou, že součin $\mathbb{H} \cdot \mathbb{G}^\top$ je nulová matice.

Postupujte následovně:

1. Dokažte rovnost
$$(\mathbb{A} \cdot \mathbb{B})^\top = \mathbb{B}^\top \cdot \mathbb{A}^\top$$
pro libovolné matice $\mathbb{A}$, $\mathbb{B}$, pro které dává shora uvedené násobení smysl.
2. Dokažte, že součin $\mathbb{A} \cdot \mathbb{B}^\top$ je nulová matice právě tehdy, když součin $\mathbb{B} \cdot \mathbb{A}^\top$ je nulová matice (pro libovolné matice $\mathbb{A}$, $\mathbb{B}$, pro které dává výše uvedené násobení smysl).

Tyto výsledky dovolují role matic $\mathbb{G}$ a $\mathbb{H}$ prohodit. Problém (∗) lze tedy řešit metodou příkladu 3.2.2.

3.7.8 Cvičení Pokud existují, nalezněte inversní matice.

1. V $\mathbb{Z}_6$: $\begin{pmatrix} 7 & 8 & 9 \\ 6 & 5 & 3 \\ 2 & 2 & 1 \end{pmatrix}$
2. V $\mathbb{Z}_{15}$, kde $t \in \mathbb{Z}_{15}$ je parametr: $\begin{pmatrix} 4 & 4 & 1 \\ 2 & 4 & 0 \\ 3 & 4 & t \end{pmatrix}$

3.7.9 Cvičení Zde je problém 26 z knihy *Sun Tzu Suan Ching*:

> *Máme množství věcí, nevíme ale kolik jich je. Počítáme-li je v trojicích, zbudou dvě. Počítáme-li je v pěticích, zbudou tři. Počítáme-li je v sedmicích, zbudou dvě. Kolik věcí máme?*

Vyřešte jej. Zde je obecné řešení od Mistra Suna:

> *Vynásobte počet přebývajících jednotek, když předměty počítáme v trojicích, číslem 70. K tomu přidejte součin 21 a počtu přebývajících jednotek, když předměty počítáme v pěticích. Dále přidejte součin 15 a počtu přebgývajících jednotek, když předměty počítáme v sedmicích. Je-li nyní výsledek 106 nebo více, odčítejte násobky 105.*

Připomíná Vám Mistrovo řešení něco?

3.7.10 Cvičení Nalezněte nejmenší nezáporné řešení soustavy:
$$\begin{array}{rcll} x & = & 2 & \text{v } \mathbb{Z}_3 \\ x & = & 3 & \text{v } \mathbb{Z}_5 \\ x & = & 4 & \text{v } \mathbb{Z}_{11} \\ x & = & 5 & \text{v } \mathbb{Z}_{13} \end{array}$$

3.7.11 Cvičení Ukažte, že řešení x soustavy z čínské věty o zbytcích (věta 3.4.1) lze napsat ve tvaru
$$x = a_1 M_1^{\varphi(m_1)} + a_2 M_2^{\varphi(m_2)} + \ldots + a_r M_r^{\varphi(m_r)} \quad \text{v } \mathbb{Z}_M$$
kde $M = m_1 m_2 \ldots m_r$ a $M_j = M/m_j$ pro $j = 1, \ldots, r$. Návod: příklad 3.4.17.

3.7.12 Cvičení Modifikujme algoritmus pro počítání s velkými čísly z příkladu 3.4.3 následovně:

[19] Celkově tedy kód ISBN detekuje jednu chybu a prohození dvou cifer, což jsou běžné písařské chyby. ISBN bylo navrženo koncem šedesátých let dvacátého století.

1. Chceme zpracovávat celá čísla v rozmezí od $-N_1$ do N_2. Zvolme proto navzájem nesoudělná čísla $m_1, \ldots, m_r$ tak, aby součin $M = m_1 \cdot m_2 \cdot \ldots \cdot m_r$ byl větší než $N_1 + N_2$.
2. Nezáporné číslo X v rozmezí od 0 do N_2 budeme zapisovat jako X, záporné číslo Y v rozmezí od $-N_1$ do 0 budeme zapisovat jako $M + Y$.
3. Dále při výpočtu součtů a součinů postupujeme analogicky příkladu 3.4.3.

Zvolte $m_1 = 3$, $m_2 = 5$, $m_3 = 7$, $m_4 = 11$ a ukažte následující:

1. Tato volba čísel umožňuje zpracovávat čísla v rozmezí od -500 do 500.
2. Vynásobte čísla -20 a 15.

V praxi se volí $m_1 = 5$, $m_2 = 7$, $m_3 = 11$, $m_4 = 13$, $m_5 = 17$, $m_6 = 19$, $m_7 = 23$ a $m_8 = 29$. To umožňuje v algoritmu pracovat s čísly v rozmezí od $N_1 = -2^{31}$ do $N_2 = 2^{31}$.

3.7.13 Cvičení (*k*-Threshold System for Sharing a Secret)
Předpokládejme, že číslo N je heslo, umožňující otevřít bankovní trezor. Dále předpokládejme, že kromě ředitele je v bance $n \geq 3$ úředníků. Jak zajistit, aby v případě, že by ředitel banky (který heslo N zná) byl nemocen, jakákoli trojice (ale žádná dvojice) úředníků byla schopna trezor otevřít? Návod: použijte čínskou větu o zbytcích následujícím způsobem.

> Ať $p_1, \ldots, p_n$ je n různých prvočísel, všechna větší než $\sqrt[3]{N}$, ale mnohem menší než $\sqrt{N}$. Popište částečnou informaci o hesle N, kterou je zapotřebí dát úředníkovi i. (Využijte k tomu p_i.)

Zobecněte na situaci, kdy chcete, aby trezor otevřela každá k-tice úředníků, ale žádná $(k-1)$-tice trezor neotevřela.

3.7.14 Cvičení Ať $m_1, m_2, \ldots, m_r$ jsou přirozená čísla $m_i \geq 2$, $i = 1, \ldots, m_r$. Označme $d_{ij} = \gcd(m_i, m_j)$. Ukažte, že následující tři podmínky jsou ekvivalentní:

1. Soustava
$$\begin{aligned} x &= a_1 \quad \text{v } \mathbb{Z}_{m_1} \\ x &= a_2 \quad \text{v } \mathbb{Z}_{m_2} \\ &\vdots \\ x &= a_r \quad \text{v } \mathbb{Z}_{m_r} \end{aligned}$$
má řešení.
2. Každá soustava
$$\begin{aligned} x &= a_i \quad \text{v } \mathbb{Z}_{m_i} \\ x &= a_j \quad \text{v } \mathbb{Z}_{m_j} \end{aligned}$$
kde $i \neq j$, má řešení.
3. Platí rovnosti
$$a_i = a_j \quad \text{v } \mathbb{Z}_{d_{ij}}$$
kde $i \neq j$.

3.7.15 Cvičení Ukažte, že přirozené číslo $n \geq 2$ je prvočíslo právě tehdy, když v $\mathbb{Z}_n$ platí rovnost $(n-1)! = -1$. Tomuto tvrzení se říká *Wilsonova věta*.
Využijte Wilsonovy věty k nalezení zbytků

1. 34! po dělení 37.

2. 49! po dělení 53.

3. 24! po dělení 29.

3.7.16 Cvičení Dokažte, že rovnost $a^p = a$ v $\mathbb{Z}_p$ platí pro jakékoli prvočíslo p a *libovolné* přirozené číslo a. Odporuje to tvrzení malé Fermatovy věty?

3.7.17 Cvičení Pro přirozené číslo $m \geq 2$ označte jako $\lambda(m)$ nejmenší přirozené číslo, pro které platí znění Eulerovy věty, tj. $\lambda(m)$ je nejmenší číslo takové, že rovnost

$$a^{\lambda(m)} = 1 \quad \text{v } \mathbb{Z}_m$$

platí pro všechna a taková, že $\gcd(a, m) = 1$.

1. Dokažte, že číslo $\lambda(m)$ skutečně existuje. (Návod: použijte Eulerovu větu a princip dobrého uspořádání.)

2. Spočtěte $\lambda(8)$.

3. Dokažte, že platí:

   (a) $\lambda(2) = 1$, $\lambda(4) = 2$ a $\lambda(2^k) = 2^{k-2}$ pro $k > 2$.

   (b) $\lambda(m) = \varphi(m)$, pro libovolné $m = p^k$, kde p je liché prvočíslo.

   (c) $\lambda(a \cdot b) = \operatorname{lcm}(\lambda(a), \lambda(b))$ pro nesoudělná a, b.

Často se ještě dodefinovává hodnota $\lambda(1) = 1$. Takto vzniklé funkci se říká *Carmichaelova funkce* λ.

3.7.18 Cvičení Aplikujte Eulerovu větu:

1. Pro číslo $2^{11\,213} - 1$ určete zbytek po dělení 11.

2. Nalezněte zbytek čísla $6\,647^{362}$ po dělení číslem $m = 7\,785\,562\,197\,230\,017\,200$. Návod: $m = 2^4 \cdot 3^3 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 31 \cdot 37 \cdot 41 \cdot 61 \cdot 73 \cdot 181$ a čínská věta o zbytcích.

3.7.19 Cvičení Popište obecně algoritmus opakovaných čtverců z příkladu 3.4.16 a ukažte jeho korektnost.

3.7.20 Cvičení Připomeňme, že v příkladech 2.3.16 jsme jako $\mathbb{R}[x]$ značili množinu všech polynomů s koeficienty v $\mathbb{R}$ a v neurčité x. *Hodnotu* $p(a)$ polynomu

$$p(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots a_1 \cdot x + a_0$$

v reálném čísle $a \in \mathbb{R}$ lze spočítat *rekursivně* následujícím způsobem (tomu se říká *Hornerovo schéma*):

1. $p(a) := a_n$

2. For $i := 1$ to n do $p(a) := a \cdot p(a) + a_{n-i}$.

Výpočty můžeme uspořádat do následující tabulky (v příkladu počítáme hodnotu $p(x) = 3x^3 - 5x^2 + 12$ v čísle $a = -4$):

	3	−5	0	12
$a = -4$		−12	68	−272
	3	−17	68	−260

Ve vrchním řádku tabulky jsou koeficienty polynomu a v dolním řádku (zleva doprava) jsou aktuální hodnoty $p(a)$ při vykonávání cyklu, konečná hodnota $p(-4) = -260$ je napravo. Do prostředního řádku ukládáme hodnoty $a \cdot p(a)$.

Spočítejte počet násobení, která jsou zapotřebí pro výpočet hodnoty polynomu stupně n Hornerovým schématem. Tento počet porovnejte s počtem násobení, která jsou zapotřebí pro výpočet hodnoty polynomu „přímočarou metodou“.

Použitím Hornerova schématu spočítejte hodnotu

1. $p(x) = -x^2 + 5x + 7$ v $a = -1$.
2. $p(x) = 3x^3 - 15x + 17$ v $a = 18$.
3. $p(x) = -12x^4 + 12x^3 + 4x^2 - 6x + 17$ v $a = 5$.

Vysvětlete, jakým způsobem se Hornerovo schéma používá v algoritmu opakovaných čtverců.

3.7.21 Cvičení O čísle $n = 18\,923$ víte, že je součinem dvou různých prvočísel p, q. Navíc víte, že $\varphi(18\,923) = 18\,648$. Nalezněte prvočísla p, q metodou důkazu věty 3.5.1.

3.7.22 Cvičení Předpokládejme, že v protokolu RSA se všechny zúčastněné strany shodly na hodnotě $N = 26$ a na tom, že zprávy z (tj. čísla z množiny $\{0, 1, \ldots, 25\}$) kódují jednotlivá písmena anglické abecedy: $A = 0$, $B = 1$, …, $Z = 25$.

1. Vytvořte veřejné a soukromé klíče pro dva uživatele (Alici a Boba).
2. Zašifrujte zprávu „AHOJ“.

Je tento způsob provozování protokolu RSA (tj. jedna zpráva = jedno písmeno) rozumný?

3.7.23 Cvičení Lze protokol RSA použít k „podpisu“ zpráv? Jak?

3.7.24 Cvičení Navrhněte následující modifikaci protokolu RSA: každý účastník si zvolí *tři* navzájem různá prvočísla p_1, p_2, p_3, a definuje $n = p_1p_2p_3$.

1. Popište podrobně tvorbu klíčů jednoho účastníka.
2. Je tento modifikovaný protokol RSA korektní?
3. Má tato modifikace protokolu RSA nějaké výhody či nevýhody oproti původnímu protokolu?

3.7.25 Cvičení Ukažte, že protokol RSA lze modifikovat náhradou Eulerovy funkce φ za Carmichaelovu funkci λ (viz cvičení 3.7.17). Přesněji:

1. Popište tvorbu veřejného a soukromého klíče pomocí Carmichaelovy funkce.
2. Dokažte korektnost nového protokolu. Inspirujte se větou 3.5.4.

3.7.26 Cvičení (Pro toto cvičení si nejprve naprogramujte algoritmus opakovaných čtverců.) Je zadán modul $n = 3\,808\,733$ protokolu RSA, veřejný exponent $e = 17$ a soukromý exponent $2\,685\,761$. Faktorizujte číslo n algoritmem z důkazu lemmatu 3.6.2.

3.7.27 Cvičení Alice posílá neznámou zprávu z účastníkům protokolu RSA s veřejnými klíči $(3\,788\,339, 13)$ a $(3\,788\,339, 7)$. Eve zachytí zprávy $c_1 = 1\,852\,941$ a $c_2 = 3\,751\,742$. Nalezněte zprávu z metodou útoku outsidera.

3.7.28 Cvičení Alice poslala stejnou zprávu z čtyřem účastníkům provozu RSA s veřejnými klíči $(n_1, e) = (3\,831\,281, 4)$, $(n_2, e) = (3\,846\,419, 4)$, $(n_3, e) = (3\,813\,973, 4)$ a $(n_4, e) = (3\,862\,823, 4)$.

Zachytili jste zprávy $c_1 = 1\,897\,715$, $c_2 = 1\,108\,813$, $c_3 = 2\,916\,413$ a $c_4 = 277\,353$.

Nalezněte zprávu z metodou útoku při malém společném veřejném exponentu. (Přesvědčete se nejprve, že čísla n_1, n_2, n_3 a n_4 jsou navzájem nesoudělná.)

3.7.29 Cvičení Co dělat, když při útoku na RSA při stejném malém veřejném exponentu *nejsou* moduly účastníků navzájem nesoudělné a Eve tedy *nemůže* použít čínskou větu o zbytcích? Návod: příklad 3.4.6.

3.7.30 Cvičení (Protokol Diffieho a Hellmana — Public Key Agreement)
Předpokládejme, že se Alice a Bob chtějí společně shodnout na tajném čísle. Veškerou výměnu informací však Alice a Bob musí provádět *veřejně*. Možný je následující postup:

1. Alice a Bob se veřejně dohodnou na (velkém) prvočísle p a nenulovém prvku a v $\mathbb{Z}_p$.
2. Alice zvolí číslo n_A a pošle Bobovi mocninu a^{n_A} v $\mathbb{Z}_p$.
3. Bob zvolí číslo n_B a pošle Alici mocninu a^{n_B} v $\mathbb{Z}_p$.
4. Dohodnutý klíč je pak hodnota
$$k = (a^{n_A})^{n_B} = (a^{n_B})^{n_A}$$
v $\mathbb{Z}_p$.

Čísla p, a, a^{n_A} a a^{n_B} jsou zcela veřejná. Jednou z možností, jak zjistit hodnotu k, je najít čísla n_A a n_B. Tomu se říká *problém diskrétního logaritmu*. Ukazuje se, že problém diskrétního logaritmu je výpočetně náročný. Metoda diskrétního logaritmu se zdá být jedinou[20] metodou, jak k zjistit.

Alice a Bob se dohodnou na $p = 3\,967$, $a = 297$ a vymění si čísla 1 210 a 930. Pokuste se nalézt číslo k.

3.7.31 Cvičení (Elgamalův protokol)[21]

Tento jednoduchý šifrovací protokol je založen na problému diskrétního logaritmu. Popíšeme tvorbu klíče:

1. Nejprve se všichni účastnící *veřejně* dohodnou na prvočísle p takovém, že $p-1$ je dělitelné prvočíslem q. Dále jako g označí invertibilní prvek tělesa $\mathbb{Z}_p$ s vlastností $g^{\frac{p-1}{q}} \neq 1$ v $\mathbb{Z}_p$. Zprávou se rozumí *nenulový prvek* tělesa $\mathbb{Z}_p$.
2. Alice zvolí jako svůj *soukromý klíč* přirozené číslo x a uveřejní svůj *veřejný klíč* $h = g^x$ v $\mathbb{Z}_p$.
3. Píšeme-li Alici zprávu m, postupujeme následovně: zvolíme náhodně číslo k a v $\mathbb{Z}_p$ spočítáme $c_1 = g^k$, $c_2 = m \cdot h^k$. Dvojici (c_1, c_2) Alici odešleme.

Vyřešte následující problémy:

1. Popište proces dešifrování.
2. Zformulujte a dokažte větu o korektnosti tohoto protokolu.
3. Zformulujte větu o bezpečnosti tohoto protokolu.

Revize kapitoly

Dozvěděli jsme se:

- ✔ Podali jsme základy lineární algebry nad $\mathbb{Z}_m$ a její aplikace v teorii lineárních kódů.
- ✔ Hlubší poznatky z počítání modulo: Eulerovu větu a čínskou větu o zbytcích. Obě věty umožňují efektivně pracovat s velkými čísly.
- ✔ Popsali jsme tvorbu klíčů a provoz při protokolu RSA. Ukázali jsme také, jak lze na RSA několika elementárními způsoby zaútočit.

Pro přípravu na zkoušku zkuste zodpovědět následující otázky:

- ✔ Navrhněte test řešitelnosti lineárních rovnic $ax = b$ v $\mathbb{Z}_m$ a dokažte korektnost tohoto testu. Sestavte lineární rovnici, která (a) má právě jedno řešení, (b) právě 2 048 různých řešení, (c) nemá žádné řešení.
- ✔ Navrhněte algoritmus, který efektivně násobí a sčítá čísla v rozmezí od $-1\,024$ do 1 024. Diskutujte výpočetní složitost tohoto algoritmu.
- ✔ Důkladně promyslete elementární útoky na RSA z odstavce 3.6 a navrhněte (jednoduché) metody, jak útoky co nejvíce ztížit.

[20] To ovšem nebylo dokázáno, viz D. Boneh a R. Lipton, Algorithms for Black-Box Fields and Their Applications to Cryptography, *Advances in Cryptology — Crypto 96*, Springer-Verlag, Berlin, 1996.

[21] V literatuře je používáno i názvů *El Gamal Protocol*, *El-Gamal Protocol*. Pojmenován je podle svého autora Tahera Elgamala, viz článek T. Elgamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Trans. Inform. Theory* 31 (1985), 469–472.

Doplňující literatura

O teorii lineárních kódů se lze dočíst v knize

☞ J. Adámek, *Kódování*, SNTL, Praha, 1989

nebo v její rozšířené verzi

☞ J. Adámek, *Foundations of Coding*, John Wiley & Sons, New York, 1991

Vzrušující historií šifrování provází kniha

☞ S. Singh, *Kniha kódů a šifer*, Argo + Dokořán, Praha, 2003

Lze také doporučit biografii jednoho z otců moderní computer science

☞ A. Hodges, *Alan Turing: The Enigma*, Random House, London, 1992

Alan Mathison Turing pracoval za druhé světové války v Bletchley Park,[22] kde se podílel na kryptoanalýze nacistických zpráv. V Bletchley Park byl také v lednu 1944 uveden do provozu první programovatelný počítač, viz např.

☞ http://www.picotech.com/applications/colossus.html

Všechny stroje typu Colossus byly po válce zničeny. Začátkem devadesátých let byl zahájen projekt na rekonstrukci Colossa a v roce 1996 byl jeden počítač spuštěn. Pojedete-li kolem, určitě se na něj běžte podívat, stojí to za to.

Ke dni 16. 6. 2007 je největším známým prvočíslem číslo $2^{32\,582\,657} - 1$ mající 9 808 358 cifer. (Toto prvočíslo bylo objeveno 4. září 2006 Stevenem R. Boonem a Curtisem Cooperem.) Prvočíslům je věnována webová stránka

☞ http://primes.utm.edu/primes/

kde se lze o prvočíslech dozvědět téměř vše. Kdo chce vědět o prvočíslech jen trošku, ať si přečte kapitolu B.

[22] Možná jste viděli film *Enigma* (2001, Manhattan Pictures International, režie Michael Apted), který se odehrává v Bletchley Park v roce 1943. Hrdina filmu Tom Jericho nemá ovšem s Alanem Turingem nic společného.

Kapitola 4

Konečná tělesa

Pokud se nápad vyplatí jednou, vyplatí se i dvakrát.
Tom Stoppard

V předchozích kapitolách jsme vyvinuli „číselné systémy“, které nalezly uplatnění v řadě aplikací. Ukázali jsme například, že každé $\mathbb{Z}_p$ je těleso o přesně p prvcích, když p je prvočíslo. Vlastnosti tělesa potom byly důležité pro vybudování lineární algebry a lineární algebra nad $\mathbb{Z}_p$ se ukázala být důležitou pro teorii lineárních kódů. Tuto kapitolu budeme věnovat další analýze konečných těles a v odstavci 4.5 ukážeme aplikaci v teorii cyklických kódů. V odstavci 4.6 ukážeme zajímavou souvislost těles a skládanek origami. Odstavec 4.6 je odbočkou — nepatří do syllabu přednášky.

Připomeňme, *jak* bylo $\mathbb{Z}_p$ zkonstruováno a *proč* jde o těleso:

1. Začali jsme s množinou celých čísel $\mathbb{Z}$, vybudovali jsme tam teorii dělitelnosti a dokázali větu o dělení (tvrzení 2.1.10).

2. Pro pevné $m > 1$ jsme vytvořili množinu $\mathbb{Z}_m$ zbytkových tříd modulo m (definice 2.3.4) a dokázali jsme, že tato množina nese přirozeným způsobem strukturu okruhu (věta 2.3.13).

3. V důsledku 3.1.6 jsme dokázali, že $\mathbb{Z}_m$ je těleso právě tehdy, když je číslo m „ireducibilní v $\mathbb{Z}$“, tj. nelze jej zapsat jako součin $a \cdot b$, kde $1 < a < m$ a $1 < b < m$. (Takovým číslům m samozřejmě říkáme prvočísla, ovšem optika ireducibility bude pro nás užitečná.)

Výše uvedené kroky budeme nyní imitovat s komplikovanější počáteční množinou než s celými čísly. Začneme s množinou polynomů nad libovolným okruhem.

4.1 Polynomy nad okruhem

4.1.1 Definice Ať $\mathbb{K}$ je komutativní okruh s jednotkou. *Polynom v neurčité x nad $\mathbb{K}$ stupně n* ($n \geq 0$) je výraz

$$a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_0 \cdot x^0, \quad a_n \neq 0$$

kde $a_n, a_{n-1}, \ldots, a_0$ jsou prvky $\mathbb{K}$, kterým říkáme *koeficienty*, prvku a_n říkáme *vedoucí koeficient*.

Prvek $0 \in \mathbb{K}$ budeme za polynom považovat též. Budeme mu říkat *nulový polynom* a jeho stupeň je $-\infty$.[1]

Jako $\mathbb{K}[x]$ označíme množinu všech polynomů v neurčité x nad $\mathbb{K}$.

4.1.2 Poznámka Pozor! Polynom jsme definovali jako výraz, *nikoli* jako funkci. Tudíž, například, *rovnost* dvou polynomů

$$a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_0 \cdot x^0 \quad \text{a} \quad b_m \cdot x^m + b_{m-1} \cdot x^{m-1} + \ldots + b_0 \cdot x^0$$

[1] Pozor! $-\infty$ je pouze symbol, který je výhodné požívat! Definujeme $-\infty + n = n - \infty = -\infty$, kde n je libovolné přirozené číslo, dále definujeme $-\infty - \infty = -\infty$. Tyto konvence nám dovolí s jinak dosti výjimečným nulovým polynomem zacházet.

nad $\mathbb{K}$ definujeme „typograficky“: musí platit $m = n$ a $a_n = b_n$, $a_{n-1} = b_{n-1}$, …, $a_0 = b_0$, tj. pokud utvoříme vektory koeficientů

$$(a_n, a_{n-1}, \ldots, a_0) \quad \text{a} \quad (b_m, b_{m-1}, \ldots, b_0)$$

obou polynomů, musí mít oba vektory stejnou délku a odpovídající položky musí být totožné.

Často je skutečně výhodné neurčitou nepsat a uvažovat o polynomech jako o vektorech koeficientů. To vysvětluje slovo *neurčitá*, symbol x používáme proto, abychom mohli polynomy zapisovat obvyklým způsobem. Zápis $3x^5 - 17x^2 + 11x + 7$ je jistě intuitivnější než $(3, 0, 0, -17, 11, 7)$, i když vektorová notace má své výhody (viz například cvičení 4.7.1 a odstavec 4.5).

Pokušení ztotožnit polynomy s funkcemi („vyhodnocováním“ v x) nás může zavést do nesnází. Dva sobě rovné polynomy totiž jistě definují sobě rovné funkce, existují ovšem i různé polynomy, které definují totožné funkce. Příklad: polynomy

$$p(x) = x + 1 \quad \text{a} \quad q(x) = x^3 + 1 \quad \text{nad } \mathbb{Z}_2$$

jsou evidentně různé (mají různý stupeň). Funkce

$$\begin{array}{ccl} a & \mapsto & p(a) \\ 0 & \mapsto & p(0) = 0 + 1 = 1 \\ 1 & \mapsto & p(1) = 1 + 1 = 0 \end{array} \quad \text{a} \quad \begin{array}{ccl} a & \mapsto & q(a) \\ 0 & \mapsto & q(0) = 0^3 + 1 = 1 \\ 1 & \mapsto & q(1) = 1^3 + 1 = 0 \end{array}$$

jsou ovšem totožné. K tomuto problému se ještě vrátíme v důsledku 4.1.14.

Sčítání a násobení polynomů definujeme obvyklým způsobem. Pro polynomy

$$p(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_0 \cdot x^0 \quad \text{a} \quad q(x) = b_m \cdot x^m + b_{m-1} \cdot x^{m-1} + \ldots + b_0 \cdot x^0$$

kde je (bez újmy na obecnosti) $m > n$, definujeme

$$p(x) + q(x) = b_m \cdot x^m + \ldots + b_{n+1} \cdot x^{n+1} + (a_n + b_n) \cdot x^n + (a_{n-1} + b_{n-1}) \cdot x^{n-1} + \ldots + (a_0 + b_0) \cdot x^0$$

a

$$p(x) \cdot q(x) = a_n \cdot b_m \cdot x^{n+m} + \ldots + \Big(\sum_{i+j=k} a_i \cdot b_j \Big) \cdot x^k + \ldots + (a_0 \cdot b_1 + a_1 \cdot b_0) \cdot x + a_0 \cdot b_0 \cdot x^0$$

Následující tvrzení je triviální:

4.1.3 Tvrzení *Ať $\mathbb{K}$ je komutativní okruh s jednotkou. Potom i množina $\mathbb{K}[x]$, spolu s operacemi sčítání a násobení polynomů, je komutativní okruh s jednotkou.*

Pro polynomy nyní budeme chtít dokázat větu o dělení. Ukazuje se, že v plné obecnosti některá tvrzení neplatí a proto se omezíme na polynomy nad tělesem. Pro stupeň polynomu $p(x)$ zavádíme obvyklé značení $\deg(p(x))$.

4.1.4 Příklad Známá rovnost $\deg(p(x) \cdot q(x)) = \deg(p(x)) + \deg(q(x))$ nemusí nad obecným okruhem platit: uvažujte o polynomech

$$p(x) = 2x + 2 \quad \text{a} \quad q(x) = 3x + 1$$

nad $\mathbb{Z}_6$. Potom platí

$$p(x) \cdot q(x) = (2x + 2) \cdot (3x + 1) = 6x^2 + 5x + 2 = 5x + 2 \quad \text{v } \mathbb{Z}_6.$$

Nalezli jsme polynomy $p(x)$, $q(x)$ stupně 1, pro které platí $\deg(p(x) \cdot q(x)) = 1$.

Výše uvedená anomálie je způsobena tím, že v $\mathbb{Z}_6$ platí $2 \cdot 3 = 0$. Pokud se omezíme na polynomy nad *tělesem*, klasická rovnost pro stupeň součinu platí (cvičení 4.7.2). Nejprve budeme potřebovat triviální, ale užitečný fakt:

4.1.5 Lemma *Ať $\mathbb{K}$ je těleso. Pak součin dvou nenulových prvků $\mathbb{K}$ je nenulový prvek.*

Důkaz. Předpokládejme $a \neq 0$ a $b \neq 0$. Protože jak a, tak b mají inversi ($\mathbb{K}$ je těleso), má inversi i součin $a \cdot b$, a to $(a \cdot b)^{-1} = b^{-1} \cdot a^{-1}$. Proto platí $a \cdot b \neq 0$. ■

4.1.6 Lemma *Ať $\mathbb{K}$ je těleso. Pak pro všechny polynomy $p(x)$, $q(x)$ nad $\mathbb{K}$ platí rovnost* $\deg(p(x)\cdot q(x)) = \deg(p(x)) + \deg(q(x))$.

Důkaz. Vzorec jistě platí, když alespoň jeden z polynomů $p(x)$ a $q(x)$ je nulový — připomeňte si konvence o symbolu $-\infty$. Můžeme tedy předpokládat, že ani jeden z polynomů $p(x)$, $q(x)$ nulový není. Označme jako a_n a b_m vedoucí koeficienty těchto polynomů (tudíž $\deg(p(x)) = n$ a $\deg(q(x)) = m$). Pak vedoucí koeficient polynomu $p(x)\cdot q(x)$ je $a_n \cdot b_m \neq 0$ ($\mathbb{K}$ je těleso). Tudíž $p(x)\cdot q(x)$ je polynom stupně $n+m$. ■

4.1.7 Věta (O dělení pro polynomy)
Ať $\mathbb{K}$ je těleso. Ať $a(x)$ a $b(x)$ jsou dva nenulové polynomy v $\mathbb{K}[x]$. Pak existují jednoznačně určené polynomy $q(x)$, $r(x)$ tak, že $\deg(r(x)) < \deg(b(x))$ *a platí rovnost* $a(x) = b(x)\cdot q(x) + r(x)$.

Důkaz. Budeme postupovat obdobně jako při důkazu věty o dělení v $\mathbb{Z}$. Nejprve ukážeme existenci a potom jednoznačnost polynomů $q(x)$, $r(x)$.

Při důkazu existence budeme postupovat indukcí podle $\deg(a(x))$. Povšimněme si, že v případě nerovnosti $\deg(a(x)) < \deg(b(x))$ je existence $q(x)$ a $r(x)$ triviální, stačí položit $q(x) = 0$ a $r(x) = a(x)$. Předpokládejme proto, že nastává situace $n = \deg(a(x)) \geq \deg(b(x)) = m$.

1. Ať $\deg(a(x)) = 0$. Pak je, díky našemu předpokladu, $\deg(b(x)) = 0$, takže $b(x) = b_0 \neq 0$. Definujeme $r(x) = 0$ a $q(x) = a(x)\cdot(b_0)^{-1}$.

2. Předpokládejme, že tvrzení platí pro všechny polynomy $a'(x)$, kde $0 \leq \deg(a'(x)) < n$.

   Pro polynom $a(x)$ stupně n definujte

   $$a'(x) = a(x) - (a_n \cdot b_m^{-1})\cdot x^{n-m}\cdot b(x)$$

   Pak je $\deg(a'(x)) < n$, protože koeficient u x^n je nula. Podle indukčního předpokladu je $a'(x) = b(x)\cdot q'(x) + r'(x)$ pro nějaké polynomy $q'(x)$ a $r'(x)$, kde $\deg(r'(x)) < \deg(b(x))$. Pak platí

   $$\begin{aligned} a(x) &= a'(x) + (a_n \cdot b_m^{-1})\cdot x^{n-m}\cdot b(x) \\ &= b(x)\cdot q'(x) + r'(x) + (a_n \cdot b_m^{-1})\cdot x^{n-m}\cdot b(x) \\ &= b(x)\cdot(q'(x) + (a_n \cdot b_m^{-1})\cdot x^{n-m}) + r'(x) \end{aligned}$$

   Definujeme $q(x) = (q'(x) + (a_n \cdot b_m^{-1})\cdot x^{n-m})$ a $r(x) = r'(x)$.

Abychom ukázali jednoznačnost, předpokládejme, že platí

$$a(x) = b(x)\cdot q_1(x) + r_1(x) = b(x)\cdot q_2(x) + r_2(x)$$

Potom je $b(x)\cdot(q_1(x) - q_2(x)) = r_2(x) - r_1(x)$. Ať jsou polynomy $q_1(x)$ a $q_2(x)$ různé. Potom je podle lematu 4.1.6 $\deg(b(x)\cdot(q_1(x) - q_2(x))) \geq \deg(b(x))$. Protože platí $\deg(r_2(x) - r_1(x)) < \deg(b(x))$, došli jsme ke sporu. Proto jsou si $q_1(x)$ a $q_2(x)$ rovny, stejně jako $r_1(x)$ a $r_2(x)$. ■

Předchozí důkaz není tak složitý, jak se zdá. Konstrukce polynomu $a'(x)$ v onom důkazu je totiž částí známého algoritmu pro dělení polynomů. Protože předchozí důkaz byl důkaz indukcí, víme, že „spuštěním“ tohoto důkazu dostaneme *rekursivní* algoritmus pro dělení polynomu polynomem, viz odstavec 1.2.

4.1.8 Příklad Jsou dány polynomy $a(x) = 2x^3 - 4x + 1$, $b(x) = 3x + 2$ v $\mathbb{R}[x]$. Najděte $q(x)$ a $r(x)$. (Připomeňme, že $\mathbb{R}$ je těleso.)

Podíl $(2x^3 - 4x + 1) : (3x + 2)$ počítáme rekursivně:

1. Vydělíme vedoucí koeficienty, tj. spočteme $\frac{2}{3}$ a to je vedoucí koeficient první aproximace: $\frac{2}{3}\cdot x^{3-1} = \frac{2}{3}x^2$.

2. Vynásobíme $(3x+2)\cdot\frac{2}{3}x^2 = 2x^3 + \frac{4}{3}x^2$ a spočítáme korekci $(2x^3 - 4x + 1) - (2x^3 + \frac{4}{3}x^2) = -\frac{4}{3}x^2 - 4x + 1$. (To je polynom $a'(x)$ z předchozího důkazu.)

3. Nyní máme počítat $(-\frac{4}{3}x^2-4x+1) : (3x+2)$. Pokračujeme rychleji. Aproximace: $-\frac{4}{9}x$, korekce: $-\frac{28}{9}x+1$.

4. Počítáme $(-\frac{28}{9}x + 1) : (3x + 2)$. Aproximace: $-\frac{28}{27}$, korekce: $\frac{83}{27}$. Algoritmus zastavujeme, protože $\deg(\frac{83}{27}) < \deg(3x + 2)$.

Tudíž $q(x) = \frac{2}{3}x^2 - \frac{4}{9}x - \frac{83}{27}$, $r(x) = \frac{83}{27}$.

Algoritmus pro dělení z předchozího příkladu se v obecném tělese nemění. Jediným rozdílem je to, že zlomky nahradíme inversemi.

4.1.9 Příklad Jsou dány polynomy $a(x) = 2x^3 - 4x + 1$, $b(x) = 3x + 2$ v $\mathbb{Z}_5$. Nalezněte $q(x)$ a $r(x)$. ($\mathbb{Z}_5$ je těleso.)

Podíl $(2x^3 - 4x + 1) : (3x + 2)$ počítáme následovně (všechny výpočty jsou v $\mathbb{Z}_5$):

1. „Vydělíme“ vedoucí koeficienty, tj. spočítáme $2 \cdot 3^{-1} = 2 \cdot 2 = 4$ a to je vedoucí koeficient první aproximace: $4 \cdot x^{3-1} = 4x^2$.

2. Spočteme $(3x+2) \cdot 4x^2 = 2x^3+3x^2$ a pak spočteme korekci $(2x^3-4x+1)-(2x^3+3x^2) = -3x^2-4x+1 = 2x^2 + x + 1$.

3. Máme spočítat $(2x^2 + x + 1) : (3x + 2)$. Pokračujeme rychleji. Aproximace: $2 \cdot 3^{-1} \cdot x = 4x$, korekce: $(2x^2 + x + 1) - (2x^2 + 8x) = -7x + 1 = 3x + 1$.

4. Počítáme $(3x + 1) : (3x + 2)$. Aproximace: $3 \cdot 3^{-1} = 1$, korekce: $-1 = 4$. Algoritmus zastavujeme, protože platí $\deg(4) < \deg(3x + 2)$.

Tudíž $q(x) = 4x^2 + 4x + 1$ a $r(x) = 4$.

Zbytek tohoto odstavce věnujeme vysvětlení rozdílu mezi polynomy jako výrazy a polynomy jako funkcemi. Ukáže se, že na rozdíl lze zapomenout, pokud má těleso $\mathbb{K}$ nekonečný počet prvků. (To vysvětluje, proč se v matematické analýze polynomy nad $\mathbb{R}$ často definují jako funkce.)

Připomeňme, že ke každému polynomu nad $\mathbb{K}$ můžeme přiřadit funkci nad $\mathbb{K}$. Například polynomu $p(x) = x^3 + 2x + 1$ nad $\mathbb{Z}_3$ přiřadíme následující funkci na $\mathbb{Z}_3$:

$$\begin{aligned} 0 &\mapsto 0^3 + 2 \cdot 0 + 1 = 1 \\ 1 &\mapsto 1^3 + 2 \cdot 1 + 1 = 3 = 0 \\ 2 &\mapsto 2^3 + 2 \cdot 2 + 1 = 13 = 1 \end{aligned}$$

Píšeme (například) $p(2) = 1$ a říkáme, že $p(x)$ *má ve* 2 *hodnotu* 1. Pro algoritmus, který hodnoty polynomu efektivně počítá, odkazujeme na cvičení 4.7.3.

4.1.10 Definice Prvek $a \in \mathbb{K}$ je *kořen* polynomu $p(x)$, pokud platí rovnost $p(a) = 0$.

4.1.11 Tvrzení *Ať $\mathbb{K}$ je těleso a ať a je prvek $\mathbb{K}$. Hodnota $p(a)$ je zbytek po dělení $p(x)$ polynomem $x - a$. Takže a je kořen polynomu $p(x)$ právě tehdy, když polynom $x - a$ dělí polynom $p(x)$.*

Důkaz. Podle věty o dělení je $p(x) = (x - a) \cdot q(x) + r(x)$. Protože je $\deg(r(x)) < 1$, je $r(x)$ prvek $\mathbb{K}$, řekněme r. Pokud spočteme hodnoty obou stran v a, pak dostaneme rovnost $p(a) = r$.

Druhé tvrzení je triviální. ∎

4.1.12 Důsledek *Polynom $p(x)$ stupně $n \geq 0$ má v tělese $\mathbb{K}$ nanejvýš n různých kořenů.*

Důkaz. Postupujeme indukcí podle n.

1. Pro $n = 0$ je $p(x) = a \neq 0$. Tudíž $p(x)$ nemá žádný kořen.

2. Předpokládejme, že každý polynom stupně $k < n$ má nanejvýš k různých kořenů. Vezmeme polynom $p(x)$ stupně n.

   Nejprve ukážeme, že z našich předpokladů plyne, že $p(x)$ nemůže mít nekonečný počet kořenů. Předpokládejme opak: ať a_1, a_2, ... je nekonečný seznam všech kořenů $p(x)$.

   Vezměme a_1 a uvažujme o rovnici $p(x) = (x - a_1) \cdot q(x)$ (používáme tvrzení 4.1.11). Pak platí $\deg(q(x)) = k < n$ a podle indukčního předpokladu existuje nanejvýš k různých kořenů polynomu $q(x)$. Všechny zbylé prvky a_2, a_3, ... však musí být také kořeny $q(x)$, spor.

   Takže polynom $p(x)$ má konečný počet kořenů. Ať $a_1, \ldots, a_s$ je jejich úplný výčet. Stejnou argumentací jako výše lze ukázat, že $a_2, \ldots, a_s$ jsou kořeny polynomu $q(x)$, tudíž, podle indukčního předpokladu platí $s - 1 \leq k < n$. Dokázali jsme, že $s \leq k + 1 \leq n$.

■

4.1.13 Poznámka Připomeňme známý fakt z algebry, že v $\mathbb{C}[x]$ má polynom $p(x)$ stupně n *přesně* n kořenů (pokud počítáme i jejich násobnosti). Tomuto výsledku se říká *Fundamentální věta algebry.*[2]

Nad jinými tělesy lze nalézt polynomy stupně n, které (dokonce i s násobnostmi) mají *méně* než n různých kořenů. Nejjednodušším příkladem je polynom

$$x^2 + x + 1 \quad \text{v } \mathbb{Z}_2[x]$$

stupně 2, který v $\mathbb{Z}_2$ žádný kořen nemá. Uvažujme o „kvadratické rovnici“ $x^2 + x + 1 = 0$. Je snadné zjistit, že rovnice nemá v $\mathbb{Z}_2$ řešení. Na kořen jsou totiž pouze dva kandidáti a oba selhávají: $0^2 + 0 + 1 = 1 \neq 0$ a $1^2 + 1 + 1 = 1 \neq 0$.

Jiným příkladem je polynom

$$x^2 + 1 \quad \text{v } \mathbb{R}[x]$$

stupně 2, který v $\mathbb{R}$ žádný kořen nemá. Jak uvidíme v příkladu 4.4.4, má tento polynom hodně společného se zavedením komplexních čísel.

4.1.14 Důsledek *V tělese $\mathbb{K}$ jsou následující podmínky ekvivalentní:*

1. *$\mathbb{K}$ má nekonečný počet prvků.*
2. *Nad $\mathbb{K}$ není zapotřebí rozlišovat mezi polynomy jako výrazy a polynomy jako funkcemi.*

Důkaz. Z 1. plyne 2.: víme, že pokud $p(x)$ a $q(x)$ jsou jako polynomy totožné, jsou totožné i jako funkce. Naopak, předpokládejme, že $p(a) = q(a)$ platí pro všechna $a \in \mathbb{K}$ (tj. $p(x)$ a $q(x)$ jsou totožné jako funkce). Vezměme polynom $h(x) = p(x) - q(x)$. Chceme ukázat, že $h(x)$ je nulový polynom. Předpokládejme, že ne. Pak je $h(x)$ polynom stupně ≥ 0 s nekonečným počtem různých kořenů. To je spor.

Ze 2. plyne 1. To plyne okamžitě z poznámky 4.1.2. ■

4.2 Eukleidův algoritmus pro polynomy

Zavedeme nepříliš překvapující terminologii.

4.2.1 Definice Jednoznačně určený polynom $r(x)$ z věty 4.1.7 nazveme *zbytek* polynomu $a(x)$ modulo $b(x)$.

Když $r(x) = 0$ řekneme, že $b(x)$ *dělí* $a(x)$ a tento fakt značíme $b(x) \mid a(x)$.

[2] O důkaz této věty se v 17. století marně pokoušela řada matematiků. Podařilo se to až v roce 1831 velkému německému matematikovi *Karlu Friedrichu Gaussovi* (1777–1855). Je zajímavé, že Gaussův velmi originální důkaz používá geometrické, nikoli algebraické, metody.

4.2.2 Poznámka Situace je opět delikátní a člověk si musí dávat pozor. Uvažujme o polynomech $a(x) = 2x+1$, $b(x) = x+3$ nad $\mathbb{Z}_5$.

Pak platí $b(x) \mid a(x)$, protože $2x+1 = (x+3)\cdot 2 + 0$. Platí ovšem i $a(x) \mid b(x)$: $x+3 = (2x+1)\cdot 3 + 0$.

Takovou situaci ovšem známe i z celých čísel: $2 \mid (-2)$ a $(-2) \mid 2$. Protože jsme v tvrzení 2.1.10 definovali zbytek po dělení jako *nezáporné* číslo, vyšel nám v celočíselném oboru největší společný dělitel jednoznačně. U polynomů bude situace jiná: viz další text.

4.2.3 Definice Polynomy $a(x)$, $b(x)$ takové, že $a(x) \mid b(x)$ a $b(x) \mid a(x)$ platí současně, nazveme *asociované*.

4.2.4 Lemma *Ať $\mathbb{K}$ je těleso. Dva nenulové polynomy $a(x)$, $b(x)$ nad $\mathbb{K}$ jsou asociované právě tehdy, když existuje $r \neq 0$ tak, že $a(x) = r \cdot b(x)$.*

Důkaz. Ať platí $a(x) \mid b(x)$ a $b(x) \mid a(x)$ současně. Potom existují polynomy $q_a(x)$, $q_b(x)$ tak, že $a(x) = b(x)\cdot q_b(x)$ a $b(x) = a(x)\cdot q_a(x)$. Proto platí rovnost $a(x) = a(x)\cdot(q_a(x)\cdot q_b(x))$ a po porovnání stupňů zjistíme, že polynomy $q_a(x)$ a $q_b(x)$ musí mít stupeň nula. Definujme $r = q_b(x)$ a to je hledaný prvek tělesa $\mathbb{K}$.

Obráceně: z toho, že platí $a(x) = r\cdot b(x)$, plyne, že $b(x) \mid a(x)$ a z platnosti $b(x) = r^{-1}\cdot a(x)$ (pracujeme v tělese) plyne, že $a(x) \mid b(x)$. ■

Snadno se nahlédne, že „být asociován“ je relace ekvivalence (viz cvičení 4.7.5) a podle předchozího lemmatu tudíž můžeme v každé třídě ekvivalence zvolit kanonického reprezentanta s vedoucím koeficientem 1.

4.2.5 Definice Polynomu s vedoucím koeficientem 1 říkáme *monický*.

4.2.6 Definice Polynomu $d(x)$ s následujícími dvěma vlastnostmi

1. $d(x)$ je *společný dělitel* $a(x)$, $b(x)$, tj. platí $d(x) \mid a(x)$ a $d(x) \mid b(x)$.
2. $d(x)$ je *největší ze společných dělitelů* $a(x)$, $b(x)$, tj. platí: jestliže $c(x)$ je polynom takový, že platí $c(x) \mid a(x)$ a $c(x) \mid b(x)$, pak platí $c(x) \mid d(x)$.

říkáme *největší společný dělitel* polynomů $a(x)$, $b(x)$ (značení $d(x) = \gcd(a(x), b(x))$).

Je-li $\gcd(a(x), b(x))$ invertibilní prvek $\mathbb{K}$, řekneme, že polynomy $a(x)$, $b(x)$ jsou *nesoudělné*.

Čtenář(ka) nyní oprávněně očekává Eukleidův algoritmus a Bezoutovu rovnost pro polynomy, viz cvičení 4.7.6 a 4.7.7. Algoritmus předvedeme na příkladech.

4.2.7 Příklad Nad $\mathbb{Z}_5$ jsou dány polynomy $a(x) = x^5+1$ a $b(x) = x^2+1$. Postupným dělením, dokud stupeň zbytku není $-\infty$, dostáváme:

$$\begin{aligned} x^5+1 &= (x^3+4x)\cdot(x^2+1) + (x+1) \\ x^2+1 &= (x+4)\cdot(x+1) + 2 \\ x+1 &= (3x+3)\cdot 2 + 0 \end{aligned}$$

Platí $\gcd(x^5+1, x^2+1) = 2$. Tento největší společný dělitel můžeme vyjádřit jako lineární kombinaci zadaných polynomů:

$$\begin{aligned} 2 &= \boxed{(x^2+1)} - (x+4)\cdot\underline{(x+1)} \\ &= \boxed{(x^2+1)} - (x+4)\cdot(\boxed{(x^5+1)} - (x^3+4x)\cdot\boxed{(x^2+1)}) \\ &= (x^4+4x^3+4x^2+x+1)\cdot\boxed{(x^2+1)} + (4x+1)\cdot\boxed{(x^5+1)} \end{aligned}$$

(Zbytky jsou opět podtrženy, „definitivní“ polynomy jsou v rámečku a se zbytkem zacházíme jako s koeficienty lineární kombinace.)

Ukázali jsme, že x^5+1 a x^2+1 jsou navzájem nesoudělné. Viz také poznámka 4.3.6.

4.2.8 Poznámka Největší společný dělitel dvou polynomů není určen jednoznačně. Každý polynom, asociovaný s největším společným dělitelem, je opět největší společný dělitel.

Pro polynomy $a(x) = x^2 + x$ a $b(x) = 2x^2 + x + 1$ nad $\mathbb{Z}_5$ můžeme Eukleidův algoritmus zahájit dvěma různými způsoby. Oba polynomy jsou totiž „stejně veliké“, přesně řečeno mají stejný stupeň. Jak uvidíme, tyto dva výpočty nám dají *dva různé* největší společné dělitele. Oba největší společní dělitelé jsou však asociovány. Všechny další výpočty jsou v $\mathbb{Z}_5$.

První výpočet dává sérii rovností

$$\begin{aligned} x^2 + x &= 3 \cdot (2x^2 + x + 1) + (3x + 2) \\ 2x^2 + x + 1 &= (4x + 1) \cdot (3x + 2) + 4 \\ 3x + 2 &= (2x + 3) \cdot 4 + 0 \end{aligned}$$

a tudíž je $\gcd(a(x), b(x)) = 4$. Druhý výpočet dává

$$\begin{aligned} 2x^2 + x + 1 &= 2 \cdot (x^2 + x) + (4x + 1) \\ x^2 + x &= (4x + 3) \cdot (4x + 1) + 2 \\ 4x + 1 &= (2x + 3) \cdot 2 + 0 \end{aligned}$$

a proto je $\gcd(a(x), b(x)) = 2$. Uvědomme si ale, že polynomy 2 a 4 jsou asociovány:

$$2 = 3 \cdot 4 \qquad 4 = 2 \cdot 2$$

Stejně jako pro čísla, můžeme zformulovat *rozšířený* Eukleidův algoritmus, který počítá největšího společného dělitele a koeficienty pro Bezoutovu rovnost.

4.2.9 Rozšířený Eukleidův algoritmus pro polynomy nad tělesem $\mathbb{K}$
Vstup: dva polynomy $a(x)$, $b(x)$ nad $\mathbb{K}$ takové, že $\deg(a(x)) \geq \deg(b(x))$.
Výstup: $d(x) = \gcd(a(x), b(x))$ a polynomy $\alpha(x)$, $\beta(x)$ takové, že platí $d(x) = \alpha(x) \cdot a(x) + \beta(x) \cdot b(x)$.

1. Je-li $b(x) = 0$, položme $d(x) := a(x)$, $\alpha(x) := 1$, $\beta(x) := 0$ a algoritmus končí.
2. Položme $\alpha_2(x) := 1$, $\alpha_1(x) = 0$, $\beta_2(x) := 0$, $\beta_1(x) = 1$.
3. Dokud platí $\deg(b(x)) > -\infty$, dělejte následující:
   - 3.1 Spočítáme $q(x)$ a $r(x)$ tak, že platí $a(x) = q(x) \cdot b(x) + r(x)$, kde $\deg(r(x)) < \deg(b(x))$.
   - 3.2 Položíme $\alpha(x) := \alpha_2(x) - q(x) \cdot \alpha_1(x)$, $\beta(x) := \beta_2(x) - q(x) \cdot \beta_1(x)$.
   - 3.3 Položíme $a(x) := b(x)$, $b(x) := r(x)$.
   - 3.4 Položíme $\alpha_2(x) := \alpha_1(x)$, $\alpha_1(x) := \alpha(x)$, $\beta_2(x) := \beta_1(x)$, $\beta_1(x) := \beta(x)$.
4. Položíme $d(x) := a(x)$, $\alpha(x) := \alpha_2(x)$, $\beta_2(x) := \beta(x)$ a algoritmus končí.

Spustíme výše uvedený algoritmus na polynomy z příkladu 4.2.7 a naše výpočty uspořádáme do následující tabulky:

$a(x)$	$b(x)$	$q(x)$	$r(x)$	$\alpha_2(x)$	$\alpha_1(x)$	$\beta_2(x)$	$\beta_1(x)$
$x^5 + 1$	$x^2 + 1$			1	0	0	1
$x^5 + 1$	$x^2 + 1$	$x^3 + 4x$	$x + 1$	0	1	1	$4x^3 + x$
$x^2 + 1$	$x + 1$	$x + 4$	2	1	$4x + 1$	$4x^3 + x$	$x^4 + 4x^3 + 4x^2 + x + 1$
$x + 1$	2	$3x + 3$	0				

(Všimněte si, že do tabulky hodnoty $\alpha(x)$, $\beta(x)$ nepíšeme, protože jsou použity jen k uložení mezivýpočtů. Navíc nemusíme vyplňovat poslední řádek tabulky — tabulku přestáváme vyplňovat v okamžiku, kdy je $r(x) = 0$, hledané hodnoty $\alpha(x)$ a $\beta(x)$ jsou hodnoty $\alpha_1(x)$ a $\beta_1(x)$ z předposledního řádku.) Je tedy $\gcd(a(x), b(x)) = 2$ a Bezoutova rovnost má tvar

$$2 = (4x+1)\cdot(x^5+1) + (x^4+4x^3+4x^2+x+1)\cdot(x^2+1)$$

Totální korektnost rozšířeného Eukleidova algoritmu pro polynomy dokážeme stejně jako pro celá čísla, viz poznámka 2.1.17 a cvičení 4.7.6.

4.3 Algebraická rozšíření

V tomto odstavci budeme imitovat postup vedoucí k $\mathbb{Z}_m$: namísto $\mathbb{Z}$ začneme s množinou $\mathbb{K}[x]$, namísto m zvolíme pevný polynom $m(x)$ a s prvky $\mathbb{K}[x]$ budeme pracovat jako se zbytky modulo $m(x)$. Začneme příkladem.

4.3.1 Příklad Ať $\mathbb{K} = \mathbb{Z}_2$ a $m(x) = x^2 + 1$. Množina $\mathbb{Z}_2[x]$ obsahuje čtyři zbytky po dělení $m(x)$:

$$0, 1, x, x+1$$

Umíme tyto zbytky sčítat a násobit? Sčítání nepředstavuje žádný problém:

$+$	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

To znamená, že zbytky sčítáme stejně jako polynomy nad $\mathbb{Z}_2$. Protože sčítání nezvyšuje stupeň, sečtením dvou zbytků získáme opět zbytek.

Násobení však stupeň zvýšit může. My však víme, co dělat: vynásobíme zbytky jako polynomy a v případě potřeby výsledek nahradíme zbytkem po dělení (srovnejte s výpočtem $2 \cdot 3 = 6 = 1$ v $\mathbb{Z}_5$):

$\cdot$	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	1	$x+1$
$x+1$	0	$x+1$	$x+1$	0

Očekáváme strukturu okruhu. Všimněte si následujícího faktu:

> Pokud obě výše uvedené tabulky omezíme na 0, 1 (tj. na původní prvky $\mathbb{Z}_2$), dostáváme původní sčítání a násobení v $\mathbb{Z}_2$!

Proto novému okruhu říkáme *rozšíření* okruhu $\mathbb{Z}_2$ — k původním dvěma číslům jsme přidali dvě „nová“, sice x a $x+1$.

Samozřejmě: z jednoho příkladu bychom neměli dělat dalekosáhlé závěry. V následujícím tvrzení popíšeme přesně, o co jde. Jednoduché důkazy přenecháme jako cvičení. Ve zbytku tohoto odstavce je $\mathbb{K}$ těleso a $m(x)$ je pevně zvolený polynom nad $\mathbb{K}$ stupně ≥ 1. Řekneme, že polynomy $a(x)$, $b(x)$ jsou *kongruentní modulo* $m(x)$, (značení $a(x) \equiv b(x) \pmod{m(x)}$), pokud existuje polynom $k(x)$ tak, že $a(x) - b(x) = k(x) \cdot m(x)$.

4.3.2 Tvrzení *Kongruence modulo $m(x)$ je relace ekvivalence na množině $\mathbb{K}[x]$, která respektuje sčítání a násobení polynomů. Proto můžeme na třídách ekvivalence modulo $m(x)$ zavést sčítání a násobení:*

$$[a(x)]_{m(x)} \oplus [b(x)]_{m(x)} = [a(x)+b(x)]_{m(x)} \quad \text{a} \quad [a(x)]_{m(x)} \odot [b(x)]_{m(x)} = [a(x)\cdot b(x)]_{m(x)}$$

Výsledná struktura, značená $\mathbb{K}[x]/m(x)$, je komutativní okruh s jednotkou.

4.3.3 Poznámka Pochopitelně, stejně jako u $\mathbb{Z}_m$, budeme často psát

$$a(x) \text{ v } \mathbb{K}[x]/m(x)$$

namísto $[a(x)]_{m(x)}$ a, například

$$a(x) \cdot b(x) \text{ v } \mathbb{K}[x]/m(x)$$

namísto $[a(x)]_{m(x)} \odot [b(x)]_{m(x)}$.

4.3.4 Definice Okruhu $\mathbb{K}[x]/m(x)$ říkáme *algebraické rozšíření* okruhu $\mathbb{K}$.

4.3.5 Poznámka Termín „rozšíření“ lze vysvětlit přesně pomocí pojmu homomorfismus okruhů:

1. Množina $\mathbb{K}$ „je“ podmnožinou množiny $\mathbb{K}[x]/m(x)$. Přesněji: prvky $\mathbb{K}[x]/m(x)$ jsou třídy ekvivalence $[a(x)]_{m(x)}$ polynomů stupně $< \deg(m(x))$ (to jsou přesně representanti zbytků modulo $m(x)$ — porovnejte se standardními representanty prvků $\mathbb{Z}_m$). Proto je třída $[a]_{m(x)}$ prvkem $\mathbb{K}[x]/m(x)$ pro každé $a \in \mathbb{K}$. To dává návod, jak $\mathbb{K}$ chápat jako podmnožinu množiny $\mathbb{K}[x]/m(x)$.
2. Inkluse $\mathbb{K}$ v $\mathbb{K}[x]/m(x)$ respektuje sčítání a násobení, tj. platí rovnosti $[a+b]_{m(x)} = [a]_{m(x)} \oplus [b]_{m(x)}$ a $[a \cdot b]_{m(x)} = [a]_{m(x)} \odot [b]_{m(x)}$. Toto vnoření navíc respektuje nulu a jednotku $\mathbb{K}$, tj. $[0]_{m(x)}$ je neutrální ke sčítání v $\mathbb{K}[x]/m(x)$ a $[1]_{m(x)}$ je neutrální k násobení v $\mathbb{K}[x]/m(x)$.

Tyto dvě vlastnosti nám dovolují každý polynom nad $\mathbb{K}$ jako polynom nad $\mathbb{K}[x]/m(x)$ pomocí homomorfismu okruhů $a \mapsto [a]_{m(x)}$. (Připomeňte si, že neurčitou můžeme odstranit a namísto s polynomy můžeme pracovat s vektory koeficientů.)

Takový postup použijeme v příkladu 4.4.3: $\mathbb{Z}_2$ budeme považovat za podmnožinu $\mathbb{Z}_2[x]/(x^2+x+1)$ a ukážeme, že x^2+x+1 (jako polynom nad $\mathbb{Z}_2[x]/(x^2+x+1)$) má v $\mathbb{Z}_2[x]/(x^2+x+1)$ kořen. Tento jev v plné obecnosti uvidíme v tvrzení 4.4.5.

4.3.6 Poznámka V příkladu 4.2.7 jsme ukázali, že pro polynomy $a(x) = x^5+1$ a $b(x) = x^2+1$ nad $\mathbb{Z}_5$ platí $\gcd(x^5+1, x^2+1) = 2$. proto jsou polynomy $a(x)$ a $b(x)$ nesoudělné.

Ukážeme, že inverse prvku x^2+1 v $\mathbb{Z}_5[x]/(x^5+1)$ existuje. Technika, kterou použijeme, je podobná hledání inverse v $\mathbb{Z}_m$. Nejprve připomeneme Bezoutovu rovnost

$$2 = (x^4+4x^3+4x^2+x+1) \cdot (x^2+1) + (4x+1) \cdot (x^5+1)$$

z příkladu 4.2.7 (rozšířený Eukleidův algoritmus). Dále vyjádříme kanonický, monický, největší společný dělitel jako kombinaci x^5+1 a x^2+1. To je snadné: „vydělíme“ výše uvedenou rovnost číslem 2 (protože pracujeme nad $\mathbb{Z}_5$, znamená to vynásobit číslem 3):

$$1 = (3x^4+2x^3+2x^2+3x+3) \cdot (x^2+1) + (2x+3) \cdot (x^5+1) \quad \text{v } \mathbb{Z}_5[x]$$

Přečtěme tuto rovnost modulo x^5+1:

$$1 = (3x^4+2x^3+2x^2+3x+3) \cdot (x^2+1) \quad \text{v } \mathbb{Z}_5[x]/(x^5+1)$$

Proto je inverse k x^2+1 modulo x^5+1 rovna polynomu $3x^4+2x^3+2x^2+3x+3$.

V algebraických rozšířeních můžeme řešit lineární rovnice podobně jako řešíme lineární rovnice v $\mathbb{Z}_m$. Vyřešme rovnici:

$$(x^2+1) \cdot p(x) = (x+1) \quad \text{v } \mathbb{Z}_5[x]/(x^5+1)$$

Víme již, že (x^2+1) je invertibilní, a proto platí

$$\begin{aligned} p(x) &= (x^2+1)^{-1} \cdot (x+1) \\ &= (3x^4+2x^3+2x^2+3x+3) \cdot (x+1) \\ &= 3x^5+2x^4+2x^3+3x^2+3x+3x^4+2x^3+2x^2+3x+3 \\ &= 3x^5+4x^3+3x+3 \end{aligned}$$

Poslední polynom je „příliš veliký“ — jeho stupeň není < 5, a proto jej nahradíme zbytkem modulo x^5+1:

$$p(x) = 3x^5+4x^3+3x+3 = 4x^3+3x \quad \text{v } \mathbb{Z}_5[x]/(x^5+1)$$

Viz také cvičení 4.7.9 a 4.7.10.

4.4 Ireducibilní polynomy

Zajímavou otázkou je samozřejmě to, kdy je $\mathbb{K}[x]/m(x)$ těleso. Očekáváme, že by se $m(x)$ mělo v $\mathbb{K}[x]$ „chovat jako prvočíslo“. Takovým polynomům říkáme ireducibilní:

4.4.1 Definice Polynom $m(x)$ je *ireducibilní*, pokud jej nelze zapsat jako součin $a(x) \cdot b(x)$ dvou polynomů kladných stupňů.

4.4.2 Tvrzení *Ať $\mathbb{K}$ je těleso a ať $m(x)$ je ireducibilní polynom nad $\mathbb{K}$. Potom je okruh $\mathbb{K}[x]/m(x)$ těleso.*

Důkaz. Potřebujeme ukázat, že každý nenulový prvek $a(x)$ má v $\mathbb{K}[x]/m(x)$ inversi. To ale plyne okamžitě z toho, že polynomy $m(x)$ a $a(x)$ jsou nesoudělné, protože $m(x)$ je ireducibilní, $0 \leq \deg(a(x)) < \deg(m(x)))$ a z Bezoutovy rovnosti. ■

4.4.3 Příklad Ať $\mathbb{K} = \mathbb{Z}_2$ a ať $m(x) = x^2 + x + 1$ nad $\mathbb{Z}_2$. Polynom $m(x)$ je ireducibilní, protože platí

$$\begin{aligned} m(x) &\neq x \cdot x \\ m(x) &\neq x \cdot (x+1) \\ m(x) &\neq (x+1) \cdot (x+1) \end{aligned}$$

Očekáváme, že $\mathbb{Z}_2[x]/(x^2 + x + 1)$ je těleso a je tomu skutečně tak:

$+$	0	1	x	$x+1$
0	0	1	x	$x+1$
1	1	0	$x+1$	x
x	x	$x+1$	0	1
$x+1$	$x+1$	x	1	0

$\cdot$	0	1	x	$x+1$
0	0	0	0	0
1	0	1	x	$x+1$
x	0	x	$x+1$	1
$x+1$	0	$x+1$	1	x

Tabulka násobení ukazuje, že $\mathbb{Z}_2[x]/(x^2 + x + 1)$ je těleso. Víme, že jde o rozšíření $\mathbb{Z}_2$ o dva nové prvky x a $x + 1$. Označme je a, b a v tomto novém značení přepišme tabulky sčítání a násobení:

$+$	0	1	a	b
0	0	1	a	b
1	1	0	b	a
a	a	b	0	1
b	b	a	1	0

$\cdot$	0	1	a	b
0	0	0	0	0
1	0	1	a	b
a	0	a	b	1
b	0	b	1	a

(4.1)

Nyní můžeme ukázat další zajímavou vlastnost, kterou $\mathbb{Z}_2[x]/(x^2 + x + 1)$ má. V poznámce 4.1.13 jsme ukázali, že rovnice $x^2 + x + 1 = 0$ nemá v $\mathbb{Z}_2$ řešení. Situace je v rozšíření $\mathbb{Z}_2[x]/(x^2 + x + 1)$ dramaticky odlišná: obě „nová čísla“ a, b jsou kořeny polynomu $x^2 + x + 1$! To plyne z následujících dvou rovnic:

$$a^2 + a + 1 = b + a + 1 = 1 + 1 = 0 \quad \text{a} \quad b^2 + b + 1 = a + b + 1 = 1 + 1 = 0$$

Zdá se, že rozšířit $\mathbb{Z}_2$ znamená *přidat* k $\mathbb{Z}_2$ kořeny polynomu $x^2 + x + 1$!

4.4.4 Příklad Pravděpodobně nejznámější instancí přidání kořenů je těleso $\mathbb{C}$ komplexních čísel. Vzniká totiž jako algebraické rozšíření $\mathbb{R}[x]/(x^2 + 1)$.

Prvky $\mathbb{R}[x]/(x^2+1)$ jsou zbytky modulo x^2+1, tj. výrazy tvaru $ax+b$, kde a, b jsou v $\mathbb{R}$. Sčítání je definováno následovně:

$$(ax + b) + (cx + d) = (a + c)x + (b + d)$$

Zbytky násobíme jako polynomy a potom spočítáme zbytek modulo $x^2 + 1$:

$$(ax + b) \cdot (cx + d) = acx^2 + (ad + bc)x + bd = (ad + bc)x + (bd - ac)$$

Kdybychom psali namísto $ax + b$ psali $ai + b$, pak bychom zjistili, že $\mathbb{R}[x]/(x^2 + 1)$ jsou stará dobrá komplexní čísla. Nepřekvapuje nás, že v $\mathbb{C}$ má polynom $x^2 + 1$ kořen. Jde o prvek x (chápaný jako prvek $\mathbb{R}[x]/(x^2 + 1)$). Napišme to těžkopádně, ale přesně:

$$[x]^2_{x^2+1} \oplus [1]_{x^2+1} = [x^2 + 1]_{x^2+1} = [0]_{x^2+1}$$

Povšimněte si ovšem, že jsme k $\mathbb{R}$ přidali celou řadu dalších prvků, než jen kořeny polynomu $x^2 + 1$. K tomu jsme byli „donuceni“ — výsledná struktura musí být těleso.

Případ komplexních čísel je instancí obecného faktu.

4.4.5 Tvrzení *Ať $\mathbb{K}$ je těleso a ať $p(x)$ je ireducibilní polynom nad $\mathbb{K}$. Potom polynom $p(x)$ má v $\mathbb{K}[x]/p(x)$ kořen.*

Důkaz. Jsou dvě možnosti: $\deg(p(x)) = 1$ a $\deg(p(x)) > 1$.

V prvním případě můžeme psát $p(x) = a_1x + a_0$, pro nějaká a_1, a_0 v $\mathbb{K}$, kde $a_1 \neq 0$. Potom je $-a_0 \cdot a_1^{-1}$ kořen polynomu $p(x)$ v $\mathbb{K}$. Protože $\mathbb{K}[x]/p(x)$ je rozšíření okruhu $\mathbb{K}$, je prvek $-a_0 \cdot a_1^{-1}$ kořenem $p(x)$ v $\mathbb{K}[x]/p(x)$.

Předpokládejme, že $\deg(p(x)) = n > 1$ a že $p(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_0$. Prvky $\mathbb{K}[x]/p(x)$ jsou polynomy stupně nanejvýš $n - 1$ a my je sčítáme a násobíme jako zbytky modulo $p(x)$. Označme jako α třídu ekvivalence polynomu x, tj. $\alpha = [x]_{p(x)}$. Spočteme hodnotu $p(x)$ v α:

$$\begin{aligned} p(\alpha) &= a_n \cdot \alpha^n + a_{n-1} \cdot \alpha^{n-1} + \ldots + a_0 \\ &= a_n \cdot [x]^n_{p(x)} + a_{n-1} \cdot [x]^{n-1}_{p(x)} + \ldots + a_0 \\ &= a_n \cdot [x^n]_{p(x)} + a_{n-1} \cdot [x^{n-1}]_{p(x)} + \ldots + a_0 \\ &= [a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \ldots + a_0]_{p(x)} \\ &= [p(x)]_{p(x)} \end{aligned}$$

Zde jsme použili faktu, že kongruence modulo $p(x)$ respektuje sčítání a násobení. Ukázali jsme, že $[p(\alpha)]_{p(x)} = [0]_{p(x)}$ in $\mathbb{K}[x]/p(x)$, neboli α je kořenem polynomu $p(x)$ v $\mathbb{K}[x]/p(x)$. ■

4.4.6 Poznámka Ve cvičení 4.7.11 dokážete, že nad tělesem $\mathbb{K}$ nemá žádný ireducibilní polynom stupně > 1 v $\mathbb{K}$ kořen.

Opak samozřejmě neplatí: polynom

$$x^4 + x^2 + 1 = (x^2 + x + 1) \cdot (x^2 + x + 1)$$

nad $\mathbb{Z}_2$ je reducibilní, přesto žádný kořen v $\mathbb{Z}_2$ nemá.

Následující důležitý fakt nebudeme dokazovat.

4.4.7 Tvrzení *Ať p je prvočíslo. Pro každé přirozené číslo $n \geq 2$ existuje ireducibilní polynom stupně n nad $\mathbb{Z}_p$. Množina ireducibilních polynomů nad $\mathbb{Z}_p$ je tedy nekonečná.*

4.4.8 Důsledek *Pro každé prvočíslo p a každé kladné přirozené číslo n existuje těleso mající přesně p^n různých prvků.*

Důkaz. V případě $n = 1$ je hledané těleso $\mathbb{Z}_p$. V případě $n > 1$ vezměme ireducibilní polynom $p(x)$ nad $\mathbb{Z}_p$ stupně n. Hledané těleso je $\mathbb{Z}_p[x]/p(x)$. Okruh $\mathbb{Z}_p[x]/p(x)$ je samozřejmě těleso, to plyne z ireducibility polynomu $p(x)$. Potřebujeme jen spočítat počet prvků množiny $\mathbb{Z}_p[x]/p(x)$. Protože prvky jsou všechny polynomy stupně nanejvýš $n - 1$, stačí spočítat počet takových polynomů. Každý polynom stupně nanejvýš $n - 1$ lze ztotožnit s vektorem délky n s položkami v $\mathbb{Z}_p$. Takových vektorů je přesně p^n. ■

4.4.9 Definice Konečnému tělesu o p^n prvcích (p je prvočíslo a n je kladné přirozené číslo) budeme říkat *Galoisovo těleso*[3] a budeme jej značit $\mathrm{GF}(p^n)$.

4.4.10 Poznámka Ve skutečnosti je *každé* konečné těleso $\mathbb{K}$ isomorfní Galoisovu tělesu $\mathrm{GF}(p^n)$ pro nějaké prvočíslo p a nějaké kladné přirozené číslo n. Tento fakt opět nedokazujeme.

Isomorfismem zde rozumíme existenci *přejmenování* prvků množiny $\mathbb{K}$ na prvky množiny $\mathrm{GF}(p^n)$, které respektuje strukturu tělesa, Tento pojem upřesníme později.

4.4.11 Poznámka Algebraická rozšíření se používají i k *důkazům nemožnosti* některých konstrukcí. Víme například, že existují vzorce umožňující řešit obecné algebraické rovnice s reálnými koeficienty až do stupně 4 včetně. Vzorce pro řešení lineárních a kvadratických rovnic si jistě dobře pamatujete. Vzorce pro řešení rovnic stupně 3 a 4 jsou již poměrně komplikované, přesto existují,[4] viz například kapitolu VIII knihy

☞ B. L. van der Waerden, *Algebra I, II*, Springer, New York, 2003

Pomocí algebraických rozšíření lze dokázat, že obecný vzorec pro řešení algebraických rovnic stupně 5 a vyšších *neexistuje*.

Podle *fundamentální věty algebry* (poznámka 4.1.13) víme, že například rovnice

$$x^{136} - \sqrt{\pi} \cdot x^{127} + 3\,497 \cdot x^{116} + 27 = 0$$

má (i s násobnostmi) přesně 136 kořenů v $\mathbb{C}$. Galoisův výsledek tvrdí, že *neexistuje vzorec používající čtyři základní algebraické operace a odmocňování*, do kterého bychom zapsali koeficienty obecné rovnice 136. stupně a který by nám tyto kořeny našel.

Samozřejmě, že některé *speciální* rovnice stupně 5 a vyšších řešit umíme. Vyřešte například v $\mathbb{C}$ rovnici $x^{17} - 2 = 0$.

4.5 Aplikace — cyklické kódy

V odstavci 3.3 jsme zavedli lineární p-kód dimense k a délky n jako vektorový podprostor V prostoru $(\mathbb{Z}_p)^n$ dimense k. Takovému kódu říkáme *cyklický*, pokud platí následující podmínka:

> Pro každé kódové slovo $(a_{n-1}, a_{n-2}, \ldots, a_1, a_0)$ je i jeho *cyclický posun* $(a_{n-2}, \ldots, a_2, a_1, a_{n-1})$ kódové slovo.

Cyklický posun má elegantní popis, pokud každý vektor

$$v = (a_{n-1}, a_{n-2}, \ldots, a_1, a_0)$$

budeme chápat jako polynom

$$v(x) = a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \ldots + a_1 \cdot x + a_0$$

stupně $\leq n-1$ nad $\mathbb{Z}_p$. Cyklický posun získáme, pokud nejdříve polynom $v(x)$ vynásobíme polynomem x:

$$x \cdot v(x) = a_{n-1} \cdot x^n + a_{n-2} \cdot x^{n-1} + \ldots + a_1 \cdot x^2 + a_0 \cdot x$$

a poté ztotožníme x^n s 1, tj. budeme-li chápat $x \cdot v(x)$ jako prvek $\mathbb{Z}_p[x]/(x^n - 1)$:

$$x \cdot v(x) = a_{n-2} \cdot x^{n-1} + \ldots + a_1 \cdot x^2 + a_0 \cdot x + a_{n-1}$$

V tomto odstavci se nebudeme pouštět do hlubokých vod, pro podrobnosti odkazujeme na knihu

[3] Evariste Galois (1811–1832) byl francouzský matematik. Zemřel na následky souboje. Noc před soubojem strávil sepisováním svých matematických výsledků, ve kterých dokázal neexistenci vzorců pro algebraické rovnice stupně 5 a vyšších. Viz poznámka 4.4.11.

[4] Ani nepřekvapí, že příběhy objevů těchto vzorců jsou příběhy tajemství, zrady a hořkosti. Hlavními aktéry byli italští matematikové Girolamo Cardano (1501–1576) a Nicolo Tartaglia (1499–1557). Tartaglia byl jedním z prvních, kteří uměli vyřešit obecné rovnice třetího stupně. Když jej Cardano požádal o sdělení metody, Tartaglia nejprve odmítl, poté svolil a Cardana zapřísáhl, aby nikdy tuto formuli nezveřejnil. Cardano zřejmě skutečně chtěl dodržet slovo, ovšem později se dozvěděl, že rovnice třetího stupně uměl řešit již Scipione del Ferro (1465–1526). Rozhodl se proto slib daný Tartagliovi porušit a v roce 1545 publikoval knihu *Ars magna*, kde popisuje řešení obecných rovnic třetího a čtvrtého stupně. Tartaglia to velmi rozzuřilo a oba muži si pak již jen vyměňovali urážky.

☞ J. Adámek, *Kódování*, SNTL, Praha, 1989

4.5.1 Lemma *Ať V je cyklický kód délky n. Pak platí:*

1. *Pro polynomy $v_1(x)$, ..., $v_r(x)$, které jsou kódovými slovy, a pro libovolné prvky α_1, ..., α_r v $\mathbb{Z}_p$, je lineární kombinace*
$$\alpha_1 \cdot v_1(x) + \ldots + \alpha_r \cdot v_r(x)$$
opět kódovým slovem.

2. *Pro kódové slovo $v(x)$ a libovolný polynom $p(x)$ v $\mathbb{Z}_p[x]/(x^n - 1)$ je součin $p(x) \cdot v(x)$ opět kódové slovo.*

Důkaz.

1. Triviální, protože V je *a priori* lineární kód.

2. Jestliže $p(x) = 0$, není co dokazovat. Ať tedy $p(x) \neq 0$ je ve tvaru
$$p(x) = a_s \cdot x^s + \ldots + a_0$$
tj. ať $p(x)$ je polynom stupně $s \in \{0, \ldots, n-1\}$. Vynásobíme-li $v(x)$ „sumandem“ $a_i \cdot x^i$ ($i \in \{0, \ldots, s\}$), provádíme vlastně i-krát cyklický posun a poté násobíme skalárem. Součin $a_i \cdot x^i \cdot v(x)$ je tudíž kódové slovo, stejně jako jejich součet:
$$p(x) \cdot v(x) = \sum_{i=0}^{s} a_i \cdot x^i \cdot v(x)$$

■

V případě cyklického kódu V můžeme generující matici (viz odstavec 3.3) nahradit *generujícím polynomem*.

4.5.2 Věta *Ať V je cyklický (n, k)-kód, který obsahuje nenulové slovo. Pak V obsahuje slovo $g(x)$ stupně $n - k$ a kódová slova jsou přesně součiny $p(x) \cdot g(x)$, kde $p(x)$ je polynom stupně $< k$.*

Důkaz. Ať $g(x)$ je kódové slovo nejmenšího stupně $s \neq -\infty$. (Zde používáme princip dobrého uspořádání.) Ukážeme, že platí rovnost $s = n - k$.

Podle lemmatu 4.5.1 víme, že všechny součiny $p(x) \cdot g(x)$ jsou kódová slova, kde $p(x)$ je libovolný polynom stupně $< k$.

Obráceně, ať $v(x)$ je kódové slovo. Ukážeme, že $v(x)$ je násobek polynomu $g(x)$ nějakým polynomem $p(x)$ stupně $< k$. Vydělíme se zbytkem
$$v(x) = q(x) \cdot g(x) + r(x)$$
a povšimneme si, že rozdíl $r(x) = v(x) - q(x) \cdot g(x)$ dvou kódových slov je opět kódové slovo. Protože platí $\deg(r(x)) < s$, platí $r(x) = 0$ (jinak by to bylo ve sporu s naší volbou $g(x)$). Tudíž platí $v(x) = q(x) \cdot g(x)$, hledaný polynom $p(x)$ je kvocient $q(x)$.

Potřebujeme ještě ukázat rovnost $k = n - s$. K tomu stačí si uvědomit, že kódová slova $g(x)$, $x \cdot g(x)$, ..., $x^{n-s-1} \cdot g(x)$ jsou lineárně nezávislá (nad $\mathbb{Z}_p$) a generují V. Tudíž rovnost $s = n - k$ platí. ■

Generující polynom cyklického kódu se používá k zakódování informace, analogicky jako u lineárních kódů. Ukážeme to na příkladu.

4.5.3 Příklad Ať $g(x) = x^2 + 2$ je polynom nad $\mathbb{Z}_3$. Považujme $g(x)$ za generující polynom cyklického kódu délky 6. Budeme tedy pracovat s algebraickým rozšířením $\mathbb{Z}_3[x]/(x^6 - 1)$. Prvky $\mathbb{Z}_3[x]/(x^6 - 1)$ jsou zbytky modulo $x^6 - 1$, tj. polynomy nad $\mathbb{Z}_3$ stupně < 6.

Víme, že kódová slova jsou přesně násobky $g(x)$ v $\mathbb{Z}_3[x]/(x^6 - 1)$. Tak například pro $p(x) = 2x^3 + x$ je součin
$$p(x) \cdot g(x) = (2x^3 + x) \cdot (x^2 + 2) = 2x^5 + 2x^3 + 2x \quad \text{v } \mathbb{Z}_3[x]/(x^6 - 1)$$
kdové slovo. Polynom $p(x)$ chápeme jako *informaci*, kterou chceme poslat (informační bity jsou koeficienty $p(x)$) a součin $2x^5 + 2x^3 + 2x$ je zpráva, kterou skutečně posíláme (porovnejte s odstavcem 3.3).

Podobně jako u cyklických kódů existuje i u cyklických kódů pojem *kontrolního polynomu* $h(x)$. Nejprve potřebujeme dokázat technický fakt o generujících polynomech.

4.5.4 Lemma *Ať $g(x)$ je generující polynom cyklického kódu délky n nad $\mathbb{Z}_p$. Potom $g(x) \mid (x^n - 1)$.*

Důkaz. Označme $d(x) = \gcd(x^n - 1, g(x))$. Podle Bezoutovy rovnosti platí

$$d(x) = \alpha(x) \cdot (x^n - 1) + \beta(x) \cdot g(x) \quad \text{v } \mathbb{Z}_p[x]$$

a tudíž

$$d(x) = \beta(x) \cdot g(x) \quad \text{v } \mathbb{Z}_p[x]/(x^n - 1)$$

Tím jsme dokázali, že $d(x)$ je kódové slovo. Protože jsme zvolili $g(x)$ jako kódové slovo nejmenšího stupně (viz důkaz věty 4.5.2), platí $\deg(g(x)) \leq \deg(d(x))$.

Celkově tedy platí, že polynomy $d(x)$ a $g(x)$ jsou asociované, a proto $g(x) \mid (x^n - 1)$. ■

4.5.5 Poznámka Samozřejmě, každého dělitele polynomu $x^n - 1$ lze chápat jako generující polynom nějakého cyklického kódu. Spolu s lemmatem 4.5.4 tedy dává množina všech dělitelů polynomu $x^n - 1$ popis všech cyklických kódů délky n.

Analogií kontrolní matice z lineárních kódů je *kontrolní polynom* cyklického kódu:

4.5.6 Věta *Ať V je cyklický kód délky n nad $\mathbb{Z}_p$. Pak existuje polynom $h(x)$, který má následující vlastnosti:*

$v(x)$ je kódové slovo právě tehdy, když $h(x) \cdot v(x) = 0$ v $\mathbb{Z}_p[x]/(x^n - 1)$.

Důkaz. Podle lemmatu 4.5.4 víme, že $g(x) \mid (x^n - 1)$. Proto existuje $h(x)$ tak, že v $\mathbb{Z}_p[x]$ platí rovnost $h(x) \cdot g(x) = x^n - 1$ a tedy platí i rovnost $h(x) \cdot g(x) = 0$ v $\mathbb{Z}_p[x]/(x^n - 1)$.

Následující výpočty jsou v $\mathbb{Z}_p[x]/(x^n - 1)$. Je-li $v(x)$ kódové slovo, platí $v(x) = p(x) \cdot g(x)$ pro nějaké $p(x)$. Pak je $h(x) \cdot v(x) = (h(x) \cdot g(x)) \cdot p(x) = 0$.

Obráceně, je-li $h(x) \cdot v(x) = 0$, pak $h(x) \cdot v(x) = p(x) \cdot (x^n - 1)$ pro nějaké $p(x)$. Pak ale platí rovnost $p(x) \cdot (x^n - 1) = p(x) \cdot (g(x) \cdot h(x))$. Takže platí $h(x) \cdot v(x) = p(x) \cdot (g(x) \cdot h(x))$ a z toho plyne $v(x) = p(x) \cdot g(x)$. ■

4.6 Odbočka — geometrické konstrukce a origami

Další zajímavou aplikací algebraických rozšíření je neexistence řešení některých geometrických problémů, například *trisekce úhlu*: neexistuje algoritmus, který by kružítkem a pravítkem rozdělil obecný úhel na třetiny. Nebudeme neexistenci trisekce úhlu pomocí kružítka a pravítka dokazovat, to by zabralo poměrně hodně času. Důkaz spočívá v tom, že geometrickými metodami *nelze* obecně vyřešit rovnici

$$4x^3 - 3x = \cos 3\alpha, \quad \text{kde } \alpha \text{ je obecný úhel.}$$

Kdybychom uměli zkonstruovat reálné řešení x výše uvedené rovnice, pak musí platit $x = \cos \alpha$.[5] Jakmile ovšem umíme zkonstruovat $\cos \alpha$, je snadné kružítkem a pravítkem sestrojit úhel α.

Dáme nyní do souvislosti konstrukce pomocí kružítka a pravítka a japonské umění *origami* (skládanek z papíru).

Konstrukce pravítkem a kružítkem znáte ze střední školy. Těmito konstrukcemi se intensivně zabývala starořecká geometrie. Axiomy pro tyto konstrukce jsou následující:

(G1) Dvěma zkonstruovanými body lze proložit přímku.

(G2) Pro dvě (nerovnoběžné) zkonstruované přímky lze zkonstruovat jejich průsečík.

[5] Zkuste to dokázat. Návod: postupně upravujte $\cos 3\alpha = \cos(2\alpha + \alpha) = \cos 2\alpha \cos \alpha - \sin 2\alpha \sin \alpha = \ldots = 4\cos^3 \alpha - 3\cos \alpha$.

(G3) Jsou-li zkonstruovány body P_1 a P_2, pak lze zkonstruovat kružnici se středem v P_1 procházející bodem P_2.

(G4) Je-li zkonstruována kružnice a je-li zkonstruována jiná kružnice nebo přímka, lze zkonstruovat jejich průsečík(y).

Výše uvedená sada axiomů vytváří poměrně dosti bohaté možnosti konstrukcí v rovině. Přesto lze ukázat, že některé konstrukce tyto axiomy *neumožňují*. Zajímavé proto je, že analogická sada axiomů pro origami tvoří podstatně silnější systém. Sada axiomů pro origami, kterou budeme používat, pochází z článku

☞ H. Huzita, Understanding Geometry through Origami Axioms, *Proceedings of the First Conference on Origami in Education and Therapy*, ed. J. Smith, British Origami Society, 1992, 37–70

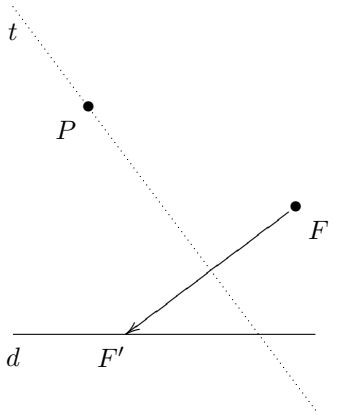
(O1) Jsou-li P_1, P_2 dva zkonstruované body, pak můžeme papír přehnout tak, aby záhyb procházel body P_1 a P_2.

(O2) Jsou-li P_1, P_2 dva zkonstruované body, pak můžeme papír přehnout tak, aby bod P_1 přešel na bod P_2.

(O3) Jsou-li l_1, l_2 dvě zkonstruované přímky (záhyby), pak můžeme papír přehnout tak, aby přímka l_1 přešla na přímku l_2.

(O4) Je-li l zkonstruovaná přímka a P zkonstruovaný bod, který na l neleží, pak můžeme papír přehnout tak, aby záhyb procházel bodem P a byl kolmý na přímku l.

(O5) Jsou dány dva zkonstruované body F, P, oba neleží na zkonstruované přímce d. Pak (kdykoli takový záhyb *existuje*) je možné papír přehnout tak, že záhyb prochází bodem P a bod F přejde na bod F ležící na přímce l:



(O6) Jsou dány dva zkonstruované body F_1, F_2 a dvě zkonstruované přímky d_1, d_2. Pak (kdykoli takový záhyb *existuje*) je možné papír přehnout tak, že bod F_1 přechází na přímku d_1 a bod F_2 přechází na přímku d_2.

Dá se ukázat, že záhyby popsané v axiomech (O1)–(O5) lze sestrojit pomocí axiomů (G1)–(G4) kružítkem a pravítkem (pokuste se o to, viz také následující příklad 4.6.1).

Než ukážeme, jak lze pomocí axiomů pro origami provést *trisekci úhlu*, zastavíme se u geometrické interpretace axiomu (O5).

4.6.1 Příklad Připomeňme z klasické geometrie, že *parabola* je útvar, zadaný *řídící přímkou* d (zvanou též *direktrix*) a *ohniskem* F (zvaným též *fokus*). Takto zadaná parabola je potom ta množina bodů roviny, které mají od direktrix d stejnou vzdálenost jako od ohniska F.

Jako příklad můžeme v kartézské soustavě[6] souřadnic zvolit bod $F = (0, b)$ a přímku d s rovnicí $y = a$. (Namalujte si obrázek.)

Obecný bod (x, y) má mít

[6]Nazváno podle francouzského matematika a filosofa René Descarta (latinsky *Renatus Cartesius*), který žil v letech 1596 až 1650. Descartovým cílem byla snaha o maximální pochopitelnost dosaženého poznání a ve svých filosofických pracech se velmi často inspiroval matematickým způsobem uvažování. Jistě znáte větu *Cogito, ergo sum.* (*Myslím, tedy jsem.* Ve francouzském originále *Je pense, donc je suis.*) z knihy *Discours de la méthode* (*Rozprava o metodě*), která Descartovi dovoluje dokázat vlastní existenci. Pochybujeme-li totiž o všem, nemůžeme pochybovat o skutečnosti pochybování samotného.

vzdálenost od ohniska F: $\sqrt{x^2 + (y-b)^2}$

stejnou jako

vzdálenost od direktrix d: $\sqrt{(y-a)^2}$

a tudíž naše parabola má rovnici (proveďte příslušné úpravy):

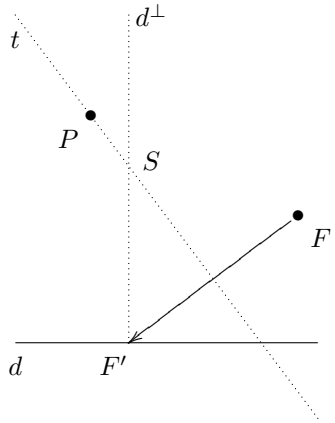
$$y(2b - 2a) = x^2 + (b^2 - a^2)$$

V případě, že $b = a$ (tj. když direktrix prochází ohniskem), parabola kolabuje v přímku $x = 0$ (tj. v osu y). V případě, že $b \neq a$, můžeme rovnici zkrátit číslem $(2b - 2a)$ a dostaneme tak rovnici

$$y = \frac{1}{2b - 2a}x^2 + \frac{a+b}{2}$$

Protože každou dvojici d, F můžeme proložit kartézské souřadnice tak, aby přímka d byla rovnoběžná s osou x a bod F ležel na ose y, je jasné, že geometrická definice paraboly je totožná s pojmem paraboly, kterou známe z analytické geometrie.

Nyní vysvětlíme, proč přímka t zkonstruovaná axiomem (O5) je tečna k parabole zadané pomocí d a F, která prochází bodem P: sestrojme v bodě F' kolmici $d^\perp$ k přímce d a označme jako S průsečík $d^\perp$ a t:



Povšimněme si, že S leží na dané parabole, protože má stejnou vzdálenost od d jako od F (přímka t je osa úsečky FF'). Přímka t je navíc tečnou dané paraboly, protože půlí úhel FSF'. (Zde využíváme faktu, že přímka t procházející bodem S paraboly je tečnou v bodě S právě tehdy, když t půlí úhel daný přímkami FS a přímkou rovnoběžnou s osou paraboly procházející bodem S.) Z toho, jak jsme argumentovali, by mělo být jasné, že tečnu t dovedeme sestrojit kružítkem a pravítkem.

Tento příklad jasně ukazuje, že někdy je lepší využít geometrických vlastností než analytické geometrie: pokuste se dokázat, že t je tečna metodami analytické geometrie. Budete k tomu potřebovat diferenciální počet.

Axiom (O6) také souvisí s tečnami k parabolám. Jeho geometrická interpretace je následující:

(O6) Pomocí F_1, d_1, F_2, d_2 jsou zadány *dvě* paraboly. Záhyb, konstruovaný axiomem (O6), je tečna k oběma parabolám současně.

Zkuste algebraickou metodou dokázat, že axiom (O6) je ekvivalentní tomu, umět vyřešit rovnici třetího stupně. Tento fakt právě způsobuje, že axiomy pro origami jsou *silnější* než axiomy pro konstrukce kružítkem a pravítkem. Jako aplikaci ukážeme, jak pomocí origami vyřešit problém trisekce úhlu:

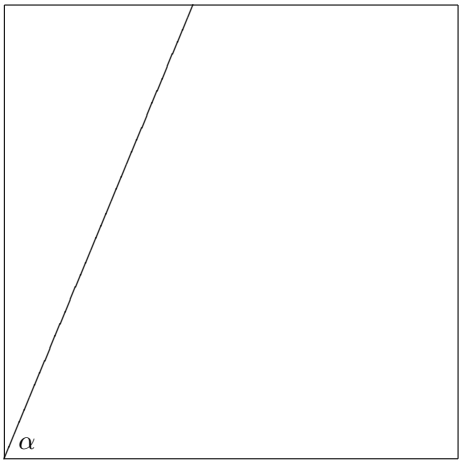
4.6.2 Příklad Metoda trisekce úhlu, kterou zde popíšeme, pochází z článku

☞ H. Abe, Trisection of Angle, *Saiensu*, říjen 1980, 8

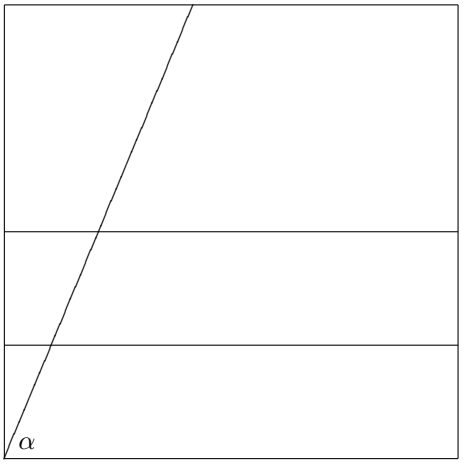
a my budeme postupovat podle stránek Thomase Hulla

☞ http://www.merrimack.edu/~thull/omfiles/geoconst.html

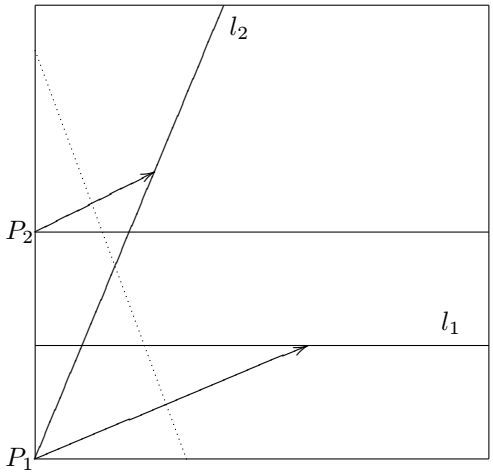
Připravme si list papíru a úhel, který chceme rozdělit na třetiny, označme α:



Přehněme papír napůl a pak ještě jednou. Dostaneme tak následující záhyby:

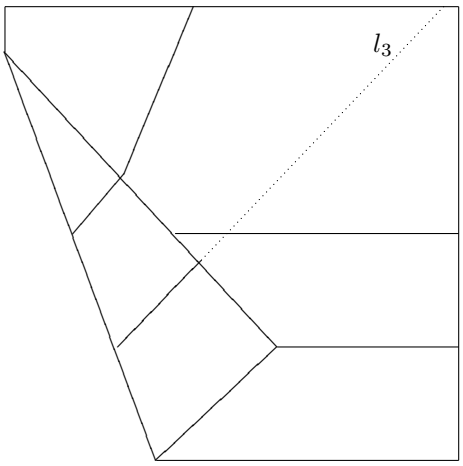


Označme nyní body P_1, P_2 a přímky l_1, l_2 tak, jak je to naznačeno na dalším obrázku a použijme axiom (O6):

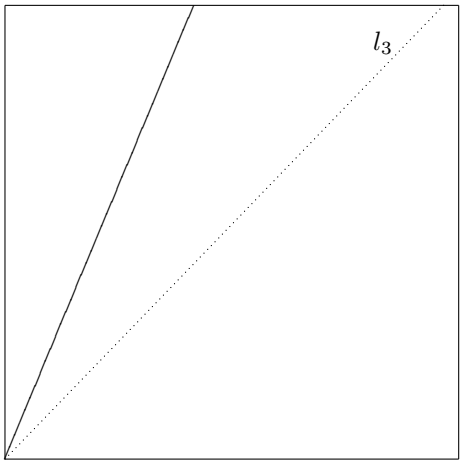


Podle záhybu z axiomu (O6) nyní papír přehneme a podél záhybu l_1 papír znovu přeložíme. Dostaneme tak

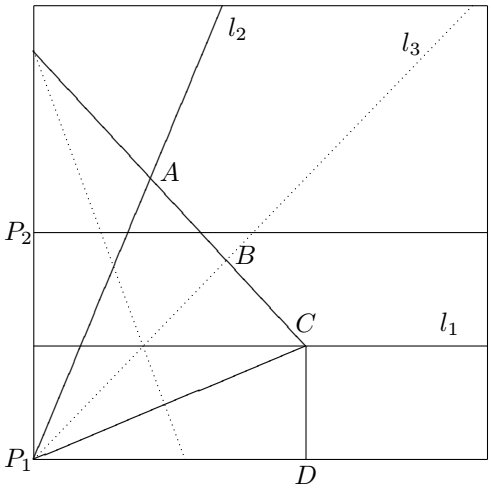
záhyb l_3:



Přímka l_3 potom vyznačuje úhel $\frac{2}{3}\alpha$:



Nyní dokážeme, že jsme skutečně rozdělili úhel α na třetiny. Označme všechny vzniklé body tak, jak je tomu na následujícím obrázku:



Pak je snadné ukázat, že trojúhelníky P_1AB, P_1BC a P_1CD jsou shodné a tudíž úhel DP_1B je roven $\frac{2}{3}\alpha$.

Velmi přístupnou formou jsou důkazy nemožnosti některých geometrických konstrukcí vyloženy v knize

☞ C. R. Haddock, *Field Theory and Its Classical Problems*, Mathematical Association of America, 1978

Zájemcům o matematicky přesný popis konstrukcí pomocí origami lze doporučit článek

☞ R. C. Alperin, A Mathematical Theory of Origami Constructions and Numbers, *New York J. Math.* 6 (2000), 119–133

a pro ukázku některých klasických skládánek origami například knihu

☞ P. Mulatinho, *Nápadité origami*, Knižní klub, 2001

Viz také cvičení 4.7.16 a 4.7.17.

Poznamenejme, že existuje klasická řecká úloha, kterou ani pomocí axiomů pro origami *nelze vyřešit*. Jde o *kvadraturu kruhu*.[7] Kvadratura kruhu je požadavek sestrojit stranu čtverce, který má stejný obsah jako zadaný kruh. Zřejmě stačí umět vyřešit kvadraturu kruhu o poloměru 1. To by znamenalo umět sestrojit úsečku délky $\sqrt{\pi}$. Číslo π je ovšem příkladem *transcendentního reálného čísla*, tj. čísla, které není řešením žádné algebraické rovnice s celočíselnými koeficienty. Axiomy pro origami však konstruují pouze (a to ještě ne všechna) *algebraická čísla*, tj. čísla, která jsou kořeny algebraických rovnic s celočíselnými koeficienty. Přesněji: čísla, zkonstruovatelná pomocí origami, tvoří nejmenší podtěleso tělesa reálných čísel, které je uzavřeno na druhé odmocniny kladných čísel a na reálné třetí odmocniny. Viz také cvičení 4.7.18.

4.7 Cvičení

4.7.1 Cvičení Ať n je přirozené číslo a ať $\mathbb{K}$ je okruh s jednotkou. Jako $\mathbb{K}[x]^{\leq n}$ označte množinu všech polynomů z $\mathbb{K}[x]$, které mají stupeň *nanejvýš* n, a jako $\mathbb{K}[x]^{=n}$ označte množinu všech polynomů z $\mathbb{K}[x]$, které mají stupeň *přesně* n.

Dokažte, že $\mathbb{K}[x]^{\leq n}$ má přesně k^{n+1} prvků a $\mathbb{K}[x]^{=n}$ má přesně $(k-1)k^n$ prvků, kdykoli má $\mathbb{K}$ přesně k prvků. (Návod: polynomy chápejte jako vektory, viz poznámka 4.1.2.)

Je $\mathbb{K}[x]^{\leq n}$ opět okruh? Pokud ano, dokažte to. Pokud ne, popište, proč ne.

4.7.2 Cvičení Ať $\mathbb{K}$ je komutativní okruh s jednotkou, ve kterém existují prvky $a, b \in \mathbb{K}$ takové, že $a \cdot b = 0$. Ukažte, že pak pro libovolná přirozená čísla n, m existují nad $\mathbb{K}$ polynomy $p(x)$, $q(x)$ tak, že $\deg(p(x)) = n$ a $\deg(q(x)) = m$ a platí $\deg(p(x) \cdot q(x)) < \deg(p(x)) + \deg(q(x))$.

Zkuste, nad nějakým $\mathbb{Z}_m$, najít polynomy $p(x)$, $q(x)$ stupně 1 tak, že $\deg(p(x) \cdot q(x)) = -\infty$.

4.7.3 Cvičení Ze cvičení 3.7.20 si připomeňte Hornerovo schéma, které počítá hodnotu polynomu $p(x)$ nad $\mathbb{R}$ v reálném čísle a. Hornerovo schéma zobecněte na algoritmus, který počítá hodnotu polynomu nad okruhem $\mathbb{K}$ v $a \in \mathbb{K}$. Dokažte totální korektnost tohoto algoritmu.

Pomocí Hornerova schématu spočítejte hodnoty

1. $5x^4 - 3x^2 + 7$ v $a = 6$, nad $\mathbb{Z}_8$.
2. $-7x^6 + 6x^4 - x^2 + 1$ v $a = 10$, nad $\mathbb{Z}_{11}$.

4.7.4 Cvičení Algoritmus pro dělení polynomů z příkladu 4.1.8 pracuje rekursivně. Napište tento algoritmus, nalezněte jeho variant a invariant a dokažte totální korektnost tohoto algoritmu.

4.7.5 Cvičení Dokažte, že relace

$a(x) \sim b(x)$ právě tehdy, když $a(x)$ a $b(x)$ jsou asociovány

je relace ekvivalence na množině všech polynomů.

4.7.6 Cvičení Podrobně vysvětlete, proč funguje (rozšířený) Eukleidův algoritmus pro polynomy. To znamená: nalezněte variant a invariant, dokažte totální korektnost.

4.7.7 Cvičení Zformulujte přesně Bezoutovu rovnost pro polynomy a naznačte její důkaz.

4.7.8 Cvičení Dokažte tvrzení 4.3.2.

[7] Stojí za zmínku, že v obecném jazyce tento obrat značí nemožnost něčeho.

4.7.9 Cvičení Vyřešte lineární rovnice:

1. $(x+1)\cdot p(x) = x^3$ v $\mathbb{Z}_3[x]/(x^2+1)$.
2. $(x^3+x+1)\cdot p(x) = (x^2+1)$ v $\mathbb{Z}_2[x]/(x^4+x+1)$.

Zformulujte a dokažte obecné tvrzení o řešitelnosti lineárních rovnic v konečném tělese. Srovnejte s větou 5.3.8.

4.7.10 Cvičení Teorii řešení *soustav* lineárních rovnic nad obecným konečným tělesem jsme nevybudovali. Všechna potřebná tvrzení z „klasické“ lineární algebry však pro soustavy nad konečným tělesem zůstávají v platnosti.

Nad tělesem $\mathbb{Z}_2[x]/(x^2+x+1)$ vyřešte (použitím GEM) soustavu lineárních rovnic:

$$\begin{array}{rcrcrcrcl} ax_1 & + & x_2 & + & x_3 & + & ax_4 & = & b \\ x_1 & + & bx_2 & + & ax_3 & + & ax_4 & = & 1 \end{array}$$

(Používáme značení z tabulky (4.1), tj. prvky $\mathbb{Z}_2[x]/(x^2+x+1)$ jsou 0, 1, a a b.)

4.7.11 Cvičení Ukažte, že žádný ireducibilní polynom nad tělesem $\mathbb{K}$ stupně > 1 nemá v $\mathbb{K}$ kořen. (Návod: použijte tvrzení 4.1.11.)

4.7.12 Cvičení Platí opak tvrzení 4.4.2? To znamená: platí, že jestliže $\mathbb{K}[x]/m(x)$ je těleso, pak je polynom $m(x)$ ireducibilní?

4.7.13 Cvičení Popište tabulku sčítání a násobení v GF(8).

4.7.14 Cvičení Ať $X = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ je množina. Existuje na X struktura tělesa?

4.7.15 Cvičení Nalezněte všechny cyklické kódy délky 6 nad $\mathbb{Z}_3$. Návod: použijte poznámku 4.5.5.

4.7.16 Cvičení Pomocí axiomů (O1)–(O6) pro origami dokažte, že je možné *duplikovat krychli*, tj., že je možné sestrojit krychli mající dvojnásobný objem než zadaná krychle. Zřejmě stačí umět duplikovat krychli o objemu 1, tj. sestrojit $\sqrt[3]{2}$. To je opět problém, který je pomocí kružítka a pravítka *neřešitelný*. Postupujte následovně:[8]

1. Vezměte si čtverec papíru (o straně dlouhé jednu jednotku délky).
2. Jako d_1, d_2 označte (v pořadí od spodní hrany papíru) záhyby, vzniklé přeložením daného papíru na třetiny. (Ukažte, že tuto konstrukci axiomy povolují.)
3. Jako F_1 označte pravý dolní roh papíru a jako F_2 označte průsečík záhybu d_1 s pravou svislou hranou papíru. Jako d_3 označte levou svislou hranu papíru. Nyní na F_1, F_2, d_1, d_3 použijte axiom (O6).
4. Přeložením nyní bod F_1 na svislé levé hraně papíru vymezuje dvě úsečky. Ukažte, že podíl jejich délek je $\sqrt[3]{2}$.

4.7.17 Cvičení Pouze pro milovníky geometrie. Vezměte si kružítko a pravítko a zvolte si jednotku délky. Tuto jednotku označte U. Dokažte následující:

1. Kružítkem a pravítkem lze sestrojit úsečku délky $r\cdot U$, kde r je libovolné kladné racionální číslo. (Návod: využijte vět o podobnostech trojúhelníků.)
2. Vymyslete interpretaci úseček záporné délky a ukažte, že kružítkem a pravítkem lze sestrojit úsečky délky $r\cdot U$, kde r je libovolné racionální číslo.
3. Označte jako $\mathbb{K}$ množinu všech zkonstruovatelných úseček a dokažte, že existují zkonstruovatelné délky, které *nejsou* tvaru $r\cdot U$ pro žádné racionální číslo r. (Návod: úhlopříčka čtverce.)
4. Dokažte, že $\mathbb{K}$ tvoří těleso. Jak se v $\mathbb{K}$ násobí a dělí?

[8] Tento elegantní postup je popsán v článku P. Messer, Problem 1054, *Crux Matematicorum* (12) č. 10 1986, 284–285.

Toto cvičení je základem důkazu neexistence trisekce úhlu a neexistence jiných geometrických konstrukcí, viz odstavec 4.6.

4.7.18 Cvičení V tomto cvičení dokážete, že „typické“ reálné číslo je transcendentní. Připomeňme definice:

1. Reálné číslo r je *algebraické*, když existuje polynom $p(x) \in \mathbb{Z}[x]$ tak, že $p(r) = 0$.
2. Reálné číslo r je *transcendentní*, pokud není algebraické.

Označte jako $\mathbb{A}$ množinu algebraických reálných čísel. Dokažte, že $\mathbb{A}$ je spočetná množina, tudíž množina $\mathbb{R} \setminus \mathbb{A}$ je nespočetná. Postupujte podle návodu (srovnejte s důkazem důsledku 1.5.6):

1. Dokažte, že $\mathbb{Z}^{\leq n}[x]$ (značení ze cvičení 4.7.1) je spočetná množina pro všechna n.
2. Protože platí $\mathbb{Z}[x] = \bigcup_{n\in\mathbb{N}} \mathbb{Z}^{\leq n}[x]$, je $\mathbb{Z}[x]$ spočetná množina.
3. Protože každý polynom $p(x) \in \mathbb{Z}[x]$ má pouze konečně mnoho kořenů, je množina $\mathbb{A}$ spočetná.

Revize kapitoly

Dozvěděli jsme se:

- ✔ Polynom lze definovat nad libovolným okruhem. S polynomy ovšem musíme obecně pracovat jako s výrazy.
- ✔ Práce s polynomy nad obecným tělesem se prakticky neliší od práce s polynomy nad reálnými čísly.
- ✔ Polynomy lze použít k rozšiřování číselných oborů. Ireducibilní polynomy hrají roli prvočísel a dovolují tak popsat všechna konečná tělesa.
- ✔ Algebraická rozšíření mají podstatný význam pro práci s cyklickými kódy.

Doplňující literatura

Seznam doplňující literatury týkající se základních vlastností polynomů, algebraických rozšíření a cyklických kódů je shodný se seznamem pro kapitolu 2.

Kapitola 5

Abstraktní výpočty

> Teorie grup je odvětví matematiky, kde člověk něco s něčím udělá a pak porovná výsledek s tím, když totéž udělá s něčím jiným nebo když s tímtéž udělá něco jiného.
>
> James Newman

V této (a následující) kapitole zvýšíme úroveň abstrakce. Zatím jsme pokročili od, řekněme, počítání v $\mathbb{Z}$ k počítání v $\mathbb{Z}_m$ (kapitoly 2 a 3). (O počítání v obecném konečném tělese se lze dozvědět v kapitole 4.) Viděli jsme například, že lineární algebra nad tělesem reálných čísel je speciálním případem lineární algebry nad obecným tělesem. Důležité bylo *být tělesem*. Požadované vlastnosti jsme shrnuli do definic (viz definice 2.3.14 a 2.3.17), ve kterých jsme *specifikovali operace* (sčítání a násobení) a jejich *rovnicové vlastnosti* (například asociativitu sčítání).

Chceme nyní množiny s operacemi splňujícími jisté rovnice studovat v plné obecnosti. Matematická disciplína, která se těmito věcmi zabývá, se jmenuje *universální algebra*.[1] Abychom zdůvodnili důležitost universální algebry pro computer science, ukážeme, jak ji lze aplikovat v oblasti *algebraických specifikací datových typů*.

5.1 Krátká návštěva u algebraických specifikací

Předpokládejme, že chceme vědět, co je seznam (jako datový typ). Přesněji, chceme *specifikovat* datový typ `LIST` a chceme říci abstraktně, co znamená, že můžeme například dva prvky tohoto datového typu zřetězit a dostat tak opět seznam.

```
1    spec LIST is
2         sorts: list
3         operations: nil: --> list;
4                      _#_:list,list --> list
5         variables: x,y,z:list                                    (5.1)
6         equations: x#nil=x;
7                    nil#x=x;
8                    x#(y#z)=(x#y)#z
9    endspec
```

Výše uvedená notace je jen zápis v *pseudokódu* (připomínajícím specifikační jazyk OBJ3). Význam jednotlivých řádků je následující:

1. Řádky `1` a `9` definují specifikaci datového typu `LIST`.

2. Řádek `2` říká, že ve hře je jen jedna *sorta*[2] věcí, této sortě říkáme `list`.

[1] Zájemcům doporučujeme jedinečnou knihu: W. Wechler, *Universal Algebra for Computer Scientists*, Springer-Verlag, Berlin, 1992.

[2] V tomto kontextu se anglické *sort* také překládá jako *druh*. Jazykoví puristé odpusťte: mně se to nelíbí.

3. Na řádcích `3` a `4` zavádíme dvě *operace*, spolu s jejich *aritami*.

   Operace `nil` nevyžaduje žádný argument (má *aritu nula*) a vrací prvek sorty `list`. Takovou operaci si samozřejmě můžeme představit jako *konstantu* `nil` sorty `list`. (Intuice: `nil` je prázdný seznam.)

   Operace `_#_` vyžaduje dva argumenty, oba sorty `list` (tj. má *aritu dva*, neboli je *binární*) a vrací prvek sorty `list`. (Intuice: `_#_` zřetězuje seznamy.)

4. Řádek `5` deklaruje *proměnné* `x`, `y`, `z` sorty `list`, které jsou použity v *rovnicích* (řádky `6` až `8`).

   Řádky `6` a `7` říkají, že se `nil` chová jako *neutrální prvek* pro `_#_` a řádek `8` říká, že `_#_` je *asociativní*.

Výše uvedené tak zřejmě popisuje, jak se má chovat prázdný seznam vzhledem ke zřetězování a že zřetězování seznamů by mělo být asociativní.

Specifikace `LIST` má řadu *modelů*, sice jakoukoli množinu vybavenou asociativní binární operací s neutrálním prvkem. Takže následující jsou příklady modelů specifikace `LIST`:

1. Množina Σ^* všech slov nad konečnou abecedou Σ, spolu s prázdným slovem ε a zřetězováním slov jako s binární operací.

2. Množina $\mathbb{N}$ všech přirozených čísel, spolu s binární operací sčítání a číslem 0 jako neutrálním prvkem.

   Při tomto pohledu je každé přirozené číslo „seznam“, číslo 0 je „prázdný seznam“ a dva „seznamy“ zřetězujeme sčítáním. Sčítání na $\mathbb{N}$ je samozřejmě asociativní a má 0 jako neutrální prvek.

3. Množina F všech funkcí $f : \mathbb{R} \longrightarrow \mathbb{R}$, spolu s funkcí id (identita) a s binární operací skládání funkcí.

   Při tomto pohledu je každá funkce $f : \mathbb{R} \longrightarrow \mathbb{R}$ „seznam“, identita id je „prázdný seznam“ a dva „seznamy“ zřetězujeme, když je skládáme jako funkce. Skládání funkcí je asociativní binární operace a funkce id je její neutrální prvek.

4. A mnoho dalších...

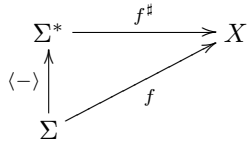
Samozřejmě, pouze první příklad (množina Σ^*) odráží naší představu seznamů nejlépe. Ve skutečnosti jde o *seznamy prvků typu* Σ. Co dělá z množiny Σ^* tak zvláštní model specifikace `LIST`? Ukazuje se, že existuje jistá *universální vlastnost*, která vyděluje Σ^* ze všech modelů specifikace `LIST`, které *interpretují prvky množiny* Σ.

Tuto universální vlastnost nyní přesně zformulujeme:

1. Existuje *kanonická* interpretace $\langle - \rangle$, která každý prvek typu Σ interpretuje jako prvek typu Σ^*.

   Touto interpretací je zobrazení
   $$\langle - \rangle : \Sigma \longrightarrow \Sigma^*$$
   které prvek $a \in \Sigma$ posílá na slovo $\langle a \rangle \in \Sigma^*$. (Intuitivně: každé písmeno lze chápat kanonicky jako slovo.)

2. Výše uvedená interpretace je „universální“: kdykoli máme model specifikace `LIST`, tj. kdykoli si vezmeme množinu X spolu s asociativní binární operací $\star$ a s jejím neutrálním prvkem $e \in X$ a kdykoli si vezmeme funkci
   $$f : \Sigma \longrightarrow X$$
   (o funkční hodnotě $f(a)$ přemýšlíme jako o tom „seznamu“ v množině X, který je přiřazen prvku $a \in \Sigma$), pak existuje jediné zobrazení
   $$f^\sharp : \Sigma^* \longrightarrow X$$
   (tj. pak můžeme každé slovo z Σ^* interpretovat jako „seznam“ v X) takové, že platí rovnost
   $$f^\sharp \circ \langle - \rangle = f$$
   Tato rovnost říká, že $f^\sharp$ interpretuje slova tvaru $\langle a \rangle$ stejně, jako f interpretuje $a \in \Sigma$: pro každé $a \in \Sigma$ platí rovnost $f^\sharp(\langle a \rangle) = f(a)$. Tuto rovnost můžeme vyjádřit pomocí diagramu — říkáme, že následující diagram



komutuje.

Zobrazení $f^\sharp$ navíc *respektuje prázdné seznamy* a *respektuje zřetězování*. Což znamená, že platí rovnosti

$$f^\sharp(\varepsilon) = e \tag{5.2}$$

a

$$f^\sharp(w_1 w_2) = f^\sharp(w_1) \star f^\sharp(w_2) \tag{5.3}$$

Zobrazení $f^\sharp$ můžeme velmi elegantně popsat:[3]

$$\begin{aligned} f^\sharp(\varepsilon) &= e \\ f^\sharp(wa_1) &= f^\sharp(w)a_1 \\ f^\sharp(wa_2) &= f^\sharp(w)a_2 \\ &\vdots \\ f^\sharp(wa_n) &= f^\sharp(w)a_n \end{aligned}$$

a pak použít strukturální indukci k důkazu, že platí rovnosti (5.2) a (5.3).

Vidíme, že ta zobrazení mezi modely specifikace `LIST`, která respektují prázdné seznamy a respektují zřetězování, jsou důležitá. Jak uvidíme, jsou tato zobrazení přirozenou volbou zobrazení, která nám umožňují dva modely specifikace `LIST` porovnat.

Co myslíme „porovnáváním modelů“ specifikace `LIST`? Předpokládejme, že máme dva modely specifikace `LIST`, řekněme X, $\star_X$, e_X a Y, $\star_Y$, e_Y. To znamená, že o prvcích X přemýšlíme jako o jedné z řady možných *implementací* seznamů, $\star_X$ je způsob, jakým se seznamy zřetězují a e_X je význačný prvek množiny X, který representuje prázdný seznam. O trojici Y, $\star_Y$, e_Y přemýšlíme jako o jiné implementaci. *Každé* zobrazení

$$f : X \longrightarrow Y$$

vytváří ze seznamu $x \in X$ seznam $f(x) \in Y$. Nicméně některá zobrazení jsou „lepší než ostatní“. Pokud o zobrazení f chceme přemýšlet jako o překladu „světa seznamů X“ do „světa seznamů Y“, mělo by platit následující:

$$f(e_X) = e_Y \quad \text{a} \quad f(x_1 \star_X x_2) = f(x_1) \star_Y f(x_2), \text{ pro všechna } x_1, x_2 \in X$$

První rovnice říká: prázdný seznam přeložte jako prázdný seznam. Druhá rovnice říká: zřetězte dva seznamy v X a výsledek přeložte do Y. Dostanete stejný výsledek, jako když nejprve oba seznamy přeložíte a poté je v Y zřetězíte.

Ukážeme nyní realističtější příklad algebraické specifikace — specifikujeme *zásobníky*, do kterých můžeme ukládat prvky čtyřprvkové množiny.

```
1    spec STACK is
2         sorts: alphabet, stack
3         operations: a1: --> alphabet;
4                     a2: --> alphabet;
5                     a3: --> alphabet;
6                     a4: --> alphabet;
7                     nil: --> stack
8                     pop(_): stack --> stack
9                     push(_,_): alphabet,stack --> stack
10        variables: x:alphabet, s:stack
11        equations: pop(nil)=nil;
12                   pop(push(x,s))=s
13   endspec
```
(5.4)

Zápis je velmi podobný specifikaci `LIST`, proto vysvětlíme pouze ty řádky, které nejsou na první pohled zřejmé:

[3] Důvodem je to, že Σ^* je množina zadaná *induktivně*, viz cvičení 1.6.28.

1. Na řádku `2` specifikujeme, že budeme pracovat se *dvěma* sortami: s prvky sorty `alphabet` (ty hodláme ukládat na zásobník) a sorty `stack` (to jsou jednotlivé zásobníky).

2. Dále specifikujeme *operace*. Jediný rozdíl je v tom, že musíme dávat pozor na arity: například na řádku `9` *neříkáme* pouze, že operace `push(_,_)` si bere dva argumenty; říkáme také, že první z nich musí být sorty `alphabet` a druhý sorty `stack`! Dále říkáme, že výsledek operace je sorty `stack`. Proto je ve vícesortovém případě pojem *arita* složitější než v jednosortovém, viz definice 6.4.3.

3. Proměnné deklarujeme na řádku `10`. Tyto proměnné jsou opět různých sort.

Jak vypadá obecný model X specifikace `STACK`? Především by se měl odrazit fakt, že jsme specifikovali dvě sorty: `alphabet` a `stack`. Proto se model sestává ze dvou disjunktních množin

$$X_a \quad \text{a} \quad X_s$$

Prvky X_a jsou sorty `alphabet`, prvky X_s jsou sorty `stack`. Přejděme k operacím:

1. Specifikace operací `a1`, ..., `a4` nám říká, že musíme v množině X_a zvolit speciální prvky a_1, a_2, a_3 a a_4. (To jsou ty čtyři prvky, které budeme ukládat na zásobník.)

2. Specifikace `nil` nám říká, že máme zvolit speciální prvek $n \in X_s$. (To je prázdný zásobník v našem modelu.)

3. Specifikace operace `pop(_):stack --> stack` říká, že máme zadat zobrazení $p : X_s \longrightarrow X_s$. (Pro „zásobník“ $x \in X_s$ je výsledek $p(x)$ ten „zásobník“ v X_s, který dostaneme, když z x odstraníme vrchní prvek.)

4. Specifikace operace `push(_,_):alphabet,stack --> stack` říká, že máme funkci $u : X_a \times X_s \longrightarrow X_s$. (Pro „písmeno“ $l \in X_a$ a „zásobník“ $x \in X_s$ je $u(l,x)$ ten „zásobník“, který dostaneme uložením l na vrchol x.)

Navíc musí samozřejmě platit specifikované rovnice:

$$p(n) = n \quad \text{a} \quad p(u(l,x)) = x \text{ pro všechna } l \in X_a \text{ a všechna } x \in X_s$$

První rovnice říká, že prázdný zásobník zůstane po vyjmutí vrchního prvku prázdným (zvolili jsme co nejjednodušší přístup). Druhá rovnice říká, že když nejprve na zásobník x uložíme písmeno l a poté je opět vyjmeme, dostaneme původní zásobník x.

Můžeme opět dát příklady modelů specifikace `STACK` a opět budeme mít pocit, že některé modely vystihují naši představu o zásobnících lépe než jiné. Příslušnou universální vlastnost „zamýšleného“ modelu specifikace `STACK` nebudeme formulovat, to uděláme později v odstavci 6.4.

5.2 Co budeme dělat?

Předchozí odstavec nám dal jistý cit pro to, co je algebraická specifikace a její model. Zopakujme to:

1. Specifikujeme sorty.

2. Specifikujeme jména operací a jejich arity.

3. Specifikujeme rovnice, které mají operace splňovat.

4. Studujeme modely specifikace a „přirozená“ zobrazení, která nám dovolují jednotlivé modely porovnávat.

5. Některé (důležité) modely mohou být popsány určitou universální vlastností.

Soustředíme se na jednosortový případ a postup v případech s více sortami naznačíme v odstavci 6.4.

Finitární typ, *algebry* (tj. modely) a *homomorfismy* definujeme v plné obecnosti v odstavci 5.4.

Naše úvahy vyvrcholí specifikací tří různých situací:

1. Situace, kdy chceme popsat základní chování násobení a dělení. Výsledkem je pojem *grupy* — definice 5.3.6. Uvidíme však, že na grupy se můžeme též dívat jako na popis pohybů (příklad 5.3.26) nebo na popis chování zlomků (cvičení 5.7.13).

2. Situace, kdy chceme popsat základní chování logických spojek `and` a `or`. Výsledkem bude pojem *svazu* — příklad 5.4.7. Svazy však také modelují situaci, kdy počítáme suprema a infima dvojic prvků — viz odstavec 6.1.

3. Situace, kdy chceme specifikovat celou sílu (klasických) pravdivostních hodnot. Výsledkem je pojem *Booleovy algebry* — definice 6.3.3.

Je před námi dlouhá cesta a proto začneme velmi zvolna, skromnou specifikací jediné binární operace.

5.3 Jednoduché algebraické struktury

Vůbec nejjednodušší (jednosortová) algebraická specifikace je ta, ve které *nespecifikujeme žádné operace* (a tím pádem ani *žádné rovnice*). Napišme tuto specifikaci:

```
spec NO_OPERATIONS is
     sorts: something
endspec
```
(5.5)

Je jasné, že modely této specifikace jsou *přesně množiny*. Viz také poznámku 5.3.13. Začněme něčím o málo složitějším — specifikací jedné binární operace:

```
spec BINARY is
     sorts: binary
     operations: _*_:binary,binary --> binary
endspec
```
(5.6)

5.3.1 Definice *Binární operace* na množině X je zobrazení $\star\colon X \times X \longrightarrow X$. Namísto $\star\,(x,y)$ budeme pro hodnotu $\star$ v (x,y) používat obvyklý infixní zápis $x \star y$.

Dvojici $\langle X, \star\rangle$ budeme říkat *grupoid*.

Grupoid, jakožto model specifikace `BINARY`, je tedy pouze množina vybavená nějakou binární operací. Zdůrazněme ale, že po binární operaci chceme, aby byla definována pro *každou* dvojici prvků. Proto například dělení na reálných číslech, tj. předpis

$$(x,y) \mapsto \frac{x}{y} \qquad \text{pro } y \neq 0$$

není binární operace na množině $\mathbb{R}$.

Ukážeme příklady grupoidů.

5.3.2 Příklady

1. Na prázdné množině $X = \emptyset$ existuje jediná binární operace, totiž prázdné zobrazení $\emptyset : \emptyset \times \emptyset \longrightarrow \emptyset$. Proto $\langle \emptyset, \emptyset\rangle$ je příklad grupoidu.

2. Na množině $X = \{a,b,c\}$ definujeme binární operaci $\star$ takto: $x \star y = x$ pro všechna $x, y \in X$. Protože X je konečná množina, můžeme operaci $\star$ popsat následující tabulkou:

$\star$	a	b	c
a	a	a	a
b	b	b	b
c	c	c	c

(5.7)

   Je-li x v i-tém řádku a y v j-tém sloupci tabulky, pak v položce (i,j) je zapsán výsledek $x \star y$.

   Dvojice $\langle X, \star\rangle$ je grupoid.

3. Ať X je množina všech zobrazení z $\{0,1\}$ do $\{0,1\}$. Množina X má čtyři prvky:

$$f_1: \begin{array}{l} 0 \mapsto 0 \\ 1 \mapsto 1 \end{array} \qquad f_2: \begin{array}{l} 0 \mapsto 0 \\ 1 \mapsto 0 \end{array}$$

$$f_3: \begin{array}{l} 0 \mapsto 1 \\ 1 \mapsto 1 \end{array} \qquad f_4: \begin{array}{l} 0 \mapsto 1 \\ 1 \mapsto 0 \end{array}$$

Skládání funkcí $\circ$ je binární operace na množině X, a proto je $\langle X, \circ \rangle$ grupoid. Příslušná tabulka je:

$\circ$	f_1	f_2	f_3	f_4
f_1	f_1	f_2	f_3	f_4
f_2	f_2	f_2	f_2	f_2
f_3	f_3	f_3	f_3	f_3
f_4	f_4	f_3	f_2	f_1

(5.8)

Při studiu grupoidů $\langle X, \star \rangle$ nevyžadujeme, aby operace $\star$ měla nějaké speciální vlastnosti, žádné rovnice jsme ve specifikaci `BINARY` nepožadovali. Vyjmenujeme nyní některé vlastnosti, které nás budou u binárních operací zajímat. Povšimněme si, že všechny jsou rovnicové povahy. (Viz cvičení 5.7.4.)

5.3.3 Definice Ať $\star$ je binární operace na množině X.

1. Operace $\star$ je *asociativní*, pokud pro všechna $x, y, z \in X$ platí rovnost $x \star (y \star z) = (x \star y) \star z$.
2. Operace $\star$ je *komutativní*, pokud pro všechna $x, y \in X$ platí rovnost $x \star y = y \star x$.
3. Prvek e_l je *levý neutrální prvek* operace $\star$, pokud pro všechna $x \in X$ platí rovnost $e_l \star x = x$.
4. Prvek e_r je *pravý neutrální prvek* operace $\star$, pokud pro všechna $x \in X$ platí rovnost $x \star e_r = x$.
5. Prvek e je *neutrální prvek* operace $\star$, pokud je pravým i levým neutrálním prvkem, tj. když pro všechna $x \in X$ platí rovnost $e \star x = x \star e = x$.

5.3.4 Příklad Operace $\star$ z tabulky (5.7) je asociativní, protože podle definice této operace pro všechna $x, y, z \in \{a, b, c\}$ platí $x \star (y \star z) = x$ a $(x \star y) \star z = x \star y = x$. Tato operace není komutativní, protože například platí $a = a \star b \neq b \star a = b$. Každý prvek množiny $\{a, b, c\}$ je pravým neutrálním prvkem operace $\star$: například a je pravý neutrální prvek, protože pro všechna $x \in \{a, b, c\}$ platí $x \star a = x$. Podobně lze ukázat, že b i c jsou pravé neutrální prvky. Operace $\star$ však nemá žádný levý neutrální prvek, a tudíž žádný neutrální prvek.

Operace $\circ$ skládání funkcí na dvouprvkové množině popsaná v tabulce (5.8) je asociativní, má neutrální prvek $e = f_1$, ale není komutativní, protože platí například $f_2 = f_2 \circ f_3 \neq f_3 \circ f_2 = f_3$.

Abstraktním výpočtem budeme rozumět využívání vlastností operací a rovností k odvozování různých vztahů. Uveďme příklad abstraktního výpočtu.

5.3.5 Příklad Ukážeme následující tvrzení:

> *Jestliže binární operace $\star$ na množině X má levý neutrální prvek e_l a pravý neutrální prvek e_r, pak platí $e_l = e_r$.*

Postupujeme takto: protože e_l je levý neutrální prvek, platí pro každé $x \in X$ rovnost $e_l \star x = x$. Speciálně tedy platí rovnost $e_l \star e_r = e_r$. Levá strana této rovnosti ale je rovna prvku e_l, protože e_r je pravý neutrální prvek. Celkově tedy dostáváme $e_l = e_r$, a to jsme chtěli ukázat.

Jako důsledek okamžitě dostáváme:

> *Každá binární operace má nanejvýš jeden neutrální prvek.*

To plyne z toho, že pokud e a e' jsou neutrální prvky, potom e je pravý neutrální prvek a e' je levý neutrální prvek, a proto podle předchozího musí platit rovnost $e = e'$.

5.3.6 Definice Ať $\langle X, \star \rangle$ je grupoid.

1. $\langle X, \star\rangle$ je *pologrupa*, je-li $\star$ asociativní operace.
2. Pologrupě $\langle X, \star\rangle$ říkáme *monoid*, jestliže operace $\star$ má neutrální prvek.
3. Monoidu $\langle X, \star\rangle$ s neutrálním prvkem e říkáme *grupa*, jestliže má každý prvek x *inversi vzhledem k* $\star$, tj. jestliže platí:

   Pro každé x existuje právě jedno x^{-1} takové, že platí $x^{-1} \star x = e = x \star x^{-1}$.

Samozřejmě, každá grupa je monoid, každý monoid je pologrupa a každá pologrupa je grupoid. Po dané struktuře totiž požadujeme postupně víc a víc. Žádnou z těchto implikací však nelze obrátit, jak ukazuje následující příklad.

5.3.7 Příklad

1. Grupoid, který není pologrupa. Na množině $\mathbb{N}$ definujte operaci $\star$ takto: $n \star m = n^m$. Jde skutečně o operaci, připomeňme totiž, že n^m je definováno pro všechny dvojice přirozených čísel n, m jako počet všech zobrazení z m-prvkové množiny do n-prvkové množiny. Protože existuje jediné zobrazení z prázdné množiny do prázdné množiny (sice prázdné zobrazení), je definováno i 0^0 a platí $0^0 = 1$.

   Protože platí
   $$2^{27} = 2 \star (3 \star 3) \neq (2 \star 3) \star 3 = 2^9$$
   není binární operace $\star$ asociativní.
2. Pologrupa, která není monoid. Takovým příkladem je pologrupa definovaná tabulkou (5.7).
3. Monoid, který není grupa. Operace násobení na přirozených číslech je asociativní a má neutrální prvek, a sice číslo 1. Prvek 2 však nemá inversi, protože neexistuje přirozené číslo x, pro které platí $x \cdot 2 = 1 = 2 \cdot x$.

Grupy souvisejí s otázkou jednoznačného řešení lineárních rovnic. To nám umožní první pohled na grupu: je to monoid (tj. struktura s „rozumným“ násobením), kde lze *jednoznačně řešit lineární rovnice*. Přesněji, platí následující věta:

5.3.8 Věta *Ať $\langle X, \star, e\rangle$ je monoid. Pak je ekvivalentní:*

1. *Každá rovnice $a \star x = b$, kde a, b jsou libovolné prvky X, má právě jedno řešení.*
2. *Každá rovnice $a \star x = e$, kde a je libovolný prvek X, má právě jedno řešení.*
3. *Na X lze definovat operaci $(-)^{-1} : X \longrightarrow X$ tak, že platí rovnosti $a \star a^{-1} = a^{-1} \star a = e$, pro libovolné $a \in X$. To znamená, že $\langle X, \star, e, (-)^{-1}\rangle$ je grupa.*

Důkaz. Z 1. plyne 2.: To je triviální.

Ze 2. plyne 3.: Pro $a \in X$ definujeme a^{-1} jako jediné řešení rovnice $a \star x = e$. Potom samozřejmě platí rovnost $a \star a^{-1} = e$. Abychom ukázali, že platí i rovnost $a^{-1} \star a = e$, stačí ukázat, že obě strany této rovnosti řeší rovnici $a \star x = a$. Neutrální prvek e zřejmě tuto rovnici řeší. Ukážeme, že platí i $a \star (a^{-1} \star a) = a$:

$$\begin{array}{rcll} a \star (a^{-1} \star a) & = & (a \star a^{-1}) \star a & \text{(asociativita } \star\text{)} \\ & = & e \star a & \text{(definice } a^{-1} \text{ jakožto jediného řešení } a \star x = e\text{)} \\ & = & a & \text{(neutralita } e\text{)} \end{array}$$

Ze 3. plyne 1.: Předpokládejme, že je zadána lineární rovnice $a \star x = b$. Nejdříve ukážeme, že prvek $a^{-1} \star b$ tuto rovnici řeší:

$$\begin{array}{rcll} a \star (a^{-1} \star b) & = & (a \star a^{-1}) \star b & \text{(asociativita } \star\text{)} \\ & = & e \star b & \text{(vlastnost } a^{-1}\text{, kterou předpokládáme)} \\ & = & b & \text{(neutralita } e\text{)} \end{array}$$

Zbývá ukázat, že jiné řešení rovnice $a \star x = b$ nemá. Předpokládejme, že nějaký prvek u tuto rovnici řeší, tj. platí $a \star u = b$. Potom platí rovnost $a^{-1} \star (a \star u) = a^{-1} \star b$. Použitím asociativity $\star$ dostáváme rovnost $(a^{-1} \star a) \star u = a^{-1} \star b$. Z definice operace $(-)^{-1}$ plyne další rovnost $e \star u = a^{-1} \star b$. Protože e je neutrální,

platí $u = a^{-1} \star b$, a to jsme chtěli dokázat. ■

Zaměříme se nyní na studium „dobrých“ zobrazení mezi jednotlivými strukturami. Z úvodu této kapitoly víme, že taková zobrazení nám umožní jednotlivé struktury porovnávat. Začneme pojmem „dobré“ podmnožiny:

5.3.9 Příklad Uvažujme o binární operaci + (sčítání) na množině $\mathbb{R}$ všech reálných čísel. Některé podmnožiny A množiny $\mathbb{R}$ jsou na sčítání „uzavřené“ v tom smyslu, že sčítáme-li dvě čísla z množiny A, padne výsledek opět do množiny A. Příklady takových množin jsou:

1. $A = \mathbb{N}$. Součet dvou přirozených čísel (v oboru reálných čísel) je přirozené číslo.
2. $A = \mathbb{Z}$. Součet dvou celých čísel (v oboru reálných čísel) je celé číslo.
3. $A = \mathbb{Q}$. Součet dvou racionálních čísel (v oboru reálných čísel) je racionální číslo.

Příkladem podmnožiny $A \subseteq \mathbb{R}$, která *není uzavřená* na operaci sčítání je množina všech iracionálních čísel. Sečteme-li například dvě iracionální čísla $\sqrt{2}$ a $-\sqrt{2}$, potom výsledek (číslo 0) není iracionální číslo.

5.3.10 Příklad Ukážeme nyní dva důležité příklady grup, které vznikají jako podmožiny uzavřené na operaci:

1. Označme jako $\mathsf{Mat}_{n\times n}(\mathbb{R})$ množinu všech matic rozměrů $n \times n$. Spolu s násobením matic tato množina tvoří jistě monoid: násobení matic je asociativní a má za neutrální prvek jednotkovou matici $\mathbb{E}$. Symbolem

   $$\mathsf{RegMat}_{n\times n}(\mathbb{R})$$

   označte podmnožinu regulárních matic. Potom je množina $\mathsf{RegMat}_{n\times n}(\mathbb{R})$ uzavřená na násobení v množně $\mathsf{Mat}_{n\times n}(\mathbb{R})$, protože součin regulárních matic je regulární. Protože jednotková matice $\mathbb{E}$ rozměrů $n \times n$ je regulární, tvoří regulární matice grupu.

2. Množina $\mathbb{Z}_m$ spolu s operací násobení je jistě monoid (násobení modulo m je asociativní a má neutrální prvek 1). Přesto $\mathbb{Z}_m$ *nikdy není grupa*, protože číslo 0 *nikdy nemá inversi*. Definujme množinu

   $$\mathbb{Z}_m^*$$

   jako množinu těch prvků v $\mathbb{Z}_m$, které inversi vzhledem k násobení mají. (Množina $\mathbb{Z}_m^*$ má přesně $\varphi(m)$ prvků. Proč?)

   Množina $\mathbb{Z}_m^*$ je zjevně uzavřená v $\mathbb{Z}_m$ na násobení, součin dvou invertibilních prvků je invertibilní, viz cvičení 3.7.2. Protože $1 \in \mathbb{Z}_m^*$, je $\mathbb{Z}_m^*$ grupa.

Tyto příklady lze zobecnit, viz cvičení 5.7.11.

Uzavřenost na operaci je užitečná vlastnost a nyní ji definujeme obecně.

5.3.11 Definice Ať $\langle X, \star\rangle$ je grupoid. Řekneme, že podmnožina $A \subseteq X$ je *uzavřená na operaci* $\star$, když pro všechna $x, y, \in A$ platí $x \star y \in A$.

Povšimněme si, že podle definice je *prázdná množina vždy uzavřena na jakoukoli binární operaci*: pro každou dvojici $x, y \in \emptyset$ totiž chceme ukázat, že $x \star y \in \emptyset$. Tato vlastnost je triviálně splněna, protože $\emptyset$ nemá žádné prvky: neexistuje totiž dvojice $x, y \in \emptyset$, pro kterou by neplatilo $x \star y \in \emptyset$.

Máme-li podmnožinu $A \subseteq X$ uzavřenou v grupoidu $\langle X, \star\rangle$ na operaci $\star$, potom lze na množině A definovat binární operaci $\star_A$: $A \times A \longrightarrow A$ takto:

$$x \star_A y = x \star y \quad \text{pro všechna } x, y \in A$$

Operaci $\star_A$ se říká *zúžení operace* $\star$ *na množinu* A a dvojici $\langle A, \star_A\rangle$ potom říkáme *podgrupoid grupoidu* $\langle X, \star\rangle$. Namísto $\star_A$ budeme psát opět $\star$, pokud to nepovede k nedorozumění.

5.3.12 Příklad Každá podmnožina $A \subseteq \{a, b, c\}$ je uzavřená na operaci $\star$, popsanou v tabulce (5.7). Pro neprázdnou podmnožinu A to plyne okamžitě z definice operace $\star$ — zvolme $x, y \in A$. Potom $x \star y = x$ a tudíž $x \star y \in A$. Je-li $A = \emptyset$, pak využijeme toho, že prázdná množina je uzavřena na jakoukoli binární operaci.

Následující podmnožiny A jsou uzavřené na binární operaci na množině $\{f_1, f_2, f_3, f_4\}$ popsané tabulkou (5.8):

1. $A = \emptyset$.
2. Kromě množiny $\{f_4\}$ jsou všechny jednoprvkové podmnožiny uzavřené na operaci $\circ$. Množina $\{f_4\}$ uzavřená není, protože platí $f_4 \circ f_4 = f_1$.
3. Z dvouprvkových podmnožin jsou uzavřené na operaci $\circ$ všechny, kromě množin $\{f_2, f_4\}$ (protože $f_4 \circ f_2 = f_3$) a $\{f_3, f_4\}$ (protože $f_4 \circ f_3 = f_2$).
4. Všechny tříprvkové podmnožiny kromě množin $\{f_1, f_2, f_4\}$, $\{f_2, f_3, f_4\}$ a $\{f_1, f_3, f_4\}$ jsou uzavřené na operaci $\circ$.
5. Samozřejmě, celá množina $\{f_1, f_2, f_3, f_4\}$ je uzavřená na operaci $\circ$.

5.3.13 Poznámka Uzavřenost podmnožiny $A \subseteq X$ na binární operaci znamená, že jsme v X vybrali „dobrou“ podmnožinu A. Množina A totiž „vydrží“ operování se svými prvky. Takové příklady znáte například z lineární algebry: prvky *podprostoru* musí „vydržet“ sčítání a násobení skalárem, neboli podprostor musí být *uzavřený* na sčítání a násobení skalárem.

Předvedeme nyní, jak lze v tomto kontextu chápat podmožiny množin. Každá podmnožina A množiny X je „dobrá“. To plyne z toho, že množiny jsou přesně modely jednosortové specifikace kde nespecifikujeme *žádné* operace (tím pádem ani žádné rovnice), viz specifikaci (5.5). Od modelu nechceme tedy vůbec nic a proto nechceme nic ani po podmnožinách. Velmi užitečný pohled je tento: množina je „pytel písku“.[4] Pytel písku ovšem nemá vnitřní strukturu, proto jakýkoli „kus“ tohoto pytle je opět pytlem písku, čili množinou.

Jiná situace je, kdy zrnka písku se slepila vlhkem do hrudek. Pytel písku pak má *strukturu*. Pokud tuto strukturu nechceme porušit, musíme z pytle odebírat pouze jednotlivé hrudky, nemáme dovoleno jednotlivé hrudky „drobit“. V tomto případě již tedy není „dobrá“ každá podmnožina. Viz také poznámku 5.5.6.

Uzavřenost na operaci je speciálním případem respektování binární operace. Zobrazením, která respektují binární operace, se říká homomorfismy grupoidů.

5.3.14 Definice Ať $\langle X, \star_X \rangle$ a $\langle Y, \star_Y \rangle$ jsou grupoidy. Zobrazení $f : X \longrightarrow Y$ je *homomorfismus grupoidů* $z\ \langle X, \star_X \rangle$ *do* $\langle Y, \star_Y \rangle$, pokud následující diagram

$$\begin{array}{ccc} X \times X & \xrightarrow{f \times f} & Y \times Y \\ {\scriptstyle \star_X}\downarrow & & \downarrow{\scriptstyle \star_Y} \\ X & \xrightarrow[f]{} & Y \end{array} \tag{5.9}$$

komutuje, tj. pokud rovnost

$$f(x \star_X y) = f(x) \star_Y f(y)$$

platí pro všechna $x, y \in X$.

Fakt, že $f : X \longrightarrow Y$ je homomorfismus grupoidů z $\langle X, \star_X \rangle$ do $\langle Y, \star_Y \rangle$ budeme značit

$$f : \langle X, \star_X \rangle \longrightarrow \langle Y, \star_Y \rangle$$

5.3.15 Příklad Množina $\mathbb{Z}$ celých čísel spolu s operací $\cdot$ (násobení) je grupoid. Analogicky, je grupoid i množina $\mathbb{Z}_4$ spolu s operací $\odot_4$ (násobení modulo 4). Zobrazení

$$[-]_4 : \mathbb{Z} \longrightarrow \mathbb{Z}_4$$

[4] To je technický termín: anglicky *a bag of sand*.

které každému celému číslu x přiřazuje $[x]_4$ (zbyek modulo 4), je homomorfismus grupoidů. To plyne z platnosti rovnosti

$$[x \cdot y]_4 = [x]_4 \odot_4 [y]_4$$

pro všechna $x, y \in \mathbb{Z}$. To je mimochodem důvod, proč má násobení modulo 4 podobné vlastnosti jako násobení celých čísel.

5.3.16 Věta *Identita je vždy homomorfismus grupoidů. Složením dvou homomorfismů grupoidů dostaneme opět homomorfismus grupoidů.*

Důkaz. Předvedeme dva důkazy: první „obvyklý“ a druhý s využitím komutujících diagramů. Druhá metoda má obrovské výhody, jak uvidíme později.

1. Zvolme libovolný grupoid $\langle X, \star_X \rangle$ a zobrazení $id_X : X \longrightarrow X$, tj. pro všechna $x \in X$ platí rovnost $id_X(x) = x$. Chceme ukázat, že pro všechna $x, y \in X$ platí

   $$id_X(x \star_X y) = id_X(x) \star_X \mathtt{i}d_X(y)$$

   To plyne okamžitě z definice identického zobrazení.

   Ať $f : \langle X, \star_X \rangle \longrightarrow \langle Y, \star_Y \rangle$ a $g : \langle Y, \star_Y \rangle \longrightarrow \langle Z, \star_Z \rangle$ jsou homomorfismy grupoidů. Chceme ukázat, že složené zobrazení $h = g \circ f$ je homomorfismus grupoidů, tj., že pro všechna $x, y \in X$ platí rovnost

   $$h(x \star_X y) = h(x) \star_Z h(y)$$

   To plyne okamžitě z definice skládání funkcí a z toho, že jak f tak g jsou homomorfismy grupoidů:

   $$\begin{aligned} h(x \star_X y) &= g\big(f(x \star_X y)\big) && \text{(skládání funkcí)} \\ &= g\big(f(x) \star_Y f(y)\big) && (f \text{ je homomorfismus grupoidů}) \\ &= g\big(f(x)\big) \star_Z g\big(f(y)\big) && (g \text{ je homomorfismus grupoidů}) \\ &= h(x) \star_Z h(y) && \text{(skládání funkcí)} \end{aligned}$$

2. Pro druhý důkaz využijeme definici homomorfismu grupoidů pomocí komutativního diagramu (5.9).

   Zvolme libovolný grupoid $\langle X, \star_X \rangle$ a zobrazení $id_X : X \longrightarrow X$. Potom platí $id_X \times id_X = id_{X \times X}$ a proto diagram

   $$\begin{array}{ccc} X \times X & \xrightarrow{id_X \times id_X} & X \times X \\ {\scriptstyle \star_X}\downarrow & & \downarrow{\scriptstyle \star_X} \\ X & \xrightarrow[id_X]{} & X \end{array}$$

   komutuje — zobrazení id_X je homomorfismus grupoidů.

   Ať $f : \langle X, \star_X \rangle \longrightarrow \langle Y, \star_Y \rangle$ a $g : \langle Y, \star_Y \rangle \longrightarrow \langle Z, \star_Z \rangle$ jsou homomorfismy grupoidů. Chceme ukázat, že složené zobrazení $h = g \circ f$ je homomorfismus grupoidů. To je jednoduché: stačí „slepit“ dva komutující diagramy:

   $$\begin{array}{ccccc} X \times X & \xrightarrow{f \times f} & Y \times Y & \xrightarrow{g \times g} & Z \times Z \\ {\scriptstyle \star_X}\downarrow & & \downarrow{\scriptstyle \star_Y} & & \downarrow{\scriptstyle \star_Z} \\ X & \xrightarrow[f]{} & Y & \xrightarrow[g]{} & Z \end{array}$$

   Diagram nalevo komutuje, protože f je homomorfismus grupoidů a diagram napravo komutuje, protože g je homomorfismus grupoidů.

■

5.3.17 Poznámka Všimněte si, že druhý způsob důkazu věty 5.3.16 byl opět příkladem abstraktního výpočtu, ovšem na úrovni homomorfismů grupoidů.

Studiem *morfismů* (ne pouze homomorfismů grupoidů) se zabývá matematická disciplína zvaná *teorie kategorií*. Kategorie je zadána svými *objekty*, *morfismy* a *pravidly skládání*. Tato data musí splňovat přirozené axiomy: skládání musí být asociativní a pro skládání musí existovat „neutrální morfismy".

Například kategorie Grpd všech grupoidů je zadána takto:

1. Objekty kategorie Grpd jsou všechny grupoidy.
2. Morfismy kategorie Grpd jsou homomorfismy grupoidů.
3. Pravidla skládání v Grpd jsou pravidla skládání funkcí. (Tj. morfismy v Grpd se skládají jako funkce.)

   Identický morfismus (tj. „neutrální" vzhledem ke skládání) grupoidu $\langle X, \star_X \rangle$ je identické zobrazení na množině X.

Teorie kategorií se tedy zabývá abstraktními výpočty velmi důsledně a jako taková se stala standardní součástí moderní computer science. Pro detaily odkazujeme například na knihu

☞ M. Barr a C. Wells, *Category Theory for Computing Science*, Prentice Hall, 1990

5.3.18 Poznámka Důkazová technika teorie kategorií (úvahy založené na komutování diagramů) se poněkud liší od té, na jakou jsme zatím byli zvyklí. Mohli bychom říci, že důkazy pomocí komutujících diagramů mají geometrickou povahu, zatímco „obvyklé" důkazy jsou „lineární" (píšeme zleva doprava a manipulujeme při tom se symboly).

Rozhodně nechceme říci, že úvahy založené na komutování diagramů jsou lepší než klasické úvahy. Geometrický vhled však může být často užitečný a může přinést překvapující důkaz. Porovnejte následující dva důkazy tvrzení

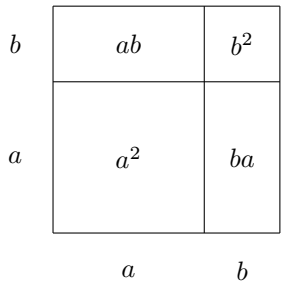
Pro všechna reálná čísla $a \geq 0$, $b \geq 0$ *platí* $(a+b)^2 = a^2 + 2ab + b^2$.

První důkaz je lineární, druhý geometrický:

1. Budeme postupovat následovně:

$$\begin{aligned} (a+b)^2 &= (a+b)\cdot(a+b) = \\ &= a\cdot(a+b) + b\cdot(a+b) = \\ &= a^2 + ab + ba + b^2 = \\ &= a^2 + 2ab + b^2 \end{aligned}$$

2. Podívejte se na následující obrázek:



Umíte namalovat obrázek, který „dokazuje" vzorec pro $(a+b)^3$?

5.3.19 Definice Homomorfismu grupoidů $f : \langle X, \star_X \rangle \longrightarrow \langle Y, \star_Y \rangle$ říkáme *isomorfismus*, pokud existuje homomorfismus $g : \langle Y, \star_Y \rangle \longrightarrow \langle X, \star_X \rangle$ tak, že platí

$$g \circ f = id_X \quad \text{a} \quad f \circ g = id_Y$$

Homomorfismu g říkáme *inverse* homomorfismu f.

Pokud existuje isomorfismus f z $\langle X, \star_X \rangle$ do $\langle Y, \star_Y \rangle$, říkáme, že $\langle X, \star_X \rangle$ a $\langle Y, \star_Y \rangle$ jsou *isomorfní*.

Dva isomorfní grupoidy se „z abstraktního hlediska neliší“. Vysvětlíme to na příkladu.

5.3.20 Příklad Množina $\{f_1, f_4\}$ je uzavřená na operaci $\circ$ v grupoidu z tabulky (5.8), tj. máme grupoid popsaný tabulkou

$\circ$	f_1	f_4
f_1	f_1	f_4
f_4	f_4	f_1

Jiným příkladem grupoidu je sčítání modulo 2, tj. množina $\{0,1\}$ vybavená operací $+$ podle tabulky

$+$	0	1
0	0	1
1	1	0

Obě tabulky jsou až na „přejmenování“ sloupců a řádků totožné, neboli grupoidy $\langle\{f_1, f_4\}, \circ\rangle$ a $\langle\mathbb{Z}_2, +\rangle$ jsou isomorfní: hledaný isomorfismus je zobrazení

$$\begin{array}{rcl} f : \{f_1, f_4\} & \longrightarrow & \mathbb{Z}_2 \\ f_1 & \mapsto & 0 \\ f_4 & \mapsto & 1 \end{array}$$

(Ukažte, že je to opravdu isomorfismus.)

Protože pologrupa je speciální případ grupoidu — vyžadujeme totiž, aby daná binární operace byla asociativní — předpokládáme, že homomorfismy pologrup by měly být speciálním případem homomorfismu grupoidů. Homomorfismus pologrup by měl respektovat dané binární operace a navíc by měl v nějakém smyslu respektovat asociativitu binárních operací. Respektování asociativního zákona je však již automatické:

5.3.21 Věta *Ať $\langle X, \star_X\rangle$ a $\langle Y, \star_Y\rangle$ jsou pologrupy. Předpokládejme, že zobrazení $f : X \longrightarrow Y$ je homomorfismus grupoidů, tj. pro všechna $x, y \in X$ platí rovnost:*

$$f(x \star_X y) = f(x) \star_Y f(y)$$

Potom z rovnosti

$$x \star_X (y \star_X z) = (x \star_X y) \star_X z$$

plyne rovnost

$$f(x) \star_Y \Big(f(y) \star_Y f(z)\Big) = \Big(f(x) \star_Y f(y)\Big) \star_Y f(z)$$

Důkaz. Zde není co dokazovat: protože asociativní zákon platí pro jakoukoli trojici prvků v množině Y, platí pochopitelně i pro trojici $f(x)$, $f(y)$, $f(z)$. ■

Homomorfismy pologrup tedy definujeme stejně jako homomorfismy grupoidů:

5.3.22 Definice Ať $\langle X, \star_X\rangle$ a $\langle Y, \star_Y\rangle$ jsou pologrupy. Zobrazení $f : X \longrightarrow Y$ je *homomorfismus pologrup z $\langle X, \star_X\rangle$ do $\langle Y, \star_Y\rangle$*, pokud následující diagram

$$\begin{array}{ccc} X \times X & \xrightarrow{f\times f} & Y \times Y \\ {\scriptstyle\star_X}\downarrow & & \downarrow{\scriptstyle\star_Y} \\ X & \xrightarrow[f]{} & Y \end{array}$$

je komutativní, tj. když pro každou dvojici $x, y \in X$ platí rovnost

$$f(x \star_X y) = f(x) \star_Y f(y)$$

Fakt, že zobrazení $f : X \longrightarrow Y$ je homomorfismus pologrup z $\langle X, \star_X\rangle$ do $\langle Y, \star_Y\rangle$ budeme značit

$$f : \langle X, \star_X\rangle \longrightarrow \langle Y, \star_Y\rangle$$

Postupme dále a věnujme se monoidům. Oproti pologrupě vyžadujeme v monoidu ještě existenci neutrálního prvku. Očekáváme tedy, že homomorfismus monoidů by měl

1. Respektovat binární operace, tj. být homomorfismem grupoidů.
2. Respektovat asociativní zákon. Víme, že to nastává automaticky.
3. Respektovat neutrální prvky. Plyne tento požadavek z respektování binárních operací? Uvidíme, že obecně nikoli.

5.3.23 Příklad Množina $\mathbb{Z}_2$ spolu s operací násobení je monoid (neutrální prvek je 1). Jednoprvková množina $\{0\}$ spolu s operací násobení je také monoid — neutrální prvek je 0. Zobrazení

$$\begin{array}{rcl} f : \{0\} & \longrightarrow & \mathbb{Z}_2 \\ 0 & \mapsto & 0 \end{array}$$

respektuje binární operace, protože platí $f(0 \cdot 0) = f(0) \cdot f(0)$. Toto zobrazení f ovšem nerespektuje neutrální prvky, protože $f(0) \neq 1$.

5.3.24 Definice Ať $\langle X, \star_X, e_X\rangle$ a $\langle Y, \star_Y, e_Y\rangle$ jsou monoidy. Homomorfismus pologrup je *homomorfismus monoidů z* $\langle X, \star_X\rangle$ *do* $\langle Y, \star_Y\rangle$, pokud platí rovnost

$$f(e_X) = e_Y$$

Fakt, že zobrazení $f : X \longrightarrow Y$ je homomorfismus monoidů z $\langle X, \star_X\rangle$ do $\langle Y, \star_Y\rangle$ budeme značit

$$f : \langle X, \star_X\rangle \longrightarrow \langle Y, \star_Y\rangle$$

Dalším krokem je definice homomorfismu grup. Zde navíc k respektování neutrálních prvků přibude respektování inversí:

5.3.25 Definice Ať $\langle X, \star_X\rangle$ a $\langle Y, \star_Y\rangle$ jsou grupy. Homomorfismus monoidů je *homomorfismus grup z* $\langle X, \star_X\rangle$ *do* $\langle Y, \star_Y\rangle$, pokud pro každé $x \in X$ platí rovnost

$$f(x^{-1}) = \big(f(x)\big)^{-1}$$

Fakt, že zobrazení $f : X \longrightarrow Y$ je homomorfismus grup z $\langle X, \star_X\rangle$ do $\langle Y, \star_Y\rangle$ budeme značit

$$f : \langle X, \star_X\rangle \longrightarrow \langle Y, \star_Y\rangle$$

Následující příklad má velký význam v teoretické fyzice. Ukazuje, že každou *konečnou* grupu si lze představit jako množinu pohybů v konečně dimensionálním vektorovém prostoru.[5]

5.3.26 Příklad Připomeňme nejprve, že množina $\mathbb{R}^n$, n je pevné kladné přirozené číslo, je vektorový prostor nad $\mathbb{R}$ dimense n. Každou matici M rozměrů $n \times n$ je pak možné si přestavit jako *lineární* zobrazení:

$$M : \mathbb{R}^n \longrightarrow \mathbb{R}^n, \quad x \mapsto M \cdot x$$

neboli jakožto *pohyb* v $\mathbb{R}^n$. Pohybem tu rozumíme transformaci kanonické báze, tj. $e_1 \mapsto M \cdot e_1, \ldots, e_n \mapsto M \cdot e_n$. Speciální pohyby jsou ty, kdy je matice M *invertibilní* (tj. *regulární*), takové pohyby jsou *reversibilní*.

Protože násobení matic odpovídá skládání lineárních zobrazení, je možné si grupu $\mathsf{RegMat}_{n\times n}(\mathbb{R})$ všech regulárních matic (viz příklad 5.3.10) představit jako *grupu reversibilních lineárních transformací v* $\mathbb{R}^n$.

Ukážeme nyní, že každou grupu $\langle X, \star, e\rangle$ o n prvcích je možné v $\mathsf{RegMat}_{n\times n}(\mathbb{R})$ *representovat*, tj. ukážeme existenci *prostého homomorfismu grup*

$$r : X \longrightarrow \mathsf{RegMat}_{n\times n}(\mathbb{R})$$

tj. zobrazení s následujícími vlastnostmi:

[5] Anglicky se tento výsledek jmenuje *a faithful representation of finite groups.*

1. Zobrazení r je *prosté*, tj. jestliže $x \neq x'$, pak $r(x) \neq r(x')$. Intuice je následující: dva různé prvky grupy představují dva různé pohyby v $\mathbb{R}^n$.

2. Zobrazení r *respektuje binární operace*, tj. platí rovnost $r(x \star x') = r(x) \cdot r(x')$. To intuitivně znamená, že součin v grupě X může být chápán jako skládání příslušných pohybů.

3. Zobrazení r *respektuje neutrální prvky*, tj. platí rovnost $r(e) = \mathbb{E}$. Intuice: neutrální prvek je representován jako neutrální pohyb (zůstaneme na místě).

4. Zobrazení r *respektuje inverse*, tj. platí, že matice $r(x^{-1})$ je inversní k matici $r(x)$. Intuice: pohyb přiřazený prvku x^{-1} je inversní k pohybu, který jsme přiřadili prvku x.

Protože grupa $\langle X, \star, e\rangle$ má n prvků, můžeme je oindexovat čísly od 1 do n, tj.

$$X = \{x_1, \ldots, x_n\}$$

Do i-tého řádku a j-tého sloupce matice $r(x)$ zapíšeme jedničku právě tehdy, když platí:

$$x \star x_i = x_j$$

Jindy zapíšeme do matice $r(x)$ samé nuly.

Například pro grupu $\mathbb{Z}_{12}^*$ invertibilních prvků v $\mathbb{Z}_{12}$ (viz příklad 5.3.10) máme $\mathbb{Z}_{12}^* = \{1, 5, 7, 11\}$. Její prvky oindexujeme

$$x_1 = 1, \quad x_2 = 5, \quad x_3 = 7, \quad x_4 = 11$$

a příslušné matice jsou tyto (spočítejte je):

$$r(x_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad r(x_2) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad r(x_3) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad r(x_4) = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Povšimněte si, že každá z výše uvedených matic má v řádcích zapsánu permutaci kanonické báze e_1, e_2, e_3, e_4. To platí naprosto obecně (dokažte to) a proto má obecně každá matice $r(x)$ determinant buď $+1$ nebo -1. Speciálně je každá matice $r(x)$ regulární, máme tedy zobrazení

$$r : X \longrightarrow \mathsf{RegMat}_{n\times n}(\mathbb{R})$$

Ukážeme nyní požadované vlastnosti zobrazení r:

1. Když platí $x \neq x'$, pak platí $x \star x_i \neq x' \star x_i$ pro každé i.[6] Proto platí $r(x) \neq r(x')$.

2. Budeme-li matice $r(x)$ a $r(x')$ násobit, objeví se v součinu v i-tém řádku a j-tém sloupci jednička přesně tehdy, když existuje k takové, že platí $x \star x_i = x_k$ a současně $x' \star x_k = x_j$. To ale přesně znamená, že platí $(x \star x') \star x_i = x_j$, neboli jedničku v i-tém řádku a j-tém sloupci matice $r(x \star x')$.

3. Rovnost $r(e) = \mathbb{E}$ je zřejmá: matice $r(e)$ má jedničky pouze na diagonále.

4. Potřebujeme dokázat rovnost $r(x^{-1}) = r(x)^{-1}$. Povšimněme si, že matice $r(x^{-1})$ je transponovaná k matici $r(x)$, protože rovnost $x \star x_i = x_j$ platí právě tehdy, když platí rovnost $x^{-1} \star x_j = x_i$. Připomeňme, že každá matice $r(x)$ obsahuje v řádcích nějakou permutaci Π kanonické báze $e_1, \ldots, e_n$. Sloupce matice $r(x^{-1})$ jsou tudíž tvořeny *stejnou* permutací Π kanonické báze $e_1, \ldots, e_n$. Při počítání součinů $r(x)\cdot r(x^{-1})$ a $r(x^{-1}) \cdot r(x)$ tudíž do položky (i, j) vždy ukládáme skalární součin $\langle e_{\Pi(i)} \mid e_{\Pi(j)}\rangle$. Protože báze $e_1, \ldots, e_n$ je ortonormální, je takový skalární součin vždy buď 0 (když $\Pi(i) \neq \Pi(j)$) nebo 1 (když $\Pi(i) = \Pi(j)$). Protože Π je permutace, platí $\Pi(i) \neq \Pi(j)$ právě tehdy, když $i \neq j$, a $\Pi(i) = \Pi(j)$ platí právě tehdy, když $i = j$. Tudíž platí rovnosti $r(x) \cdot r(x^{-1}) = \mathbb{E} = r(x^{-1}) \cdot r(x)$, a to jsme chtěli dokázat.

[6] Tuto vlastnost jsme používali při důkazu Fermatovy věty. Znamená to, že můžete Fermatovu větu zobecnit pro konečné grupy? Pokuste se o to! Viz cvičení 5.7.12.

5.4 Více operací na jedné množině

5.4.1 Definice *Finitární typ* Ω je spočetná posloupnost navzájem disjunktních množin $\Omega_0, \Omega_1, \ldots$. Prvkům množiny Ω_n říkáme *formální n-ární operace*.

Formální operace budou v jednotlivých algebrách *interpretována* jako jistá zobrazení. Například formální binární operace (tj. operace arity 2) bude interpretována jako binární operace na množině. Každá binární operace na množině X je zobrazení z $X \times X$ do X. Mohli bychom říci, že binární operace je definována na *druhé mocnině* množiny X. Obecně pro $n \geq 0$ definujeme

$$\begin{aligned} X^0 &= 1 \quad \text{kde } 1 \text{ je pevně zvolená jednoprvková množina} \\ X^{n+1} &= X \times X^n \end{aligned}$$

Množina X^0 má tedy jediný prvek, prvky množiny X^n, pro $n > 0$, jsou uspořádané n-tice prvků množiny X.

Toto exponenciální značení nyní rozšíříme i na značení pro zobrazení: pro zobrazení $f : X \longrightarrow Y$ je

$$f^0 : X^0 \longrightarrow Y^0$$

identické zobrazení na množině 1 a pro $n > 0$ je

$$f^n : X^n \longrightarrow Y^n$$

zobrazení, které uspořádanou n-tici $(x_1, \ldots, x_n) \in X^n$ pošle na uspořádanou n-tici $\big(f(x_1), \ldots, f(x_n)\big) \in Y^n$.

5.4.2 Definice Ať Ω je finitární typ. *Algebra* $\mathbb{X}$ *typu* Ω je množina X spolu s kolekcí zobrazení

$$\omega_{\mathbb{X}} : X^n \longrightarrow X \quad \text{pro všechna } n \geq 0 \text{ a všechna } \omega \in \Omega_n$$

Množině X říkáme *nosná množina algebry* $\mathbb{X}$ a zobrazení $\omega_{\mathbb{X}}$ nazveme *interpretace formální operace* ω *v algebře* $\mathbb{X}$.

Jsou-li $\mathbb{X}$ a $\mathbb{Y}$ algebry typu Ω s nosnými množinami X a Y, potom zobrazení

$$f : X \longrightarrow Y$$

nazveme *homomorfismus algeber typu* Ω, pokud následující diagram

$$\begin{array}{ccc} X^n & \xrightarrow{f^n} & Y^n \\ {\scriptstyle \omega_{\mathbb{X}}}\downarrow & & \downarrow{\scriptstyle \omega_{\mathbb{Y}}} \\ X & \xrightarrow[f]{} & Y \end{array}$$

je komutativní pro všechna $n \geq 0$ a všechna $\omega \in \Omega_n$.

V případě $n = 0$ to znamená, že pro všechna $\omega \in \Omega_0$ platí rovnost

$$f(\omega_{\mathbb{X}}) = \omega_{\mathbb{Y}} \quad (f \text{ respektuje všechny konstanty})$$

a v případě $n > 0$ to znamená, že pro všechna $\omega \in \Omega_n$ a všechna $x_1, \ldots, x_n$ platí rovnost

$$f\big(\omega_{\mathbb{X}}(x_1, \ldots, x_n)\big) = \omega_{\mathbb{Y}}\big(f(x_1), \ldots, f(x_n)\big) \quad (f \text{ respektuje všechny } n\text{-ární operace})$$

Nyní zavedeme pojem isomorfismu pro obecné algebry. Zhruba řečeno, dvě algebry jsou isomorfní, když se liší pouze jmény svých prvků. Definici napíšeme kategoriálně, porovnejte ji s definicí 5.3.19.

5.4.3 Definice Ať $\mathbb{X}$ a $\mathbb{Y}$ jsou algebry typu Ω. Řekneme, že homomorfismus $f : \mathbb{X} \longrightarrow \mathbb{Y}$ je *isomorfismus*, když existuje homomorfismus $g : \mathbb{Y} \longrightarrow \mathbb{X}$ takový, že diagram

$$\begin{array}{ccccc} \mathbb{X} & \xrightarrow{f} & \mathbb{Y} & & \\ & \searrow & \downarrow{\scriptstyle g} & \swarrow & \\ & & \mathbb{X} & \xrightarrow[f]{} & \mathbb{Y} \end{array}$$

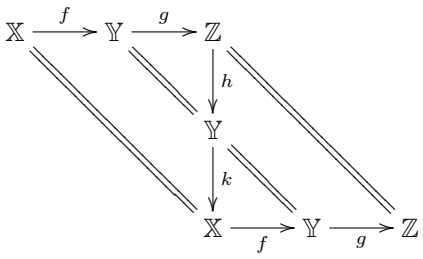
komutuje.

Algebry $\mathbb{X}$, $\mathbb{Y}$ nazveme *isomorfní*, pokud existuje isomorfismus $f : \mathbb{X} \longrightarrow \mathbb{Y}$.

Jako další příklad aplikace kategoriální techniky dokážeme následující jednoduché tvrzení:

5.4.4 Lemma *Složení dvou isomorfismů je isomorfismus.*

Důkaz. Ať $f : \mathbb{X} \longrightarrow \mathbb{Y}$ a $g : \mathbb{Y} \longrightarrow \mathbb{Z}$ jsou isomorfismy. Podívejte se na následující komutativní diagram:



■

5.4.5 Příklad (Grupoidy)
Grupoidy jsou algebry typu Ω, který specifikuje pouze jednu formální binární operaci $\star$ (ta bude v grupoidu interpretována jako „násobení“). Typ Ω je tedy

$$\Omega_2 = \{\star\} \quad \Omega_n = \emptyset \text{ pro } n \neq 2$$

Algebry typu Ω jsou přesně grupoidy a homomorfismy algeber typu Ω jsou přesně homomorfismy grupoidů.

Povšimněme si, že pologrupy jsou algebry stejného typu jako grupoidy. V pologrupě však navíc vyžadujeme platnost *rovnosti* pro asociativní zákon.

5.4.6 Příklad (Grupy)
Grupy jsou algebry typu Ω, který specifikuje jednu formální binární operaci $\star$ (ta bude v grupě interpretována jako „násobení“), jednu formální 0-ární operaci e (ta bude v grupě interpretována jako neutrální prvek) a jednu formální unární operaci $(-)^{-1}$ (ta bude v grupě interpretována jako operace inverse). Typ Ω je tedy

$$\Omega_0 = \{e\} \quad \Omega_1 = \{(-)^{-1}\} \quad \Omega_2 = \{\star\} \quad \Omega_n = \emptyset \text{ pro } n > 2$$

Algebra typu Ω však ještě nemusí být grupa. Chceme, aby všechny specifikované operace splňovaly všechny *rovnosti* z definice grupy.

5.4.7 Příklad (Svazy)
Dalším příkladem finitárního typu je posloupnost množin Ω_0, Ω_1,..., kde $\Omega_2 = \{\wedge, \vee\}$ a $\Omega_n = \emptyset$ pro všechna $n \neq 2$. Tento finitární typ tedy specifikuje pouze dvě formální binární operace: $\wedge$ (budeme jí říkat *průsek*) a $\vee$ (této operaci budeme říkat *spojení*).

Algebra $\mathbb{X}$ typu Ω je tedy množina X vybavená dvojicí binárních operací

$$\wedge_{\mathbb{X}} : X \times X \longrightarrow X \quad \text{a} \quad \vee_{\mathbb{X}} : X \times X \longrightarrow X$$

V dalším budeme index $\mathbb{X}$ u $\wedge_{\mathbb{X}}$ a $\vee_{\mathbb{X}}$ vynechávat. Algebra typu Ω je tedy trojice $\langle X, \wedge, \vee \rangle$.

Algebře $\langle X, \wedge, \vee \rangle$ typu Ω budeme říkat *svaz*, pokud tyto operace splňují pro všechna $x, y, z \in X$ následující sadu rovností:

1. *Komutativita* $\wedge$: $x \wedge y = y \wedge x$.
2. *Asociativita* $\wedge$: $x \wedge (y \wedge z) = (x \wedge y) \wedge z$.
3. *Idempotence* $\wedge$: $x \wedge x = x$.
4. *Absorpce* $\wedge$: $x \vee (x \wedge y) = x$.
5. *Komutativita* $\vee$: $x \vee y = y \vee x$.
6. *Asociativita* $\vee$: $x \vee (y \vee z) = (x \vee y) \vee z$.
7. *Idempotence* $\vee$: $x \vee x = x$.
8. *Absorpce* $\vee$: $x \wedge (x \vee y) = x$.

Více si o svazech povíme v odstavci 6.1.

5.4.8 Poznámka V dalším odstavci budeme obecně studovat *rovnice*. Již nyní bychom však měli mít vyvinutý cit pro definici homomorfismu: homomorfismus musí respektovat všechny specifikované operace, respektování rovností pak bude automatické. (Srovnejte s větou 5.3.21.)

5.5 Volné algebry a rovnice

Jsme již nyní blízko k tomu, abychom byli schopni říci, co je (jednosortová) algebraická specifikace. Pojem finitárního typu zachycuje myšlenku specifikace operací. Nyní vysvětlíme, jak popsat *specifikaci rovnic*. Začneme příkladem specifikace rovnice pro asociativní zákon.

5.5.1 Příklad V tomto příkladu popíšeme alternativní způsob vyjádření rovnosti asociativity. Budeme postupovat podobně jako při definici typu. Typ je specifikací nejrůznějších *formálních* operací, které jsou pak *interpretovány* jako konkrétní zobrazení. Chceme-li specifikovat „formální rovnici“, můžeme nejprve specifikovat formální pravou a levou stranu rovnice, potom je interpretovat a vyžadovat rovnost.

Ať Ω je typ grupoidů. Tento typ obsahuje jedinou formální binární operaci $\star$ — viz 5.4.5. Zvolme pevnou algebru typu Ω. Pro větší názornost si představme, že volíme například grupoid $\mathbb{Z}_5$, kde formální operace $\star$ je interpretována jako sčítání modulo 5.

Asociativní zákon pro sčítání modulo 5 znamená, že pro všechny prvky $x, y, z \in \mathbb{Z}_5$ platí rovnost

$$x + (y + z) = (x + y) + z$$

Prvky x, y a z tu hrají roli proměnných, které probíhají celou množinu $\mathbb{Z}_5$. Uspořádané trojice prvků $\mathbb{Z}_5$ ale přesně odpovídají zobrazením z pevné tříprvkové množiny V do $\mathbb{Z}_5$. Označme prvky této pevné množiny V jako v_1, v_2, v_3.

1. Jestliže x, y a z jsou prvky v $\mathbb{Z}_5$, potom lze sestrojit toto zobrazení

$$\begin{array}{rcl} \rho : V & \longrightarrow & \mathbb{Z}_5 \\ v_1 & \mapsto & x \\ v_2 & \mapsto & y \\ v_3 & \mapsto & z \end{array}$$

2. Obráceně, zobrazení $\rho : V \longrightarrow \mathbb{Z}_5$ určuje trojici prvků $\mathbb{Z}_5$: totiž $\rho(v_1)$, $\rho(v_2)$ a $\rho(v_3)$.

V dalším budeme prvkům množiny V říkat *formální proměnné* a funkci $\rho : V \longrightarrow \mathbb{Z}_5$ *interpretace formálních proměnných v* $\mathbb{Z}_5$.

Formální levá strana asociativního zákona — sice $x + (y + z)$ — potom je $v_1 \star (v_2 \star v_3)$. Všimněme si, že jsme zásadně používali formální proměnné i formální operace. Podobně, formální pravá strana rovnice — sice $(x + y) + z$ — je $(v_1 \star v_2) \star v_3$.

Jak budeme interpretovat například formální levou stranu? To zařídí interpretace formálních proměnných. Vezměme například tuto interpretaci:

$$\begin{array}{rcl} \rho : V & \longrightarrow & \mathbb{Z}_5 \\ v_1 & \mapsto & 0 \\ v_2 & \mapsto & 2 \\ v_3 & \mapsto & 3 \end{array}$$

Potom $v_1 \star (v_2 \star v_3)$ je interpretováno jako $0 + (2 + 3)$ a $(v_1 \star v_2) \star v_3$ je interpretováno jako $(0 + 2) + 3$. Interpretovali jsme tedy jak formální proměnné, tak formální operace. Protože interpretace levé i pravé strany jsou si rovny, ukázali jsme, že v $\mathbb{Z}_5$ platí

$$0 + (2 + 3) = (0 + 2) + 3$$

Chceme-li nyní vyjádřit, že podobná rovnost platí pro všechny možné trojice prvků v $\mathbb{Z}_5$, stačí požadovat, aby se levá strana rovnala pravé straně při *každé interpretaci formálních proměnných*, viz definice 5.5.4.

Pojem rovnice teď definujeme obecně.

5.5.2 Definice Ať V je pevná množina a Ω je finitární typ. Množině V budeme v tomto kontextu říkat *množina formálních proměnných*.

Je-li $\mathbb{X}$ algebra typu Ω s nosnou množinou X, potom *interpretace formálních proměnných v algebře* $\mathbb{X}$ je jakékoli zobrazení

$$\rho : V \longrightarrow X$$

Množina $T_\Omega(V)$ *termů typu* Ω *nad množinou* V je definována induktivně[7] podle následujících pravidel:

1. Je-li $v \in V$, potom $v \in T_\Omega(V)$.

   Jinými slovy: každá formální proměnná je term.

2. Je-li $\omega \in \Omega_0$, potom $\omega \in T_\Omega(V)$.

   Neboli: každá formální konstanta je term.

3. Je-li $n > 0$, je-li $\omega \in \Omega_n$ a jsou-li $t_1, \ldots, t_n$ prvky $T_\Omega(V)$, potom $\omega(t_1, \ldots, t_n)$ je prvek množiny $T_\Omega(V)$.

   To znamená: „aplikací“ formální n-ární operace na n-tici termů vzniká opět term.

Formální rovnice v proměnných V je zápis

$$\forall V.\, t_1 \approx t_2$$

kde t_1 a t_2 jsou prvky $T_\Omega(V)$.

Ať $\mathbb{X}$ je nějaká algebra typu Ω s nosnou množinou X. Uvědomme si, že induktivní zadání množiny $T_\Omega(V)$ dovoluje definovat *rozšíření* zobrazení $\rho : V \longrightarrow X$ na zobrazení $\rho^\sharp : T_\Omega(V) \longrightarrow X$. Pochopitelně, toto rozšíření bude definováno rekursivně.

Definujeme:

1. $\rho^\sharp(v) = \rho(v)$, pro všechna $v \in V$.

2. $\rho^\sharp(\omega) = \omega_{\mathbb{X}}$, pro všechna $\omega \in \Omega_0$.

3. Je-li $n > 0$, je-li $\omega \in \Omega_n$ a jsou-li $t_1, \ldots, t_n$ prvky $T_\Omega(V)$, potom

$$\rho^\sharp\big(\omega(t_1, \ldots, t_n)\big) = \omega_{\mathbb{X}}\big(\rho^\sharp(t_1), \ldots, \rho^\sharp(t_n)\big)$$

Nyní jsme na stopě universální vlastnosti, kterou splňuje množina termů. Nejprve si všimněme, že $T_\Omega(V)$ je (velmi přirozeným způsobem) nosná množina algebry $\mathbb{T}_\Omega(V)$ typu Ω. Ukážeme, jak jsou formální operace typu Ω interpretovány v množině $T_\Omega(V)$:

1. Každá formální konstanta $\omega \in \Omega_0$ je interpretována jako term $\omega \in T_\Omega(V)$.

2. Každá formální operace $\omega \in \Omega_n$, $n > 0$, je interpretována jako zobrazení

$$\omega_{T_\Omega(V)} : T_\Omega(V)^n \longrightarrow T_\Omega(V)$$

   které n-tici termů $t_1, \ldots, t_n$ v $T_\Omega(V)$ zobrazí na term $\omega(t_1, \ldots, t_n)$.

Výše zmíněné rozšíření $\rho^\sharp : T_\Omega(V) \longrightarrow X$ je pak homomorfismus algeber typu Ω z $\mathbb{T}_\Omega(V)$ do $\mathbb{X}$. Opravdu: zobrazení $\rho^\sharp$ respektuje:

1. Každou konstantu, protože pro všechna $\omega \in \Omega_0$ platí rovnost $\rho^\sharp(\omega) = \omega_{\mathbb{X}}$.

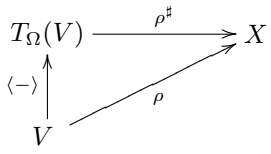
2. Každou operaci arity $n > 0$, protože pro všechny $\omega \in \Omega_n$ platí rovnost

$$\rho^\sharp\big(\omega_{\mathbb{T}_\Omega(V)}(t_1, \ldots, t_n)\big) = \omega_{\mathbb{X}}\big(\rho^\sharp(t_1), \ldots, \rho^\sharp(t_n)\big)$$

Toto rozšíření $\rho^\sharp$ je navíc jediné možné — zobrazení $\rho : V \longrightarrow X$ nelze na homomorfismus rozšířit jinak než jako zobrazení $\rho^\sharp : \mathbb{T}_\Omega(V) \longrightarrow \mathbb{X}$.

[7] Induktivně zadané množiny jsou definovány obecně v odstavci 1.4.

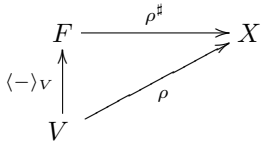
Neviděli jsme to již někde? Nakresleme komutativní diagram:



kde jsme jako $\langle - \rangle : V \longrightarrow T_\Omega(V)$ označili zobrazení, které každou proměnnou chápe jako term.

Ano, to je přesně to, co jsme viděli při úvahách o modelu Σ^* specifikace `LIST` popsané v (5.1).

5.5.3 Definice Ať Ω je finitární typ a ať $\mathbb{F}$ je algebra typu Ω s nosnou množinou F. Ať V je jakákoli množina („proměnných“). Řekneme, že zobrazení $\langle - \rangle_V : V \longrightarrow F$ *předkládá* $\mathbb{F}$ *jako volnou algebru typu* Ω *nad* V, když platí:

> Pro jakoukoli algebru $\mathbb{X}$ typu Ω s nosnou množinou X a pro jakékoli zobrazení $\rho : V \longrightarrow X$ („interpretace proměnných“) existuje právě jedno zobrazení $\rho^\sharp : F \longrightarrow X$, které je homomorfismus algeber typu Ω a pro které diagram
>
> 
>
>
> komutuje.

Zobrazení $\langle - \rangle : V \longrightarrow T_\Omega(V)$ tedy předkládá algebru termů $\mathbb{T}_V(\Omega)$ jako volnou algebru typu Ω nad V. Tuto vlastnost nyní využijeme k definici splňování rovnice.

5.5.4 Definice Řekneme, že algebra $\mathbb{X}$ typu Ω s nosnou množinou X *splňuje formální rovnici* $\forall V.\, t_1 \approx t_2$, když platí rovnost $\rho^\sharp(t_1) = \rho^\sharp(t_2)$ pro všechna $\rho : V \longrightarrow X$.

Tento fakt značíme $\mathbb{X} \models \forall V.\, t_1 \approx t_2$.

Víme, že algebra termů $\mathbb{T}_\Omega(V)$ je volná mezi *všemi* algebrami typu Ω. Nyní naznačíme, jak sestrojit algebru *volnou mezi algebrami, které splňují jednu formální rovnici*. Takovou algebru zkonstruujeme z algebry termů tak, že si platnost oné formální rovnice *vynutíme*: v algebře termů prostě „slepíme“ dohromady pravou a levou stranu dané formální rovnice. Nicméně jak uvidíme, slepení pravé a levé strany může mít za následek slepení celé řady dalších věcí. Chceme totiž, aby výsledná množina byla opět algebra.

Začněme s formální rovnicí $\forall V.\, t_1 \approx t_2$. Co znamená slepit termy t_1 a t_2? Určitě budeme požadovat *skutečnou rovnost* obou termů. Nicméně když tyto sobě rovné termy dosadíme jako argumenty nějaké (například) unární operace ω, musí si výsledné termy $\omega(t_1)$ a $\omega(t_2)$ být opět rovny. Podobně pro (například) binární operaci τ bychom měli slepit například termy $\tau(t_1, t_1)$ a $\tau(t_1, t_2)$. A tak dále. Výsledkem takové rekursivní definice je pak binární relace na množině termů $T_\Omega(V)$, která je relace ekvivalence a která respektuje všechny operace. Takovým relacím (na obecných algebrách) říkáme *kongruence* a nyní je definujeme obecně.

5.5.5 Definice Ať $\mathbb{X}$ algebra typu Ω. nad množinou proměnných V. *Kongruence* R na $\mathbb{X}$ je relace ekvivalence na nosné množině X algebry $\mathbb{X}$, která respektuje strukturu $\mathbb{X}$, tj. pro kterou platí:

> jestliže $\omega \in \Omega_n$ je formální operace a jestliže $x_1, \ldots, x_n$, $x'_1, \ldots, x'_n$ jsou prvky X, pro které platí $x_1\ R\ x'_1$, $\ldots$, $x_n\ R\ x'_n$, pak platí $\omega_{\mathbb{X}}(x_1, \ldots, x_n)\ R\ \omega_{\mathbb{X}}(x'_1, \ldots, x'_n)$.

5.5.6 Poznámka Kongruence je návodem, jak prvky algebry „slepit“ tak, aby výsledkem byla opět algebra. Slepování prvků množiny pomocí relace ekvivalence bylo popsáno v poznámce 2.2.5. U algeber ovšem čekáme, že ne každé slepování je „dobré“. Opět pomůže pohled pytlů písku, viz poznámku 5.3.13.

Pokud si představíme naši algebru jako pytel písku, kde se zrnka slepila do hrudek tvaru krychle (stejné velikosti), pak rozhodně například slepením dvou hrudek do jedné nevznikne hrudka tvaru krychle. Aby vznikla algebra, jsme donuceni slepovat hrudky do krychlí po osmicích.

„Dobrost“ kongruencí ukazuje následující věta: nosnou množinu můžeme podle kongruence „slepit“ a na příslušné faktorové množině pak přirozeným způsobem vzniká struktura algebry.

5.5.7 Věta *Ať R je kongruence na algebře $\mathbb{X}$ typu Ω. Potom lze na množině X/R vytvořit* faktorovou algebru *$\mathbb{X}/R$, když pro každou formální operaci $\omega \in \Omega_n$ definujeme*

$$\omega_{\mathbb{X}/R}([x_1]_R, \ldots, [x_n]_R) = [\omega_{\mathbb{X}}(x_1, \ldots, x_n)]_R$$

kde jsme, pro $x \in X$, označili jako $[x]_R$ třídu kongruence určenou representantem x, tj. $[x]_R = \{x' \in X \mid x \; R \; x'\}$.[8]

Důkaz. Protože platí $X/R = \{[x]_R \mid x \in X\}$, stačí ukázat, že jsme skutečně, pro každé $\omega \in \Omega_n$, definovali zobrazení

$$\omega_{\mathbb{X}/R} : (X/R)^n \longrightarrow X/R$$

To ale plyne okamžitě z definice kongruence: jestliže platí

$$[x_1]_R = [x'_1]_R, \ldots, [x_n]_R = [x'_n]_R$$

pak platí

$$[\omega_{\mathbb{X}}(x_1, \ldots, x_n)]_R = [\omega_{\mathbb{X}}(x_1, \ldots, x_n)]_R$$

Definice operací na faktorové množině tedy nezávisí na výběru representantů, a to jsme potřebovali ukázat. ■

Předchozí věta vypadá velmi abstraktně. Faktorové algebry jsme však už viděli a umíme s nimi pracovat:

5.5.8 Příklad Připomeňme, že $\langle \mathbb{Z}, +, \cdot, 0, 1 \rangle$ je komutativní okruh s jednotkou (viz příklady 2.3.16) a relace kongruence modulo m je kongruence na tomto okruhu ve smyslu definice 5.5.5 (to tvrdí tvrzení 2.3.3). Podle předchozí věty proto můžeme utvořit faktorovou algebru. Tato algebra není nic jiného než komutativní okruh s jednotkou značený $\mathbb{Z}_m$.

Speciálně, pro každou kongruenci R na algebře $\mathbb{T}_\Omega(V)$ termů nad V máme k dispozici faktorovou algebru $\mathbb{T}_\Omega(V)/R$. Navíc platí: pokud je term t kongruentní s termem t', pak v algebře $\mathbb{T}_\Omega(V)/R$ platí $[t]_R = [t']_R$. To je přesně to, co potřebujeme k vynucování formálních rovnic.

Vezměme nyní formální rovnici $\forall V.\, t_1 \approx t_2$ a nejprve vytvořme nejmenší kongruenci R, která obsahuje dvojici (t_1, t_2) a potom vytvořme algebru $\mathbb{T}_\Omega(V)/R$ spolu se zobrazením $[-]_R : V \longrightarrow \mathbb{T}_\Omega(V)/R$ (které každé formální proměné v přiřadí třídu ekvivalence $[v]_R$). Potom zobrazení $[-]_R : V \longrightarrow \mathbb{T}_\Omega(V)/R$ předkládá $\mathbb{T}_\Omega(V)/R$ jako volnou algebru mezi všemi algebrami splňujícími formální rovnici $\forall V.\, t_1 \approx t_2$.

5.5.9 Poznámka Vzniká otázka, jestli můžeme vždy najít nejmenší kongruenci, která slepuje termy t_1 a t_2. Odpověď je kladná a dokonce je snadné tuto kongruenci induktivně zadat (srovnejte s příkladem 2.3.6). Sada pravidel[9]

$$\frac{}{t_1 \; R \; t_2}\,(e) \quad | \quad \frac{}{t \; R \; t}\,(r) \quad | \quad \frac{t \; R \; t'}{t' \; R \; t}\,(s) \quad | \quad \frac{t \; R \; t' \quad t' \; R \; t''}{t \; R \; t''}\,(t)$$

induktivně zadává nejmenší ekvivalenci (pravidla (r), (s), (t) zaručují po řadě reflexivitu, symetrii a transitivitu), která slepuje t_1 a t_2 (to zaručuje pravidlo (e)). Dále, pro každé $\omega \in \Omega_n$, přidáme pravidlo

$$\frac{t_1 \; R \; t'_1, \ldots, t_n \; R \; t'_n}{\omega(t_1, \ldots, t_n) \; R \; \omega(t'_1, \ldots, t'_n)}\,(\omega)$$

které z R vytváří kongruenci.

Začínáme-li s *množinou* formálních rovnic E, postupujeme podobně jako v případu rovnice jediné. Sestrojíme nejmenší kongruenci R, která obsahuje všechny dvojice termů, které si mají být rovny. Algebra $\mathbb{T}_\Omega(V)/R$ je pak volná nad množinou V mezi všemi algebrami, které splňují všechny formální rovnice z množiny E (viz cvičení 5.7.19).

[8] Tj. $[x]_R$ je množina bodů x' „slepených“ s bodem x. Srovnejte se značením tříd modulo $[c]_m$ z definice 2.3.4 a s poznámkou 2.2.5.

[9] Namítnete, že v odstavci 1.4 jsou induktivní definice zavedeny pouze pro množiny slov nad konečnou abecedou. Jednoduchou modifikací úvah z odstavce 1.4 však můžeme notaci odvozovacích pravidel použít pro náš příklad. Pokuste se o to.

5.5.10 Poznámka Přístup k volným algebrám, který jsme zvolili, je přístupem klasické universální algebry. Daleko obecnější (a více koncepční) přístup k volným objektům dává teorie kategorií. Jde o pojem *adjungovaných funktorů*.[10] Odkazujeme na knihu

☞ M. Barr a C. Wells, *Category Theory for Computing Science*, Prentice Hall, 1990

5.6 Aplikace — iniciální sémantika algebraických specifikací

V odstavci 5.1 jsme uvedli příklady algebraických specifikací. Viděli jsme, že pro danou specifikaci obecně existuje celá řada modelů, některé z nich dost bizarní. Teorie vybudovaná v této kapitole nám nyní umožní přesně definovat (rovnicovou) algebraickou specifikaci a říci, který její model je ten „nejlepší“. Protože zatím nemáme k dispozici obecnou vícesortovou teorii, omezíme se na jednosortový případ. Více sortami se budeme zabývat v odstavci 6.4.

V jednosortovém případě nemusíme sorty specifikovat — studujeme jedinou sortu. Specifikace by se měla sestávat z konečného seznamu operací a konečného seznamu rovnic. K tomu se přesně hodí pojmy finitárního typu a formálních rovnic, které jsme zavedli v předchozích odstavcích. Pojem finitárního typu (definice 5.4.1) je však příliš obecný: dovoluje nám specifikovat *nekonečně mnoho* finitárních operací. Proto zavedeme následující pojem:

5.6.1 Definice Řekneme, že finitární typ Ω je *silně finitární*, když obsahuje pouze konečně mnoho neprázdných konečných množin Ω_n.

V silně finitárním typu tedy specifikujeme pouze konečně mnoho finitárních operací. To nám dovolí obecně říci, co je rovnicová algebraická specifikace:

5.6.2 Definice Jednosortová *rovnicová algebraická specifikace* je dvojice $\mathcal{S} = (\Omega, E)$, kde Ω je silně finitární typ a E je konečná množina formálních rovnic.

Pro specifikaci $\mathcal{S} = (\Omega, E)$ definujeme její *zamýšlený model*: jde o algebru typu Ω, která je volná nad prázdnou množinou mezi všemi algebrami splňujícími všechny formální rovnice z množiny E. Této algebře říkáme *iniciální sémantika*[11] algebraické specifikace $\mathcal{S} = (\Omega, E)$. Iniciální sémantika vystihuje přesně doktrínu *no-junk-no-confusion*:

1. *No junk* znamená, že nezavádíme žádné zbytečné symboly. Proto si jako množinu proměnných volíme prázdnou množinu.
2. *No confusion* znamená, že slepujeme pouze ty věci, které slepit musíme. Protože slepování se v algebrách děje pomocí kongruencí, vyžadujeme po kongruenci, aby byla *nejmenší* z těch kongruencí, které vynucují platnost všech formálních rovnic z množiny E.

Podívejme se opět na specifikaci seznamů:

```
1    spec LIST is
2         sorts: list
3         operations: nil: --> list;
4                     _#_:list,list --> list
5         variables: x,y,z:list
6         equations: x#nil=x;
7                    nil#x=x;
8                    x#(y#z)=(x#y)#z
9    endspec
```
(5.10)

Modely této specifikace jsou přesně monoidy, viz definice 5.3.6.

Iniciální sémantika této specifikace je volná algebra nad prázdnou množinou proměnných. Tato algebra obsahuje třídy ekvivalence termů, které můžeme vytvořit: `nil#nil`, `nil#(nil#nil)`, `(nil#nil)#nil` a tak dále. Ovšem například termy `nil#(nil#nil)` a `(nil#nil)#nil` jsou díky asociativitě ztotožněny. Stejně tak,

[10] Anglicky *adjoint functors*.
[11] Algebrám, volným nad prázdnou množinou proměnných, se říká *iniciální*, odtud terminologie.

díky rovnostem pro neutralitu, jsou ztotožněny termy `nil#(nil#nil)` a `nil`. Je snadné zjistit, že zamýšlený model obsahuje pouze jeden prvek, sice třídu ekvivalence prvku `nil` — prázdný seznam!

Znamená to, že specifikace `LIST` je nesprávná? Ne. Když si znovu přečteme specifikaci `LIST`, zjistíme, že jsme pouze specifikovali, jak zřetězovat prázdný seznam. O něco lepší specifikace by byla taková, kde řekneme, že chceme zřetězovat prázdný seznam a dva konkrétní (pevně zadané) seznamy. Ve specifikaci `LIST_2` si jako jména těchto seznamů zvolíme `a1` a `a2`:

```
1    spec LIST_2 is
2         sorts: list
3         operations: nil: --> list;
4                     a1: --> list;
5                     a2: --> list;
6                     _#_:list,list --> list                    (5.11)
7          variables: x,y,z:list
8          equations: x#nil=x;
9                     nil#x=x;
10                    x#(y#z)=(x#y)#z
11   endspec
```

Podstatný rozdíl se odehrává na řádcích `4` a `5`: specifikovali jsme dvě nové konstanty. Iniciální sémantika specifikace `LIST_2` obsahuje následující prvky (díky asociativitě nezávorkujeme): `nil`, `a1`, `a2`, `a1#a1`, `a2#a2`, `a1#a2`, `a2#a1`, `a1#a1#a1`, `a1#a1#a2`, `a1#a2#a1`, `a1#a2#a2`, `a2#a1#a1`, `a2#a1#a2`, `a2#a2#a1`, `a2#a2#a2`, a tak dále.

5.6.3 Poznámka Mezi inicialitou a volností je ještě jedna souvislost. Viděli jsme, že zobrazení $\langle - \rangle : \Sigma \longrightarrow \Sigma^*$ předkládá Σ^* jako volný monoid nad množinou Σ. Proto iniciální monoid (tj. volný monoid nad prázdnou množinou) obsahuje jediný prvek, sice „prázdný seznam“. Viděli jsme také, že když prvky množiny Σ zavedeme jako *nové konstanty*, je příslušná iniciální algebra isomorfní volnému monoidu nad Σ. To je speciální případ obecného faktu:

> *Ať Ω je finitární typ a ať V je množina (formálních proměnných), disjunktní s množinou formálních konstant. Definujte Ω_V jako finitární typ*
>
> $$\Omega_0 \cup V, \Omega_1, \Omega_2, \ldots$$
>
> *(typ Ω_V tedy vzniká z typu Ω přidáním každé formální proměnné do množiny nových formálních konstant). Pak volná algebra typu Ω nad V je isomorfní s iniciální algebrou typu Ω_V.*

5.6.4 Poznámka Jistě jste si všimli, že úzkostlivě mluvíme a *rovnicových* specifikacích. Rovnice však často k vyjádření toho, co chceme říci, nemusí stačit. Například v definici tělesa (definice 2.3.17) tvrdíme něco, co bychom mohli nazvat *implikací*:

> jestliže $a \neq 0$, pak existuje a^{-1}.

Veškerou techniku k zavedení pojmu *formální implikace* jsme již vybudovali. Přesto se formálním implikacím nebudeme věnovat, protože by si to vyžadovalo zavedení dalších technických pojmů. Odkazujeme na knihu

☞ W. Wechler, *Universal Algebra for Computer Scientists*, Springer-Verlag, Berlin, 1992

5.7 Cvičení

5.7.1 Cvičení Napište (rovnicovou) algebraickou specifikaci okruhu. Modely této specifikace by měly být přesně okruhy.

Pokuste se říci, co jsou „přirozená“ zobrazení mezi dvěma okruhy. Viděli jste již příklady takových zobrazení? (Přečtěte si případně poznámku 4.3.5.)

5.7.2 Cvičení Modelům specifikace `STACK` říkejme zásobníky. Pokuste se říci, co je „přirozené“ zobrazení mezi dvěma zásobníky.

5.7.3 Cvičení Specifikujte vektorové prostory nad pevným (konečným) tělesem $\mathrm{GF}(p^n)$ (viz definice 4.4.9). (Výsledná specifikace by měla být vícesortová. Jaké sorty budete potřebovat?) Modely této specifikace by měly být přesně vektorové prostory nad $\mathrm{GF}(p^n)$. Co jsou „přirozená“ zobrazení mezi vektorovými prostory nad $\mathrm{GF}(p^n)$?

5.7.4 Cvičení Ke specifikaci `BINARY` přidejte rovnice, které odpovídají vlastnostem binární operace z definice 5.3.3. Výsledné specifikace nazvěte `ASSOCIATIVE`, `COMMUTATIVE`, `LEFT_NEUTRAL`, `RIGHT_NEUTRAL` a `NEUTRAL`.

5.7.5 Cvičení Jak poznáme existenci neutrálního prvku z tabulky binární operace? Jak poznáme existenci inverse? Jak poznáme komutativitu?

Pozor! Asociativita se z tabulky poznává těžko. (Viz také cvičení 5.7.6.) Pokud to jde, doplňte následující tabulku tak, aby výsledná binární operace byla asociativní.

$\star$	1	2	3	4
1	1	2	3	4
2	2	1	3	4
3	3	4	3	4
4				

5.7.6 Cvičení Ať $\langle X, \star\rangle$ je grupoid. Pro každé $a \in X$ definujte dvě binární operace:

$$x \underset{a}{\bullet} y = x \star (a \star y) \qquad x \underset{a}{\blacksquare} y = (x \star a) \star y$$

Dokažte, že $\star$ je asociativní operace právě tehdy, když jsou operace $\underset{a}{\bullet}$ a $\underset{a}{\blacksquare}$ stejné pro všechna $a \in X$.

Na základě výše uvedeného navrhněte algoritmus, který zjistí asociativitu operace $\star$ zadané tabulkou. (Návod: promyslete, jak vypadá například tabulka operace $\underset{a}{\bullet}$.) Dokažte totální korektnost tohoto algoritmu.

5.7.7 Cvičení Na množině reálných čísel $\mathbb{R}$ je definována binární operace $\star$ následovně:

$$x \star y = x + y + x^2 y$$

Ukažte, že platí:

1. Operace $\star$ má neutrální prvek e.
2. Pro každé x existuje právě jedno y tak, že $x \star y = e$. Tomuto jednoznačně určenému y říkáme *pravá inverse k* x.
3. Existuje x tak, že rovnost $y \star x = e$ neplatí pro žádné y. To znamená, že *levá inverse k* x obecně neexistuje.

5.7.8 Cvičení [12] Ať $\langle X, \star\rangle$ je pologrupa. Předpokládejte, že existuje pravý neutrální prvek e_r a že pro každé x existuje právě jedna pravá inverse x_r (tj. platí $x \star x_r = e_r$). Ukažte, že potom $\langle X, \star\rangle$ musí být grupa.

Postupujte následovně:

1. Ukažte, že pro všechna $x, y, a \in X$ platí: jestliže $x \star a = y \star a$, potom $x = y$. (Tj. v $\langle X, \star\rangle$ lze *krátit zprava*.)
2. Ukažte, že e_r je levý neutrální prvek, tj. pro všechna $x \in X$ platí $e_r \star x = x$. (Návod: ukažte, že pro každé $x \in X$ platí $(e_r \star x) \star x_r = x \star x_r$. K tomu budete potřebovat asociativitu.)
3. Ukažte, že x_r je levá inverse k x, tj. platí $x_r \star x = e_r$. (Návod: ukažte, že pro každé $x \in X$ platí $(x_r \star x) \star x_r = e_r \star x_r$. Zde opět využijte asociativitu.)

Co lze tedy usoudit o grupoidu z příkladu 5.7.7?

[12] Porovnejte toto cvičení s větou 5.3.8: toto cvičení ukazuje, že na grupu se můžeme dívat jako na pologrupu (tj. pouze „rozumné“ násobení, kde předem nemáme zaručenu existenci neutrálního prvku), kde existuje *pravý neutrální prvek* e_r a kde umíme jednoznačně řešit pouze lineární rovnice tvaru $a \star x = e_r$. Ze všech axiomů pro grupu tak lze říci „půlku“. Taková redukce požadovaných rovnic má často důležité praktické důsledky — jistě nechceme, aby naše specifikace byla zbytečně dlouhá.

5.7.9 Cvičení Označte jako $[M, M]$ množinu všech funkcí z množiny M do množiny M.

1. Ukažte, že $[M, M]$ spolu s operací skládání funkcí $\circ$ tvoří monoid.
2. Ukažte, že množina M má alespoň dva prvky právě tehdy, když operace $\circ$ na $[M, M]$ není komutativní.
3. Charakterisujte všechny prvky f z $[M, M]$ (tj. všechny funkce $f : M \longrightarrow M$) takové, že platí rovnost $f \circ g = f$ pro všechny g z $[M, M]$.

   Takovým prvkům f se v obecném monoidu říká *levé nuly* — chovají se totiž při „násobení“ zleva jako nula.
4. Ukažte, že pokud obecná grupa $\langle X, \star, e\rangle$ obsahuje levou nulu, potom je množina X jednoprvková. Návod: označte levou nulu jako f a ukažte nejprve, že $e = f$. Potom ukažte, že pro všechna $x \in X$ platí $x = e$.

   Odvoďte z toho, že je-li M alespoň dvouprvková množina, potom $\langle [M, M], \circ\rangle$ není grupa.

5.7.10 Cvičení Ať $\langle X, \star\rangle$ je pologrupa. Řekneme, že prvky $a, b \in X$ spolu *komutují*, pokud platí rovnost $a \star b = b \star a$. Ukažte, že platí:

Jestliže a komutuje s x i y, potom a komutuje s $x \star y$.

5.7.11 Cvičení V tomto cvičení zobecníte příklad 5.3.10. Ať $\langle X, \star, e\rangle$ je monoid. Označte jako X^* množinu invertibilních prvků X a ukažte, že X^* je grupa.

5.7.12 Cvičení Následující výsledek je zobecněním malé Fermatovy věty 3.4.7 a Eulerovy věty 3.4.13 ze $\mathbb{Z}_n$ na konečné grupy.

Ať $\langle X, \star, e\rangle$ je konečná grupa mající n prvků. Označme x^n součin n činitelů x. Pak platí $x^n = e$.

Dokažte tuto větu. Návod: inspirujte se důkazem věty 3.4.13. Vysvětlete, jak z této věty plyne plyne Eulerova věta.

5.7.13 Cvičení V tomto cvičení ukážeme, že grupy jsou modely specifikace práce se zlomky.

1. Ať $\langle X, \star, e, (-)^{-1}\rangle$ je grupa. Definujte x/y jako $x \star y^{-1}$.
   (a) Dokažte, že $\langle X, /\rangle$ je grupoid.
   (b) Dokažte, že pro všechna $x, y, z \in X$ platí následující čtyři rovnice:
      i. $x/x = e$,
      ii. $x/e = x$,
      iii. $e/(x/y) = y/x$,
      iv. $(x/z)/(y/z) = x/y$.

   Dokažte, že jestliže $\star$ je komutativní, pak pro všechna $x, y \in X$ platí rovnice $(x \star y)/x = y$.
2. Ať $\langle X, /\rangle$ je grupoid, ve kterém existuje prvek $e \in X$ tak, že platí rovnosti i.–iv. Definujte $x \star y$ jako $x/(e/y)$. Dokažte, že $\langle X, \star\rangle$ je grupa. Postupujte přitom podle následujících kroků:
   (a) Ukažte, že $\star$ je opravdu binární operace na množině X.
   (b) Dokažte, že e je neutrální prvek operace $\star$.
   (c) Definujte x^{-1} jako e/x a dokažte, že x^{-1} je inverse x vzhledem k $\star$.
   (d) K důkazu asociativity $\star$ dokažte nejdříve, že pro všechna $x, y, z \in X$ platí následujících pět podmínek:
      v. $(x \star z)/(y \star z) = x/y$,
      vi. $(x/z) \star (z/y) = x/y$,
      vii. jestliže $x/y = e$, pak $x = y$,
      viii. jestliže $x/z = y/z$, pak $x = y$,
      ix. $(x \star y)/y = x$.

Nyní dokažte rovnost

$$(x \star (y \star z))/(y \star z) = ((x \star y) \star z))/(y \star z)$$

a z ní odvoďte rovnici asociativity $x \star (y \star z) = (x \star y) \star z$.

Dokažte, že pokud pro všechna $x, y \in X$ platí rovnost $(x \star y)/x = y$, pak je operace $\star$ komutativní. (Návod: dokažte, že z rovnosti $(x \star y)/x = y$ plyne $(x \star y)/(y \star x) = e$. Pak použijte vii.)

Napište specifikaci zlomků `FRACTIONS`.

5.7.14 Cvičení Na množině $P = \{1, 2, 3, 4, 5\}$ je zadána binární operace $\star$ následující tabulkou:

$\star$	1	2	3	4	5
1	2	3	4	2	3
2	3	4	2	3	4
3	4	2	3	4	2
4	2	3	4	2	3
5	3	4	2	3	4

Dokažte, že grupoid $\langle P, \star \rangle$ není grupa. Dokažte, že existuje množina $A \subseteq P$ tak, že A je uzavřená na operaci $\star$ a $\langle A, \star \rangle$ je grupa.

5.7.15 Cvičení Dejte příklad grupoidů $\langle X, \star_X \rangle$, $\langle Y, \star_Y \rangle$ a zobrazení $f : X \longrightarrow Y$, které *není* homomorfismem grupoidů.

5.7.16 Cvičení Dokažte, že $\langle \mathbb{R}, + \rangle$ (reálná čísla s operací sčítání) a $\langle (0, +\infty), \cdot \rangle$ (kladná reálná čísla s operací násobení) jsou grupy. Dokažte, že zobrazení

$$\ln : (0, +\infty) \longrightarrow \mathbb{R}$$

(přirozený logaritmus) je homomorfismus grup. Popište přesně, které věty z matematické analýzy používáte. Jde o isomorfismus grup?

5.7.17 Cvičení Připomeňte si, že $\mathbb{R}$ je těleso, a proto $\langle \mathbb{R} \setminus \{0\}, \cdot \rangle$ (nenulová reálná čísla s operací násobení) je grupa.

Dokažte, že zobrazení

$$\det : \mathsf{RegMat}_{n \times n}(\mathbb{R}) \longrightarrow \mathbb{R} \setminus \{0\}$$

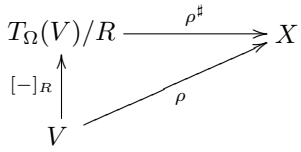
(determinant) je homomorfismus grup (viz příklad 5.3.10). Zmiňte všechny věty z klasické lineární algebry, které k důkazu potřebujete.

5.7.18 Cvičení Budou tvrzení ze cvičení 5.7.17 platit, když nahradíte těleso $\mathbb{R}$ obecným tělesem $\mathbb{K}$?

5.7.19 Cvičení Ať E je množina formálních rovnic a ať R je nejmenší kongruence na $\mathbb{T}_\Omega(V)$, která ztotožňuje všechny dvojice termů z množiny E. Dokažte, že $\mathbb{T}_\Omega(V)/R$ je volná algebra nad V mezi všemi algebrami splňujícími E. Přesněji, dokažte:

Kanonické zobrazení $[-]_R : V \longrightarrow T_\Omega(V)$, které zobrazuje formální proměnnou $v \in V$ na třídu ekvivalence $[v]_R$, splňuje universální vlastnost:

Ať $\mathbb{X}$ je jakákoli algebra splňující všechny rovnice z množiny E a ať $\rho : V \longrightarrow X$ je libovolné zobrazení (kde X je nosná množina algebry $\mathbb{X}$). Pak existuje jediný homomorfismus $\rho^\sharp : \mathbb{T}_\Omega(V)/R \longrightarrow \mathbb{X}$ algeber tak, že diagram



komutuje.

5.7.20 Cvičení Popište volné pologrupy.

5.7.21 Cvičení Popište iniciální sémantiku specifikace

```
1    spec NAT is
2         sorts: nat
3         operations: zero: --> nat;
4                     succ(_):nat --> nat
5    endspec
```
(5.12)

5.7.22 Cvičení Popište iniciální sémantiku specifikace

```
1    spec NAT_2 is
2         sorts: nat
3         operations: zero: --> nat;
4                     one: --> nat;
5                     _+_: nat,nat --> nat
6         variables: x,y,z: nat
7         equations: x+(y+z)=(x+y)+z;
8                    x+zero=x;
9                    zero+x=x;
10                   x+y=y+x
11   endspec
```
(5.13)

5.7.23 Cvičení Pouze pro statečné: vyberte si svůj oblíbený imperativní jazyk a zadejte jeho syntaxi jako (vícesortovou) algebraickou specifikaci. Co jsou modely a homomorfismy této specifikace? (Návod: přemýšlejte o implementaci programovacího jazyka.)

Revize kapitoly

Dozvěděli jsme se:

- ✔ Při rovnicovém popisu abstraktního datového typu je třeba rozlišit tři věci: sorty, operace a rovnice. Tyto tři věci říkají *kde*, *s čím* a *jak*.
  - ✔ Specifikace sort říká, s jakými *typy dat* chceme pracovat.
  - ✔ Při specifikaci operací říkáme, jaké *arity* mají mít naše operace. Specifikujeme tedy typ a počet argumentů a typ výsledku operace.
  - ✔ Při specifikaci rovnic popisujeme *zákonitosti*, které mají specifikované objekty splňovat.
- ✔ Modely specifikací jsou *algebry*, přirozená zobrazení mezi algebrami jsou *homomorfismy*.
- ✔ Homomorfismus musí *respektovat všechny specifikované operace*, respektování rovností se děje automaticky.
- ✔ Iniciální algebry jsou *no-junk-no-confusion* sémantikou rovnicových specifikací.

Pro přípravu na zkoušku zkuste zodpovědět následující otázky:

- ✔ Vyberte si nějakou syntaktickou konstrukci svého oblíbeného imperativního programovacího jazyka. Dovedete ji popsat rovnicovou specifikací?
- ✔ Dejte příklad *junk* modelu nějaké specifikace.
- ✔ Dejte příklad *confusion* modelu nějaké specifikace.

Doplňující literatura

Kromě vynikající knihy o universální algebře

☞ W. Wechler, *Universal Algebra for Computer Scientists*, Springer-Verlag, Berlin, 1992

kterou jsme v této kapitole několikrát doporučovali, se lze o algebraických specifikacích dočíst například v knize

☞ J. A. Bergstra, J. Heering a P. Klint, *Algebraic Specification*, Addison-Wesley, New York, 1989

ve které je ovšem pro zápis specifikací použit jiný jazyk, než náš zápis v pseudo-OBJ3. Jazyk OBJ3 je naopak použit ve skriptu

☞ K. Richta a J. Velebil, *Sémantika programovacích jazyků*, Karolinum, Praha 1997

a dobrým úvodem do OBJ3 je kniha

☞ J. A. Goguen, D. Coleman a R. Gallimore (eds.), *Applications of Algebraic Specification Using* OBJ, Cambridge University Press, 1992

Teorie grup má, kromě aplikací v jiných oblastech (například v moderní fyzice), významné místo při studiu kryptosystémů. Například v kapitole VI knihy

☞ N. Koblitz, *A Course in Number Theory and Cryptography*, Springer, New York, 1994

je popsána teorie eliptických křivek nad (konečným) tělesem a kryptosystémů na nich založených.

V současnosti prožívá velký rozmach *teorie koalgeber*, umožňující popisovat i nekonečné chování datových typů, viz například

☞ J. J. M. M. Rutten, *Elements of Stream Calculus (An Extensive Exercise in Coinduction)*, Technical Report SEN-R0120, CWI, Amsterdam, 2001, http://homepages.cwi.nl/∼janr/papers/

☞ P. Aczel, J. Adámek, S. Milius a J. Velebil: Infinite Trees and Completely Iterative Theories — A Coalgebraic View (A Fundamental Study), *Theoret. Comput. Sci.*, 300 (2003), 1–45

Kapitola 6

Abstraktní výpočty — část 2

> Všechny součástky by do sebe měly zapadat bez použití síly. Pamatujte, že součástky, které skládáte, jste rozložili Vy. Pokud je tedy složit nemůžete, musí k tomu být důvod. V žádném případě nepoužívejte kladivo.
>
> manuál IBM

V této kapitole se nejdříve zaměříme na svazy (struktura zavedená v příkladu 5.4.7). Ukážeme, že svazy jsou přirozeným začátkem specifikace pravdivostních hodnot. Přesněji, svazy jsou modely rovnicové specifikace základního chování logických spojek `and` a `or`. Na svazy však můžeme také pohlížet jako na speciální *posety* (viz definice 6.1.1), ve kterých lze počítat *infima* a *suprema* všech dvojic prvků. Tento dvojí pohled je velmi užitečný — často je jednodušší zjistit, zda existují infima a suprema všech dvojic, než ověřovat osm axiomů pro svaz.

Chceme-li ovšem specifikovat *klasické* pravdivostní hodnoty, chceme víc než jen spojky `and` a `or`: požadujeme pravdivostní hodnoty `true` a `false`, požadujeme unární operaci `negation(_)` a výsledná bohatá struktura by navíc měla splňovat další rovnice. To nás v definici 6.3.3 přivede k pojmu *Booleova algebra*.

Nakonec se budeme v plné obecnosti věnovat (rovnicovým) algebraickým specifikacím. Zavedeme *vícesortové algebry* a v odstavci 6.4 podáme *iniciální sémantiku* algebraických specifikací.

6.1 Znovu svazy

Předpokládejme, že chceme specifikovat základní chování logických spojek `and` a `or`. Mohli bychom tak napsat například následující specifikaci:

```
1   spec BASIC_TRUTH_VALUES is
2        sorts: values
3        operations: _and_:values,values --> values;
4                    _or_:values,values --> values
5        variables: x,y,z:values
6        equations: x and y = y and x;
7                   (x and y) and z = x and (y and z);
8                   x and x = x;
9                   x or (x and y) = x;
10                  x or y = y or x;
11                  (x or y) or z = x or (y or z);
12                  x or x = x;
13                  x and (x or y) = x
14    endspec
```
(6.1)

Jistě jste poznali, že řádky 6–13 popisují přesně oněch osm rovnic z příkladu 5.4.7, takže modely této specifikace jsou přesně svazy.

Nyní ukážeme jinou techniku, která nám dovolí poznat, zda daná struktura je svaz nebo není. Tato technika je založena na tom, že svazy jsou množiny vybavené jistým uspořádáním. Připomeňte si poznámku 2.2.4.

6.1.1 Definice *Poset* (také: *uspořádaná množina*) je dvojice $\langle X, \sqsubseteq \rangle$, kde X je množina a $\sqsubseteq$ je relace uspořádání na X.

6.1.2 Definice Ať $\langle X, \sqsubseteq \rangle$ je poset, $M \subseteq X$. Řekneme, že prvek $x \in X$ je:

1. *Horní odhad množiny* M, pokud platí $m \sqsubseteq x$ pro všechna $m \in M$.
2. *Dolní odhad množiny* M, pokud platí $x \sqsubseteq m$ pro všechna $m \in M$.
3. *Nejmenší horní odhad množiny* M (také: *supremum množiny* M, značení $\sup_{\sqsubseteq} M$), pokud je x nejmenší ze všech horních odhadů, tj. pokud jsou splněny následující dvě podmínky:
   (a) Prvek x je horní odhad množiny M.
   (b) Je-li x' horní odhad množiny M, potom platí $x \sqsubseteq x'$.
4. *Největší dolní odhad množiny* M (také: *infimum množiny* M, značení $\inf_{\sqsubseteq} M$), pokud je x největší ze všech horních odhadů, tj. pokud jsou splněny následující dvě podmínky:
   (a) Prvek x je dolní odhad množiny M.
   (b) Je-li x' dolní odhad množiny M, potom platí $x' \sqsubseteq x$.

6.1.3 Příklad V tomto příkladě vysvětlíme, že nejrozumnější představa o supremu množiny M je tato: *supremum M je prvek zeshora „naplácnutý“ na množinu M*. Podobnou (duální) představu lze mít o infimu množiny M.

Reálná čísla $\mathbb{R}$ spolu s klasickou relací $\leq$ tvoří jistě poset: $\leq$ je reflexivní, transitivní a antisymetrická binární relace na množině $\mathbb{R}$.

Zvolme jako M množinu $\{m \in \mathbb{R} \mid m < 3\}$ a ukažme, že její supremum v posetu $\langle \mathbb{R}, \leq \rangle$ je číslo 3:

1. Číslo 3 je horní odhad množiny M, protože pro každé číslo m z množiny M platí nerovnost $m \leq 3$ (platí dokonce ostrá nerovnost).
2. Zvolme jakýkoli horní odhad x' množiny M. Chceme ukázat, že platí $3 \leq x'$.

   Kdyby platila nerovnost $x' < 3$, pak číslo $m = x' + \frac{3-x'}{2}$ je jistě v množině M, ale nerovnost $m \leq x'$ neplatí. To je spor s tím, že x' je horní odhad. Musí tedy platit $3 \leq x'$.

Povšimněme si, že $3 \notin M$. Supremum množiny tedy nemusí být největší prvek množiny (naše množina M největší prvek vůbec nemá).

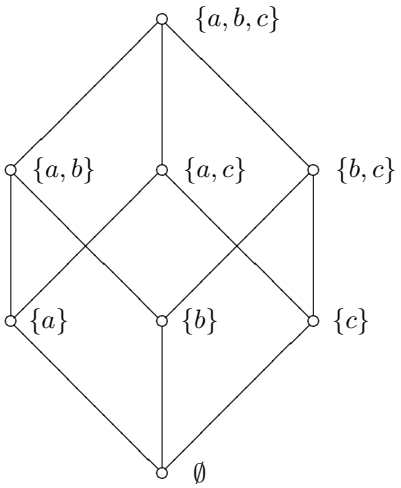
Ukážeme, že každý poset, ve kterém pro každou dvojici prvků existuje *infimum* a *supremum*, lze chápat jako množinu vybavenou dvěma binárními operacemi. Nejprve uvedeme příklad.

6.1.4 Příklad Označme jako X množinu všech podmnožin tříprvkové množiny $\{a, b, c\}$. Množina X má tedy osm různých prvků:

$$\emptyset,\ \{a\},\ \{b\},\ \{c\},\ \{a, b\},\ \{a, c\},\ \{b, c\},\ \{a, b, c\}$$

Prvky množiny X uspořádáme relací inkluse, tj. pro $x, y \in X$ je x „menší nebo rovno“ y právě tehdy, když $x \subseteq y$. Protože inkluse je reflexivní, transitivní a antisymetrická binární relace, jde skutečně o uspořádání.

Uspořádanou množinu $\langle X, \subseteq \rangle$ můžeme zakreslit pomocí Hasseho diagramu (viz poznámku 6.1.6) takto:



(6.2)

Tento orientovaný graf (každá hrana je orientována směrem odzdola nahoru) znázorňuje pouze nezbytně nutnou informaci k zadání uspořádané množiny. Celá relace uspořádání je reflexivním a transitivním obalem relace z obrázku.

Zvolme nějakou dvojici prvků této uspořádané množiny, například dvojici $\{a\}$, $\{c\}$.

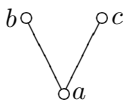
1. Horní odhady dvojice $\{a\}$, $\{c\}$ jsou: $\{a, c\}$, $\{a, b, c\}$. Jsou to ty prvky množiny X, které jsou větší nebo rovny oběma prvkům $\{a\}$, $\{c\}$ současně (tj. shora odhadují $\{a\}$ a $\{c\}$).
2. Nejmenší (ve smyslu uspořádání inklusí) horní odhad dvojice $\{a\}$, $\{c\}$ je: $\{a, c\}$. Nejmenšímu hornímu odhadu se říká *supremum*.
3. Jediným dolním odhadem dvojice $\{a\}$, $\{c\}$ je: $\emptyset$. Je to ten prvek množiny X, které je menší nebo roven oběma prvkům $\{a\}$, $\{c\}$ současně (tj. zdola odhaduje $\{a\}$ a $\{c\}$).
4. Největší (ve smyslu uspořádání inklusí) dolní odhad dvojice $\{a\}$, $\{c\}$ je: $\emptyset$. Největšímu dolnímu odhadu se říká *infimum*.

Podobnými úvahami bychom zjistili, že supremum a infimum existuje pro každou dvojici prvků v X. Navíc, infimum prvků $x, y \in X$ lze „spočítat“ přímo — je to množina $x \cap y$. Podobně, jejich supremum je množina $x \cup y$.

Suprema a infima dvojic tedy nejenom existují, ale jsou dány hodnotami dvou binárních operací na množině X, sice sjednocením a průnikem množin.

Dále uvedeme příklad uspořádané množiny, kde některá suprema dvojic existují, neexistují však pro všechny dvojice.

6.1.5 Příklad Označme $X = \{a, b, c\}$ a nakresleme Hasseho diagram



Supremum dvojice b a c neexistuje, protože nelze najít žádný (tím pádem ani nejmenší) horní odhad. Uvědomme si ale, že zde existují infima všech dvojic. Obecně tedy z existencí infim všech dvojic neplyne existence suprem všech dvojic.

6.1.6 Poznámka Obrázkům posetů z předchozích příkladů říkáme *Hasseho*[1] *diagramy*. V takových obrázcích kreslíme jen nezbytně nutnou informaci o daném uspořádání: kreslíme jen takzvané *sousedy* v relaci $\sqsubseteq$. Obecně, dva prvky x, y posetu $\langle X, \sqsubseteq \rangle$ jsou sousedé, pokud ze vztahu $x \sqsubseteq z \sqsubseteq y$ plyne $x = z$ nebo $y = z$.

[1] Pojmenováno podle německého matematika Helmuta Hasseho (1898–1979).

6.1.7 Tvrzení *V uspořádané množině $\langle X, \sqsubseteq\rangle$ má každá dvojice prvků nejvýše jedno supremum a nejvýše jedno infimum.*

Důkaz. Dokážeme pouze tvrzení o supremech, tvrzení o infimech dokážeme analogicky.

Předpokládejme, že s_1 a s_2 jsou suprema dvojice prvků x, y. Potom musí platit $s_1 \sqsubseteq s_2$ (protože s_1 je supremum a s_2 je horní odhad) a současně $s_2 \sqsubseteq s_1$ (protože s_2 je supremum a s_1 je horní odhad). Protože $\sqsubseteq$ je antisymetrická relace, platí $s_1 = s_2$. ■

Příklad 6.1.4 je typický: uspořádaná množina $\langle X, R\rangle$, ve které existují suprema a infima všech dvojic, dává vzniknout dvěma binárním operacím

$$\begin{aligned}(x,y) &\mapsto \sup_{\sqsubseteq}\{x,y\}\\ (x,y) &\mapsto \inf_{\sqsubseteq}\{x,y\}\end{aligned}$$

na množině X. Ukážeme, že množina X spolu s takto vzniklými operacemi je svaz.

Připomeňme, že *svaz* je množina X vybavená dvojicí binárních operací

$$\wedge : X \times X \longrightarrow X \quad \text{a} \quad \vee : X \times X \longrightarrow X$$

a tyto operace splňují pro všechna $x, y, z \in X$ následující sadu rovností:

1. *Komutativita* $\wedge$: $x \wedge y = y \wedge x$.
2. *Asociativita* $\wedge$: $x \wedge (y \wedge z) = (x \wedge y) \wedge z$.
3. *Idempotence* $\wedge$: $x \wedge x = x$.
4. *Absorpce* $\wedge$: $x \vee (x \wedge y) = x$.
5. *Komutativita* $\vee$: $x \vee y = y \vee x$.
6. *Asociativita* $\vee$: $x \vee (y \vee z) = (x \vee y) \vee z$.
7. *Idempotence* $\vee$: $x \vee x = x$.
8. *Absorpce* $\vee$: $x \wedge (x \vee y) = x$.

Zobecněním situace z příkladu 6.1.4 je následující věta.

6.1.8 Věta *Ať $\langle X, \sqsubseteq\rangle$ je uspořádaná množina, ve které existují suprema a infima všech dvojic prvků množiny X. Definujte binární operace $\wedge$ a $\vee$ takto:*

$$\begin{aligned}x \vee y &= \sup_{\sqsubseteq}\{x,y\}\\ x \wedge y &= \inf_{\sqsubseteq}\{x,y\}\end{aligned}$$

Potom je $\langle X, \wedge, \vee\rangle$ svaz.

Důkaz. Protože předpokládáme, že supremum existuje pro každou dvojici prvků a protože supremum je určeno jednoznačně (viz tvrzení 6.1.7), definuje rovnost

$$x \vee y = \sup_{\sqsubseteq}\{x,y\}$$

skutečně binární operaci $\vee$ na množině X. Podobně je binární operací i $\wedge$. Zbývá tedy pro $\vee$ a $\wedge$ ověřit platnost osmi rovností z definice svazu. Ověříme rovnosti 1. až 4.:

1. *Komutativita* $\wedge$: rovnost $x \wedge y = y \wedge x$ platí, protože $\inf_{\sqsubseteq}\{x,y\} = \inf_{\sqsubseteq}\{y,x\}$.

2. *Asociativita* $\wedge$: chceme ukázat rovnost $x \wedge (y \wedge z) = (x \wedge y) \wedge z$. Stačí ukázat

   (a) $x \wedge (y \wedge z) \sqsubseteq (x \wedge y) \wedge z$

   (b) $(x \wedge y) \wedge z \sqsubseteq x \wedge (y \wedge z)$

   a použít antisymetrii relace uspořádání $\sqsubseteq$. Ukážeme platnost prvního vztahu, druhý vztah se ověří analogicky. Protože $x \wedge (y \wedge z) = \inf_{\sqsubseteq}\{x, \inf_{\sqsubseteq}\{y,z\}\}$ a $(x \wedge y) \wedge z = \inf_{\sqsubseteq}\{\inf_{\sqsubseteq}\{x,y\}, z\}$, stačí ukázat, že $\inf_{\sqsubseteq}\{x, \inf_{\sqsubseteq}\{y,z\}\}$ je dolní odhad pro dvojici $\inf_{\sqsubseteq}\{x,y\}$ a z.

   Vztah $\inf_{\sqsubseteq}\{x, \inf_{\sqsubseteq}\{y,z\}\} \sqsubseteq \inf_{\sqsubseteq}\{x,y\}$ platí, protože levá strana je dolní odhad dvojice x a y.

   Vztah $\inf_{\sqsubseteq}\{x, \inf_{\sqsubseteq}\{y,z\}\} \sqsubseteq z$ platí, protože platí $\inf_{\sqsubseteq}\{y,z\} \sqsubseteq z$.

3. *Idempotence* $\wedge$: rovnost $x \wedge x = x$ platí, protože $\inf_{\sqsubseteq}\{x, x\} = \inf_{\sqsubseteq}\{x\} = x$.

4. *Absorpce* $\wedge$: pro důkaz rovnosti $x \vee (x \wedge y) = x$ stačí ukázat, že x je supremem dvojice x a $x \wedge y$.

   (a) x je horní odhad, protože platí $x \sqsubseteq x$ a $x \wedge y \sqsubseteq x$. První vztah platí díky reflexivitě $\sqsubseteq$ a druhý plyne z toho, že $x \wedge y = \inf_{\sqsubseteq}\{x, y\}$ a z toho, že infimum je dolní odhad.

   (b) Abychom ukázali, že x je nejmenší horní odhad, vezměme libovolné z takové, že $x \sqsubseteq z$ a současně $x \wedge y \sqsubseteq z$. Chceme ukázat, že platí $x \sqsubseteq z$. To je ale triviálně splněno.

Zbytek, tj. rovnice 5.–8., se dokáže analogicky. ■

Platí však i obrácení předchozí věty: jakmile zadáme svaz, vzniká tak uspořádaná množina, ve které má každá dvojice prvků supremum a infimum.

6.1.9 Věta *Ať $\langle X, \wedge, \vee\rangle$ je svaz. Pro libovolné $x, y \in X$ platí:*

$$x \wedge y = x \quad \textit{právě tehdy, když} \quad x \vee y = y$$

Definujeme-li na X binární relaci $\sqsubseteq$ předpisem

$$x \sqsubseteq y \quad \textit{právě tehdy, když} \quad x \wedge y = x$$

potom je relace $\sqsubseteq$ reflexivní, transitivní a antisymetrická. Dvojice $\langle X, \sqsubseteq\rangle$ je tedy uspořádaná množina.

Navíc v $\langle X, \sqsubseteq\rangle$ existují suprema a infima všech dvouprvkových množin a platí

$$\begin{aligned} \sup_{\sqsubseteq}\{x, y\} &= x \vee y \\ \inf_{\sqsubseteq}\{x, y\} &= x \wedge y \end{aligned}$$

Důkaz. Platí-li $x \wedge y = x$, potom platí $x \vee y = (x \wedge y) \vee y$ a použijeme-li nyní komutativitu $\vee$ a $\wedge$ a absorpci $\wedge$, dostaneme rovnost $x \vee y = y$.

Obrácená implikace (jestliže $x \vee y = y$, potom $x \wedge y = x$) se dokáže analogicky.

Ukážeme nyní, že námi definovaná binární relace $\sqsubseteq$ je reflexivní, transitivní a antisymetrická.

1. Reflexivita: pro všechna $x \in X$ platí $x \sqsubseteq x$, protože $x \wedge x = x$.

2. Transitivita: zvolme $x, y, z \in X$ tak, že platí $x \sqsubseteq y$ a současně $y \sqsubseteq z$. Chceme ukázat, že platí $x \sqsubseteq z$, neboli, že $x \wedge z = x$.

   Platí $x \wedge z = (x \wedge y) \wedge z$, protože $x = x \wedge y$. Protože $\wedge$ je asociativní, dostáváme $(x \wedge y) \wedge z = x \wedge (y \wedge z)$. Protože $y \wedge z = y$, platí $x \wedge (y \wedge z) = x \wedge y = x$. To jsme chtěli ukázat.

3. Antisymetrie: zvolme $x, y \in X$ tak, že platí $x \sqsubseteq y$ a současně $y \sqsubseteq x$. Chceme ukázat, že platí $x = y$.

   Protože $x \wedge y = x$ a $y \wedge x = y$, je $x = y$, neboť $\wedge$ je komutativní.

Ukázali jsme, že relace $\sqsubseteq$ je uspořádání. Zbývá ukázat, že existují suprema a infima všech dvojic. Ukážeme, že $x \vee y$ je supremum dvojice x, y.

1. Abychom ukázali, že $x \vee y$ je horní odhad dvojice x a y, musíme ukázat, že $x \sqsubseteq x \vee y$ a $y \sqsubseteq x \vee y$. Ukážeme platnost prvního vztahu, platnost druhého se ukáže analogicky.

   Vztah $x \sqsubseteq x \vee y$ platí právě tehdy, když platí rovnost $x \wedge (x \vee y) = x$. Poslední rovnost plyne z absorpce operace $\vee$.

2. Ukážeme, že $x \vee y$ je nejmenší horní odhad dvojice x a y. Zvolíme tedy libovolné z tak, že platí $x \sqsubseteq z$ a současně $y \sqsubseteq z$ a chceme ukázat, že platí $x \vee y \sqsubseteq z$. Budeme využívat toho, že $a \sqsubseteq b$ platí právě tehdy, když $a \vee b = b$ (viz první část důkazu).

   Platí $(x \vee y) \vee z = x \vee (y \vee z)$, protože $\vee$ je asociativní. Protože $y \sqsubseteq z$, je $x \vee (y \vee z) = x \vee z$ a protože $x \sqsubseteq z$, platí $x \vee z = z$. Celkově $(x \vee y) \vee z = z$, čili platí $x \vee y \sqsubseteq z$.

Nyní bychom měli dokázat, že $x \wedge y$ je infimum x a y. To se dokáže analogicky. ■

Věty 6.1.8 a 6.1.9 tak ukazují, že osm rovnic pro operace ve svazu (a tudíž osm rovnic pro základní chování logických spojek `and` a `or`) popisuje přesně chování suprem a infim dvojic prvků.

V následujících dvou příkladech předvedeme typický způsob „abstraktních výpočtů“ ve svazu.

6.1.10 Příklad Uspořádání $\sqsubseteq$ ve svazu $\langle X, \wedge, \vee\rangle$ si můžeme představit, jako *relaci dokazatelnosti* z logiky.

1. Prvky množiny X si představujeme jako formule a vztah $x \sqsubseteq y$ jako fakt, že z platnosti formule x lze *dokázat* platnost formule y.
2. Fakta o supremech a infimech se pak stávají známými fakty z teorie důkazů matematické logiky:
   (a) Platnost nerovnosti $x \sqsubseteq y \wedge z$ je ekvivalentní platnosti *dvou* nerovností *současně*: $x \sqsubseteq y$ a $x \sqsubseteq z$. To je známý fakt z logiky: důkaz toho, že z předpokladu x plyne konjunkce $y \wedge z$ je podán právě tehdy, když umíme sestavit *dva* důkazy *současně*, sice důkaz y z předpokladu x a důkaz z z předpokladu x. Je tento překlad do jazyka logiky správný? Ano, je. Víme totiž, že $\wedge$ je infimum v uspořádání $\sqsubseteq$, a proto platí:
   $$x \sqsubseteq y \wedge z \quad \text{právě když} \quad x \sqsubseteq y \text{ a současně } x \sqsubseteq z$$
   (b) Platnost nerovnosti $x \vee y \sqsubseteq z$ je ekvivalentní platnosti *dvou* nerovností *současně*: $x \sqsubseteq z$ a $y \sqsubseteq z$. To je známý fakt z logiky: důkaz toho, že z předpokladu $x \vee y$ plyne konjunkce z je podán právě tehdy, když umíme sestavit *dva* důkazy *současně*, sice důkaz z z předpokladu x a důkaz z z předpokladu y. Formálním zdůvodněním je fakt, že $\vee$ je supremum v uspořádání $\sqsubseteq$:
   $$x \vee y \sqsubseteq z \quad \text{právě když} \quad x \sqsubseteq z \text{ a současně } y \sqsubseteq z$$

6.1.11 Příklad V tomto příkladu začneme důkaz toho, že každý svaz je „téměř distributivní“. Ukážeme totiž, že v každém svazu $\langle X, \wedge, \vee\rangle$ platí pro všechna x, y, z *distributivní nerovnost*

$$x \vee (y \wedge z) \sqsubseteq (x \vee y) \wedge (x \vee z),$$

kde $\sqsubseteq$ je uspořádání z věty 6.1.9. Využijeme pohledu na výpočty ve svazu z příkladu 6.1.10.

1. Stačí ukázat $x \sqsubseteq (x \vee y) \wedge (x \vee z)$ a současně $y \wedge z \sqsubseteq (x \vee y) \wedge (x \vee z)$.
2. Pro nerovnost $x \sqsubseteq (x \vee y) \wedge (x \vee z)$ stačí ukázat, že $x \sqsubseteq (x \vee y)$ a současně $x \sqsubseteq (x \vee z)$.
   Obě tyto nerovnosti jsou však triviální, protože $x \vee y$ je horním odhadem x a $x \vee z$ také.
3. Pro nerovnost $y \wedge z \sqsubseteq (x \vee y) \wedge (x \vee z)$ stačí ukázat, že $y \wedge z \sqsubseteq x \vee y$ a současně $y \wedge z \sqsubseteq x \vee z$.
4. Ukážeme nerovnost $y \wedge z \sqsubseteq x \vee y$. Pozor! Už nelze dál „rozkládat“, protože logické spojky stojí na špatných stranách nerovnosti. Využijeme ale transitivitu relace $\sqsubseteq$: platí totiž zjevně $y \wedge z \sqsubseteq y$ a současně $y \sqsubseteq x \vee y$. Proto (transitivita $\sqsubseteq$) platí $y \wedge z \sqsubseteq x \vee y$.
5. Ukážeme nerovnost $y \wedge z \sqsubseteq x \vee z$. I zde jsou logické spojky na špatných stranách nerovnosti a i zde využijeme transitivitu $\sqsubseteq$: platí totiž zjevně $y \wedge z \sqsubseteq z$ a současně $z \sqsubseteq x \vee z$. Proto (transitivita $\sqsubseteq$) platí $y \wedge z \sqsubseteq x \vee z$.
6. Důkaz distributivní nerovnosti je hotov.

Máte-li rádi notaci deduktivních pravidel, můžeme předchozí úvahy znázornit tak, jak se to většinou v logice dělá:

$$\dfrac{\dfrac{}{x \sqsubseteq x \vee y} \quad \dfrac{}{x \sqsubseteq x \vee z}}{x \sqsubseteq (x \vee y) \wedge (x \vee z)} \qquad \dfrac{\dfrac{\dfrac{}{y \wedge z \sqsubseteq y} \quad \dfrac{}{y \sqsubseteq x \vee y}}{y \wedge z \sqsubseteq x \vee y} \quad \dfrac{\dfrac{}{y \wedge z \sqsubseteq z} \quad \dfrac{}{z \sqsubseteq x \vee z}}{y \wedge z \sqsubseteq x \vee z}}{y \wedge z \sqsubseteq (x \vee y) \wedge (x \vee z)}$$

$$\dfrac{x \sqsubseteq (x \vee y) \wedge (x \vee z) \qquad y \wedge z \sqsubseteq (x \vee y) \wedge (x \vee z)}{x \vee (y \wedge z) \sqsubseteq (x \vee y) \wedge (x \vee z)}$$

Viz také cvičení 6.5.8.

6.2 Distributivní svazy

Specifikace (6.1) `BASIC_TRUTH_VALUES` popisuje skutečně jen základní vlastnosti spojek `and` a `or`. „Rozumný“ model spojek `and` a `or` by navíc měl splňovat *distributivní zákony*, tj. pro všechny prvky x, y a z v našem svazu by měly platit rovnice

$$\begin{aligned} x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \\ x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \end{aligned}$$

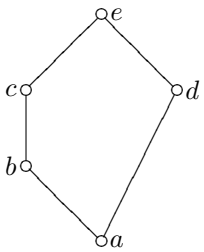
Specifikaci (6.1) tedy vylepšíme následujícím způsobem:

```
 1  spec DISTRIBUTIVE_TRUTH_VALUES is
 2       sorts: values
 3       operations: _and_:values,values --> values;
 4                   _or_:values,values --> values
 5       variables: x,y,z:values
 6       equations: x and y = y and x;
 7                  (x and y) and z = x and (y and z);
 8                  x and x = x;
 9                  x or (x and y) = x;
10                  x or y = y or x;
11                  (x or y) or z = x or (y or z);
12                  x or x = x;
13                  x and (x or y) = x;
14                  x and (y or z) = (x and y) or (x and z);
15                  x or (y and z) = (x or y) and (x or z)
16   endspec
```
(6.3)

Vše, co jsme udělali, je specifikace dvou nových rovnic (řádky `14` a `15`). Je tato změna podstatná? Neplynou řádky `14` a `15` z rovnic na řádcích `6`–`13`?

V dalším příkladu ukážeme, že přidání distributivních zákonů *je* podstatnou změnou. Předvedeme svaz (tj. model specifikace `BASIC_TRUTH_VALUES`), který není modelem specifikace `DISTRIBUTIVE_TRUTH_VALUES`.

6.2.1 Příklad Ať $X = \{a, b, c, d, e\}$ a ať uspořádání na X je definováno následovně:



Protože pro všechny dvojice existují suprema a inifima, je tento poset svaz. Neplatí však ani jeden z distributivních zákonů.

1. $c \wedge (b \vee d) \neq (c \wedge b) \vee (c \wedge d)$. Levá strana je rovna c a pravá strana je rovna b.
2. $b \vee (c \wedge d) \neq (b \vee c) \wedge (b \vee d)$. Levá strana je rovna b a pravá strana je rovna c.

V obecném svazu tedy žádný z distributivních zákonů nemusí platit. Ukážeme, že pokud platí jeden z nich, pak druhý z nich platí automaticky. Proto je jeden z řádků `14`, `15` ve specifikaci `DISTRIBUTIVE_TRUTH_VALUES` přebytečný.

6.2.2 Tvrzení *Ve svazu $\langle X, \vee, \wedge \rangle$ jsou následující podmínky ekvivalentní:*

1. *Pro všechna $x, y, z \in X$ platí $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.*
2. *Pro všechna $x, y, z \in X$ platí $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.*

Důkaz. Ukážeme, že z 1. plyne 2., obrácená implikace se dokáže analogicky.

Protože předpokládáme platnost 1., platí rovnost

$$(x \vee y) \wedge (x \vee z) = \big((x \vee y) \wedge x\big) \vee \big((x \vee y) \wedge z\ \big)$$

Ve výrazu napravo je levá závorka rovna x (absorpce $\vee$) a na pravou závorku použijeme opět distributivní zákon 1.:

$$\big((x \vee y) \wedge x\big) \vee \big((x \vee y) \wedge z\ \big) = \big(x\big) \vee \big((x \wedge z) \vee (y \wedge z)\big)$$

Použijeme-li nyní asociativitu a poté absorpci $\wedge$, dostaneme

$$\big(x\big) \vee \big((x \wedge z) \vee (y \wedge z)\big) = \big(x \vee (x \wedge z)\big) \vee (y \wedge z) = x \vee (y \wedge z)$$

Ukázali jsme, že z 1. plyne 2. ■

Předchozí tvrzení dovoluje specifikaci `DISTRIBUTIVE_TRUTH_VALUES` vylepšit — řádek `15` lze vynechat a modely se tím nezmění!

```
 1   spec DISTRIBUTIVE_TRUTH_VALUES(BETTER) is
 2        sorts: values
 3        operations: _and_:values,values --> values;
 4                    _or_:values,values --> values
 5        variables: x,y,z:values
 6        equations: x and y = y and x;
 7                   (x and y) and z = x and (y and z);
 8                   x and x = x;
 9                   x or (x and y) = x;
10                   x or y = y or x;
11                   (x or y) or z = x or (y or z);
12                   x or x = x;
13                   x and (x or y) = x;
14                   x and (y or z) = (x and y) or (x and z)
15    endspec
```
(6.4)

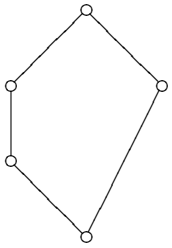
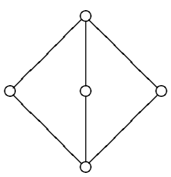
Modely specifikace `DISTRIBUTIVE_TRUTH_VALUES(BETTER)` si zasluhují název: říkáme jim distributivní svazy.

6.2.3 Definice Svaz je *distributivní*, pokud splňuje rovnost $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

6.2.4 Příklad Svaz z příkladu 6.1.4 je distributivní, protože pro libovolné $x, y, z \in X$ platí $x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$ (to je známá vlastnost průniku a sjednocení množin).

Existuje technika, která nám dovolí zjistit, zda daný svaz je či není distributivní? Odpověď je ano: podíváme se na Hasseho diagram daného svazu a hledáme „zakázané podsvazy“. To je smysl následující věty. Uvádíme ji bez důkazu.

6.2.5 Věta *Svaz je distributivní právě tehdy, když neobsahuje*

ani svaz  *ani svaz* 

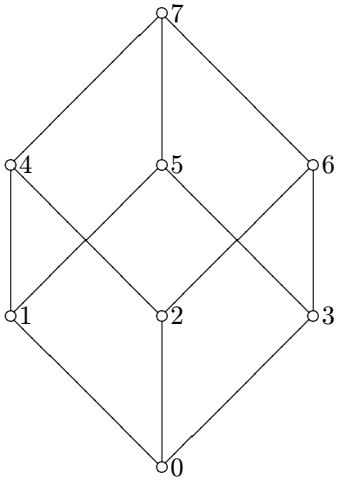
jako podsvazy.

6.2.6 Poznámka Svazu na levé straně ve větě 6.2.5 říkáme *pentagon* a svazu napravo říkáme 3-*diamant*.

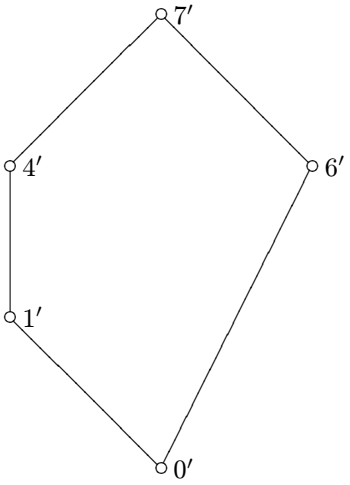
6.2.7 Poznámka Zatím jsme neřekli, co je podsvaz. Z příkladu 5.3.9 připomeňme pojem podmnožiny uzavřené na binární operaci. Pro grupoid $\langle X, \star\rangle$ pak každá podmnožina $A \subseteq X$, která je na operaci $\star$ uzavřená, dává vzniknout grupoidu na množině A, sice grupoidu $\langle A, \star\rangle$. Tomuto grupoidu pak říkáme *podgrupoid* grupoidu $\langle X, \star\rangle$. Ekvivalentně lze pojem podgrupoidu popsat pomocí homomorfismů grupoidů: chceme, aby inkluse byla homomorfismem grupoidů.

U svazů bude situace analogická: budeme vyžadovat, aby daná podmnožina byla uzavřená na operace průseku a spojení.

6.2.8 Příklad Uvažujme o svazu z příkladu 6.1.4. V Hasseho diagramu (6.2) však přejmenujeme vrcholy, abychom si zjednodušili značení.



Z tohoto obrázku nyní „vyjmeme“ vrcholy 0, 1, 4, 6 a 7. Dostaneme tak nový Hasseho diagram (představme si, že jsme vytvořili kopie vrcholů 0, 1, 4, 6 a 7 a označili je $0'$, $1'$, $4'$, $6'$ a $7'$).



Pozor! Objevila se zde zdánlivě nová hrana, sice hrana spojující $0'$ a $6'$. Hrana spojující 0 a 6 sice v původním Hasseho diagramu nebyla, ovšem Hasseho diagram je komprimovaná informace o uspořádání — v reflexivním a transitivním obalu hrana spojující vrcholy 0 a 6 je.

Dostáváme tedy svaz (o kterém víme, že není distributivní). Tento svaz je podsvazem původního svazu právě tehdy, když zobrazení „umazávající čárku“ je homomorfismem svazů.

Protože od homomorfismů svazů chceme pouze respektování průseku a spojení, snadno nahlédneme, že pětiúhelník není podsvazem původního svazu. Platí totiž:

$$4' \wedge 6' = 0'$$

ale

$$4 \wedge 6 = 2$$

Zobrazení „umazávající čárku“ tedy nerespektuje průseky, tj. není homomorfismem svazů a daný pětiúhelník není podsvaz.

6.2.9 Poznámka Víme, že distributivní svazy jsou modely pravdivostních hodnot. Jako takové se používají ve *fuzzy logice*. Více se lze dovědět například v knihách

- ☞ V. Novák, *Fuzzy množiny a jejich aplikace*, SNTL, Praha, 1986
- ☞ M. Navara a P. Olšák, *Základy fuzzy množin*, FEL ČVUT, 2002

nebo v příkladu 6.3.6.

6.3 Booleovy algebry

Distributivní svazy jsou „lepším“ modelem pravdivostních hodnot, než obecné svazy, protože v nich platí distributivní zákony. Chceme-li popsat (klasické) pravdivostní hodnoty v plné síle, musíme specifikovat:

1. Dvě konstanty: $\top$ (čtěte: *top* — to modeluje pravdivostní hodnotu `true`) a $\bot$ (čtěte: *bottom* — to modeluje pravdivostní hodnotu `false`).
2. Unární operaci $(-)'$ (čtěte: *komplement* — to modeluje unární operaci `negation(_)`).

Očekáváme, že tato nová data budou muset splňovat další rovnosti.

Rovnosti pro $\top$ a $\bot$ jsou následující:

$$x \wedge \top = x \quad \text{a} \quad x \wedge \bot = \bot$$

Tyto rovnice nám říkají, že $\top$ je největší prvek a $\bot$ je nejmenší prvek v uspořádání daném větou 6.1.9. To je ve shodě s naší intuicí, že `true` je „nejvíce pravdivá“ a `false` je „nejméně pravdivá“ pravdivostní hodnota.

Dříve než zmíníme rovnosti pro $(-)'$, dokážeme následující tvrzení:

6.3.1 Tvrzení *V distributivním svazu s největším a nejmenším prvkem platí, že pro každé x existuje nejvýše jedno y takové, že platí $x \wedge y = \bot$ a současně $x \vee y = \top$.*

Důkaz. Předpokládejme, že pro dané x existují y_1 a y_2 s vlastnostmi:

$$\begin{array}{ll} x \wedge y_1 = \bot & x \vee y_1 = \top \\ x \wedge y_2 = \bot & x \vee y_2 = \top \end{array}$$

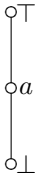
Opakovaným použitím těchto rovností a distributivního zákona dostaneme:

$$\begin{aligned} y_1 &= y_1 \wedge \top = y_1 \wedge (x \vee y_2) = (y_1 \wedge x) \vee (y_1 \wedge y_2) = (y_2 \wedge x) \vee (y_2 \wedge y_1) = \\ &= y_2 \wedge (x \vee y_1) = y_2 \wedge \top = y_2 \end{aligned}$$

a to jsme chtěli ukázat. ■

Předchozí tvrzení říká, že v distributivním svazu s nejmenším a největším prvkem může pro prvek x existovat nejvýše jeden *komplementární prvek* y, tj. takový, že průsek $x \wedge y$ dává „nic“ a spojení $x \vee y$ dává „všechno“. Jsou však i distributivní svazy, kde takový komplementární prvek neexistuje, jak ukazuje následující příklad:

6.3.2 Příklad Svaz daný Hasseho diagramem



je jistě distributivní svaz s největším a nejmenším prvkem (použijte třeba větu 6.2.5).

Je zřejmé, že pro prvek a neexistuje y tak, že platí $a \wedge y = \bot$ a současně $a \vee y = \top$.

Rovnosti pro operaci $(-)'$ jsou

$$x \wedge x' = \bot \quad \text{a} \quad x \vee x' = \top$$

Tyto rovnosti vystihují, že chceme popisovat *klasickou* logiku. Nejprve si uvědomme, že okamžitě z definice plyne rovnost $x'' = x$ (použijte tvrzení 6.3.1). Dále jde z daných axiomů dokázat například platnost de Morganova zákona

$$(x \wedge y)' = x' \vee y'$$

Stačí totiž ukázat, že platí

$$(x' \vee y') \wedge (x \wedge y) = \bot \quad \text{a} \quad (x' \vee y') \vee (x \wedge y) = \top$$

a pak použít tvrzení 6.3.1. Obě rovnosti plynou z několikerého použití distributivních zákonů na levé strany rovností — viz cvičení 6.5.11.

Nyní jsme připraveni napsat specifikaci klasických pravdivostních hodnot. Povšimněte si, že rozšiřujeme specifikaci (6.4) `DISTRIBUTIVE_TRUTH_VALUES(BETTER)` specifikováním dvou nových konstant `true` a `false` a jedné nové unární operace `negation`. Tyto nové operace musí splňovat dodatečné rovnice (řádky 18–21).

```
 1  spec CLASSICAL_TRUTH_VALUES is
 2       sorts: values
 3       operations: _and_:values,values --> values;
 4                   _or_:values,values --> values;
 5                   true: --> values;
 6                   false: --> values;
 7                   negation(_): values --> values;
 8       variables: x,y,z:values
 9       equations: x and y = y and x;
10                  (x and y) and z = x and (y and z);
11                  x and x = x;
12                  x or (x and y) = x;
13                  x or y = y or x;
14                  (x or y) or z = x or (y or z);
15                  x or x = x;
16                  x and (x or y) = x;
17                  x and (y or z) = (x and y) or (x and z);
18                  x and true = x;
19                  x and false = false;
20                  x and negation(x) = false;
21                  x or negation(x) = true
22   endspec
```
(6.5)

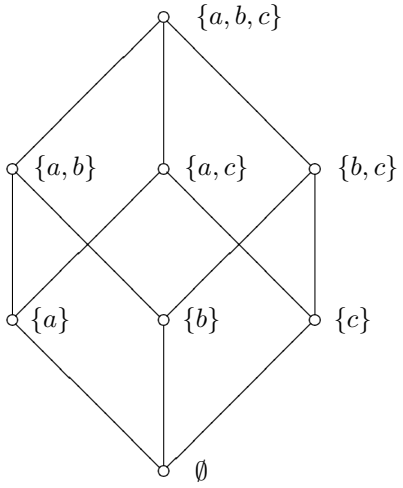
6.3.3 Definice Modelům specifikace `CLASSICAL_TRUTH_VALUES` říkáme *Booleovy*[2] *algebry*.

Booleova algebra je tedy šestice $\langle X, \wedge, \vee, \top, \bot, (-)' \rangle$, kde

[2]Nazváno na počest anglického matematika George Boolea (1815–1864), který ve své knize *An Investigation into the Laws of Thought, on which are founded the Mathematical Theories of Logic and Probabilities* z roku 1854 poprvé studoval logiku jako algebraickou strukturu. Boole zemřel na zápal plic v 49 letech. Promokl, přednášel v mokrých šatech a jeho manželka jej doma uložila do postele a polévala studenou vodou, protože (jak věřila) následek musí být vyléčen příčinou.

1. $\langle X, \wedge, \vee \rangle$ je distributivní svaz (řádky 9–17 specifikace (6.5)).
2. $\top$ je největší a $\bot$ je nejmenší prvek svazu $\langle X, \wedge, \vee \rangle$, tj. pro všechna $x \in X$ platí rovnosti
$$x \wedge \top = x \quad \text{a} \quad x \wedge \bot = \bot$$
(řádky 18 a 19 specifikace (6.5)).
3. $(-)' : X \longrightarrow X$ je unární operace (nazvaná *komplement*), která pro všechna $x \in X$ splňuje rovnosti
$$x \wedge x' = \bot \quad \text{a} \quad x \vee x' = \top$$
(řádky 20 a 21 specifikace (6.5)).

6.3.4 Příklad Hasseho diagram



je příkladem Booleovy algebry: $\bot = \emptyset$, $\top = \{a, b, c\}$ a pro $x \subseteq \{a, b, c\}$ je $x' = \{a, b, c\} \setminus x$.

Následující důležitý výsledek nebudeme dokazovat. Říká, že *konečná Booleova algebra* má pevně danou strukturu, viz také cvičení 6.5.14.

6.3.5 Věta Pro libovolnou množinu M označme jako X množinu všech podmnožin množiny M:
$$X = \{A \mid A \subseteq M\}$$
Potom
$$\langle X, \cap, \cup, M, \emptyset, M \setminus - \rangle$$
je Booleova algebra. Každá *konečná* Booleova algebra je isomorfní výše uvedené Booleově algebře pro nějakou konečnou množinu M a má tudíž 2^n prvků pro nějaké přirozené číslo n.

6.3.6 Příklad V tomto příkladě naznačíme, jak používá svazy a Booleovy algebry fuzzy logika. Přesněji, naznačíme, jak vybudovat teorii fuzzy množin. Budeme si představovat, že chceme studovat podmnožiny jistého (pro jednoduchost konečného) universa
$$U = \{u_1, \ldots, u_n\}$$
Vybrat klasickou podmnožinu A množiny U samozřejmě znamená zvolit ty prvky z U, které do množiny A patří. V teorii fuzzy množin ovšem chceme vybírat prvky U, které do fuzzy podmnožiny patří pouze s nějakou *pravdivostní hodnotou*, která ovšem nemusí nutně být pouze `true` nebo `false`. K tomu se výborně hodí teorie svazů a Booleových algeber.

Ať $\langle X, \wedge, \vee, \top, \bot \rangle$ je svaz (ne nutně distributivní), který má top a bottom. Tomuto svazu budeme říkat *svaz atributů*. Budeme pracovat se zápisy
$$u.x$$

kde u je prvek universa U a x je prvek svazu X. Tento zápis si chceme představit jako prvek u, jehož „existence“ má pravdivostní hodnotu x. Přesněji, budeme studovat množiny

$$A = \{u_1.x_1, \ldots, u_n.x_n\}$$

a tento zápis bude znamenat, že

$$[\![u_1 \in A]\!] = x_1, \ldots, [\![u_n \in A]\!] = x_n$$

kde dvojitými hranatými závorkami značíme pravdivostní hodnotu výrazu uvnitř.

S dvouprvkovým svazem atributů $\top$ a $\bot$ jsme zvyklí pracovat v klasické teorii množin: zápis $[\![u \in A]\!] = \bot$ samozřejmě znamená, že u v A neleží a zápis $[\![u \in A]\!] = \top$ znamená, že u v A leží.

Je zřejmé, že universum U si (pro libovolný svaz atributů) nyní můžeme představit takto:

$$U = \{u_1.\top, \ldots, u_n.\top\}$$

tj. v universu U mají všechny prvky atribut $\top$ (pravdivostní hodnota toho, že u leží v universu, je `true`).

Naopak množina, která nemá žádné prvky (tj. prázdná množina), je množina

$$\emptyset = \{u_1.\bot, \ldots, u_n.\bot\}$$

tj. v prázdné množině $\emptyset$ mají všechny prvky atribut $\bot$ (pravdivostní hodnota toho, že u leží v prázdné množině, je `false`).

Představme si nyní, že jsme zvolili dvě fuzzy podmnožiny

$$A = \{u_1.x_1, \ldots, u_n.x_n\} \quad \text{and} \quad B = \{u_1.y_1, \ldots, u_n.y_n\}$$

Co by měla znamenat například inkluse $A \subseteq B$? Zřejmě to, že atribut x prvku u v A je nanejvýš roven atributu y stejného prvku u v B, neboli platí

$$[\![u_1 \in A]\!] \sqsubseteq [\![u_1 \in B]\!], \ldots, [\![u_n \in A]\!] \sqsubseteq [\![u_n \in B]\!]$$

To přesně znamená, že pro prvek u je „pravdivější“ říci, že leží v množině B. Neboli: z toho, že u leží v A plyne, že u leží v B.

Jak modelovat průnik a sjednocení podmnožin? K tomu využijeme operace $\wedge$ a $\vee$ ze svazu atributů:

$$A \cap B = \{u_1.x_1 \wedge y_1, \ldots, u_n.x_n \wedge y_n\} \quad A \cup B = \{u_1.x_1 \vee y_1, \ldots, u_n.x_n \vee y_n\}$$

Interpretace například průniku je zřejmá: $[\![u \in A \cap B]\!] = [\![u \in A]\!] \wedge [\![u \in B]\!]$ (připomeňme, že $\wedge$ modeluje spojku `and`).

Platí některé zákonitosti, na které jsme zvyklí? Platí například distributivní zákon $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$? Obecně nikoli, k tomu zřejmě potřebujeme *distributivní* svaz atributů (zkuste to dokázat).

Dále je jasné, že pro definici rozdílu množin $A \setminus B$, který se má chovat tak, jak jsme zvyklí, potřebujeme, aby svaz atributů byla Booleova algebra s operací komplementu $(-)'$:

$$[\![u \in A \setminus B]\!] = [\![u \in A]\!] \wedge [\![u \in B]\!]'$$

Pokuste se dokázat, že pak platí všechny zákony pro podmnožiny, na které jste zvyklí. Které rovnosti musíte ověřit?

Předchozí úvahy se opět dají elegantněji popsat v teorii kategorií, klíčovým slovem je *konečný součin svazů* a *Booleových algeber*.[3] Odkazujeme na knihu

☞ M. Barr a C. Wells, *Category Theory for Computing Science*, Prentice Hall, 1990

6.3.7 Poznámka Úvahy předchozího příkladu, kdy svazem atributů je Booleova algebra, vedou k důležité partii matematiky, která studuje *nezávislost* některých tvrzení na axiomech teorie množin. Než budeme moci naznačit, o co jde, zmíníme dobře známý příklad z klasické geometrie.

Řecký geometr Eukleides ve své knize $\Sigma\tau o\iota\chi\epsilon\iota\omega\eta$ (*Základy*) zformuloval pět axiomů, které jsou nyní známy jako axiomy *eukleidovské geometrie*. Jde o následujících pět axiomů:

[3] Anglicky: *finite products*.

(E1) Dva různé body lze spojit právě jednou úsečkou.

(E2) Každou úsečku lze libovolně spojitě prodloužit.

(E3) Pro dva různé body P_1, P_2 existuje právě jedna kružnice se středem v P_1 procházející bodem P_2.

(E4) Každé dva pravé úhly mají stejnou velikost.

(E5) Každým bodem P, který neleží na přímce l, lze proložit právě jednu rovnoběžku s přímkou l.

V historii matematiky bylo učiněno mnoho pokusů dokázat pátý axiom z předchozích čtyř. Tato obsese je snad nejlépe zhmotněna italem Giorolamem Saccherim (1667–1733), který důkazu pátého axiomu z prvních čtyř zasvětil celý svůj život. V roce 1733 zveřejnil dílo *Euclides ab Omni Naevo Vindicatus* (*Eukleides vší poskvrny zbavený*), ve kterém dokazuje pátý axiom z prvních čtyř. Saccheriho důkaz je však špatně.

Eukleidovy axiomy pro geometrii a způsob vyvozování důsledků, který Eukleides v díle *Základy* používá, měly obrovský vliv na středověkou filosofii a teologii. Filosofové byli geometrickou metodou fascinováni a řada klasických prací z filosofie je napsána stylem *definice, věta, důkaz*, například dílo

☞ B. Spinoza, *Etika*, Svoboda, Praha, 1977

Benedicta Spinozy (1632–1677) má plný titul *Etika vyložená způsobem užívaným v geometrii* a geometrickou metodou v pěti oddílech pojednává o Bohu, teorii poznání, psychologii a o svobodné vůli.

Teprve v letech 1826–1829 byl v pracech maďara Jánose Bolyaie (1802–1860) a rusa Nikolaje Ivanoviče Lobačevského (1793–1856) světu představen první model *neeukleidovské*, *hyperbolické* geometrie.[4] V neeukleidovské hyperbolické geometrii pátý Eukleidův axiom neplatí. Přesněji: bodem mimo přímku lze vést dokonce *nekonečně mnoho rovnoběžek*. Přímky v hyperbolické geometrii jsou totiž modelovány hyperbolami. Tak například typický trojúhelník v hyperbolické geometrii vypadá takto:

Tudíž součet vnitřních úhlů *každého* trojúhelníka je v hyperbolické geometrii *menší* než π. Jistě znáte i *sférickou geometrii* (tj. geometrii na sféře), kde je součet vniřních úhlů *každého* trojúhelníka *větší* než π (a bodem mimo přímku nelze vést *žádnou* rovnoběžku). Přímky ve sférické geometrii jsou totiž modelovány *hlavními kružnicemi* na sféře.[5]

Hyperbolická i sférická geometrie (přesněji: kombinace obou) mají velký význam v moderní fyzice, viz například

☞ R. Penrose, *The Road to Reality — A Complete Guide to the Laws of the Universe*, Jonathan Cape, Londýn, 2004

a pro vynikající knihu o neeukleidovských geometriích odkazujeme na knihu

☞ R. v.B. Rucker, *Geometry, Relativity and the Fourth Dimension*, Dover Publications, New York, 1977

která je psána velmi přístupným jazykem.

Pátý axiom o rovnoběžkách je pravděpodobně nejznámější instancí tvrzení *nezávislého na axiomech*. Podobně v teorii množin existuje celá řada tvrzení, která jsou nezávislá na základních axiomech teorie množin. Člověk se tak může rozhodnout, zda pracovat v teorii množin, kde takové tvrzení platí (analogie eukleidovské geometrie); nebo v teorii množin, kde toto tvrzení neplatí (analogie neeukleidovské geometrie). Příkladem takového tvrzení je *axiom výběru*. Nejjednodušší zápis axiomu výběru je tvrzení[6]

(AC) *Kartézský součin libovolného systému neprázdných množin je neprázdný.*

[4] Také se jí říká *Lobačevského* geometrie.

[5] Přísně vzato, tento model *nesplňuje první Eukleidův axiom* (přesvědčete se o tom). To se dá snadno napravit: pracujte na „severní hemisféře“, kde ztotožníte protilehlé body rovníku.

[6] Anglicky *Axiom of Choice*, odtud značení (AC).

Povšimněte si, že toto tvrzení triviálně platí, pokud máme *konečný* systém neprázdných množin. Pokud však chceme mluvit o *libovolných* systémech neprázdných množin, stává se toto tvrzení nezávislým na axiomech a my se můžeme rozhodnout buď pro jeho pravdivost nebo jeho nepravdivost.[7] Ačkoli se tvrzení axiomu výběru zdá intuitivně zřejmé, plynou z něj i tvrzení, která bychom v teorii množin „nejraději neviděli“. Pomocí axiomu výběru tak lze například rozřezat jednotkovou kouli v $\mathbb{R}^3$ na konečně mnoho kusů a tyto kusy lze pak poskládat ve *dvě* jednotkové koule v $\mathbb{R}^3$ (tomu se říká *Banachův-Tarského paradox*).

Technika, kterou se nezávislost množinových tvrzení dokazuje, je technika, kterou jsme začali budovat v příkladu 6.3.6.

6.4 Vícesortový případ

Vícesortový přístup se velmi přirozeně objevuje v aplikacích (viz cvičení 5.7.23). Při specifikaci zásobníků v odstavci 5.1 jsme viděli, že je přirozené využít dvou sort. Připomeňme tu specifikaci:

```
1    spec STACK is
2         sorts: alphabet, stack
3         operations: a1: --> alphabet;
4                     a2: --> alphabet;
5                     a3: --> alphabet;
6                     a4: --> alphabet;
7                     nil: --> stack                                  (6.6)
8                     pop(_): stack --> stack
9                     push(_,_): alphabet,stack --> stack
10        variables: x:alphabet, s:stack
11        equations: pop(nil)=nil;
12                   pop(push(x,s))=s
13   endspec
```

V tomto odstavci podáme jen základní definice, takže získáme základní představu, jak se v kontextu více sort pracuje. Pro detaily odkazujeme na knihu

☞ W. Wechler, *Universal Algebra for Computer Scientists*, Springer-Verlag, Berlin, 1992.

6.4.1 Definice Ať S je neprázdná množina, budeme jí říkat množina *sort*. Kolekci navzájem disjunktních množin X_s, $s \in S$ budeme říkat *S-sortová množina* a budeme ji značit $\langle X_s \mid s \in S\rangle$. A *$S$-sortové zobrazení* z $\langle X_s \mid s \in S\rangle$ do $\langle Y_s \mid s \in S\rangle$ je kolekce zobrazení $f_s : X_s \longrightarrow Y_s$, $s \in S$.

6.4.2 Příklad Ať $S = \{\texttt{alphabet}, \texttt{stack}\}$.

S-sortová množina je jakákoli dvojice disjunktních množin $X_{\texttt{alphabet}}$ a $X_{\texttt{stack}}$.

Pro dvě S-sortové množiny $\langle X_{\texttt{alphabet}}, X_{\texttt{stack}}\rangle$ $\langle Y_{\texttt{alphabet}}, Y_{\texttt{stack}}\rangle$ je S-sortové zobrazení dvojice zobrazení, které „respektují sorty“, tj. dvojice zobrazení

$$f_{\texttt{alphabet}} : X_{\texttt{alphabet}} \longrightarrow Y_{\texttt{alphabet}} \quad f_{\texttt{stack}} : X_{\texttt{stack}} \longrightarrow Y_{\texttt{stack}}$$

Řekneme nyní, co je arita ve vícesortovém případě. Například operace `push` má aritu

```
alphabet,stack --> stack
```

Arita ve vícesortovém případě se tedy sestává ze dvou věcí: *domainu* (v našem příkladu `alphabet,stack`, obecně jde o konečné slovo nad množinou sort) a *kodomainu* (v našem příkladu `stack`, obecně jde o jednu sortu). Řekneme tedy, že arita je uspořádaná dvojice (w, s), kde w je slovo nad S a s je prvek S. Přesněji (připomeňme, že S^* značí množinu všech slov nad S):

[7] Podobnou věc jsme viděli u principu dobrého uspořádání 1.1.15 — tvrzení, že *konečná* neprázdná množina přirozených čísel má nejmenší prvek, je triviálně pravdivé. Pokud se chceme bavit o nejmenších prvcích *libovolných* neprázdných podmnožin přirozených čísel, stává se principem nezávislým na základních axiomech. Pro model přirozených čísel, který není dobře uspořádán, odkazujeme na poznámku 1.1.20.

6.4.3 Definice *Arita* (nad S) je dvojice $(w, s) \in S^* \times S$. *Finitární typ* Ω (nad S) je kolekce navzájem disjunktních množin

$$\Omega_{(w,s)}$$

které obsahují *formální operace* arity (w, s).

Připomeňme, že v odstavci 5.4 jsme zavedli exponenciální značení. Symbolem X^n, $n \geq 1$, jsme označili množinu všech uspořádaných n-tic prvků prvků množiny X a symbolem X^0 jsme označili pevnou jednoprvkovou množinu. Analogicky jsme pro zobrazení $f : X \longrightarrow Y$ označili symbolem $f^n : X^n \longrightarrow Y^n$ v případě $n = 0$ identitu na jednoprvkové množině a v případě $n \geq 1$ jsme tak označili zobrazení, posílající n-tici $(x_1, \ldots, x_n)$ na n-tici $(f(x_1), \ldots, f(x_n))$.

Exponenciální značení rozšíříme na S-sortové množiny $X = \langle X_s \mid s \in S\rangle$. Jako exponenty dovolíme slova nad abecedou S:

1. X^ε je pevná jednoprvková množina.
2. Pro $s \in S$ definujeme X^s jako s-tou komponentu S-sortové množiny X, tj. $X^s = X_s$.
3. Pro slovo $w = s_1 s_2 \ldots s_n \in S^*$ definujeme $X^w = X_{s_1} \times X_{s_2} \times \ldots \times X_{s_n}$.

Pro S-sortové zobrazení $f : X \longrightarrow Y$, kde $X = \langle X_s \mid s \in S\rangle$ a $Y = \langle Y_s \mid s \in S\rangle$, definujeme:

1. $f^\varepsilon : X^\varepsilon \longrightarrow Y^\varepsilon$ je identita.
2. Pro $s \in S$ definujeme $f^s : X^s \longrightarrow Y^s$ jako s-tou komponentu f, tj. $f_s : X_s \longrightarrow X_s$.
3. Pro slovo $w = s_1 s_2 \ldots s_n \in S^*$ definujeme $f^w : X^w \longrightarrow Y^w$ jako zobrazení, které posílá n-tici

$$(x_1, x_2, \ldots, x_n) \in X_{s_1} \times X_{s_2} \times \ldots \times X_{s_n}$$

na n-tici

$$(f_{s_1}(x_1), f_{s_2}(x_2), \ldots, f_{s_n}(x_n)) \in Y_{s_1} \times Y_{s_2} \times \ldots \times Y_{s_n}.$$

6.4.4 Definice *Algebra* $\mathbb{X}$ finitárního typu Ω (nad S) je zadána

1. *Nosnou S-sortovou množinou* $X = \langle X_s \mid s \in S\rangle$.
2. Kolekcí zobrazení $\omega_{\mathbb{X}} : X^w \longrightarrow X_s$, kterým říkáme *interpretace formální operace* ω, pro každé $\omega \in \Omega_{(w,s)}$.

Ať $\mathbb{X}$ a $\mathbb{Y}$ jsou dvě algebry finitárního typu Ω s nosnými S-sortovými množinami $X = \langle X_s \mid s \in S\rangle$ a $Y = \langle Y_s \mid s \in S\rangle$. S-sortovému zobrazení $f : X \longrightarrow Y$ budeme říkat *homomorfismus algeber typu* Ω, pokud následující diagram

$$\begin{array}{ccc} X^w & \xrightarrow{f^w} & Y^w \\ {\scriptstyle \omega_{\mathbb{X}}}\downarrow & & \downarrow{\scriptstyle \omega_{\mathbb{Y}}} \\ X_s & \xrightarrow[f_s]{} & Y_s \end{array}$$

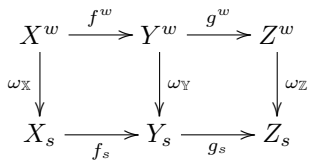
komutuje pro všechna $\omega \in \Omega_{(w,s)}$.

Tento fakt značíme $f : \mathbb{X} \longrightarrow \mathbb{Y}$.

6.4.5 Poznámka Srovnejte důkaz následující věty s druhou variantou důkazu věty 5.3.16. Vidíte sílu teorie kategorií?

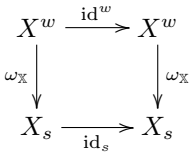
6.4.6 Tvrzení *Ať $f : \mathbb{X} \longrightarrow \mathbb{Y}$ and $g : \mathbb{Y} \longrightarrow \mathbb{Z}$ jsou homomorfismy algeber finitárního typu Ω. Pak složení $g \circ f$ je homomorfismus z $\mathbb{X}$ do $\mathbb{Z}$. Navíc S-sortové identické zobrazení id_X na X je vždy homomorfismus.*

Důkaz. První tvrzení je jednoduché, protože diagram



komutuje pro všechna $\omega \in \Omega_{(w,s)}$.

Abychom dokázali druhé tvrzení, všimněte si, že id^w je vždy identické zobrazení a proto diagram



komutuje pro všechna $\omega \in \Omega_{(w,s)}$. ■

6.4.7 Poznámka Tvrzení 6.4.6 říká, že algebry typu Ω a jejich homomorfismy tvoří kategorii, porovnejte to s poznámkou 5.3.17.

Než budeme moci ve vícesortovém případě definovat rovnice, musíme definovat termy.

6.4.8 Definice Ať V je pevná S-sortová množina a ať Ω je finitární typ nad S. V tomto kontextu budeme množině V říkat S-sortová *množina formálních proměnných*.

Ať $\mathbb{X}$ je algebra typu Ω s nosnou S-sortovou množinou X. *Interpretace formálních proměnných v* X je jakékoli S-sortové zobrazení

$$\rho : V \longrightarrow X$$

S-sortová množina $T_\Omega(V)$ *termů typu* Ω *nad* V je definována induktivně následujícími pravidly:

1. Jestliže $v \in V_s$, pak v je prvek $T_\Omega(V)$ sorty s.

   Jinými slovy: každá formální proměnná je term a sorty jsou při tom respektovány.

2. Jestliže $\omega \in \Omega_{(\varepsilon,s)}$ je konstanta sorty s, pak ω je prvek $T_\Omega(V)$ sorty s.

3. Jestliže $w = s_1 s_2 \dots s_n$ je slovo nad S, $n \geq 1$, a jestliže $\omega \in \Omega_{(w,s)}$ je formální operace arity (w, s) a jestliže $t_1, \dots, t_n$ jsou prvky množiny $T_\Omega(V)$ sort $s_1, s_2, \dots, s_n$, pak $f(t_1, \dots, t_n)$ je prvkek $T_\Omega(V)$ sorty s.

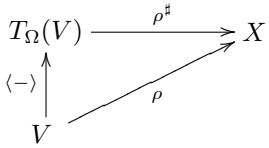
   To znamená: „aplikací“ formální (w, s)-ární operace na n-tici termů „správných“ sort dostáváme term sorty s.

Formální rovnice sorty s *v proměnných z množiny* V je zápis

$$\forall V.\, t_1 \approx t_2$$

kde t_1 a t_2 jsou prvky množiny $T_\Omega(V)$ sorty s.

Stejně jako v jednosortovém případě lze dokázat, že S-sortová množina $T_\Omega(V)$ termů nad V je nosnou množinou algebry $\mathbb{T}_\Omega(V)$ typu Ω. S-sortové zobrazení $\langle - \rangle : V \longrightarrow T_\Omega(V)$ navíc předkládá $\mathbb{T}_\Omega(V)$ jako volnou algebru nad V:

> Pro libovolnou algebru $\mathbb{X}$ s nosnou S-sortovou množinou X a pro libovolné S-sortové zobrazení $\rho : V \longrightarrow X$ existuje jediné S-sortové zobrazení $\rho^\sharp : T_\Omega(V) \longrightarrow X$, pro které následující diagram
>
> 
>
> komutuje. Zobrazení $\rho^\sharp$ je navíc homomorfismus algeber.

6.4.9 Definice Řekneme, že algebra $\mathbb{X}$ typu Ω s nosnou S-sortovou množinou X *splňuje formální rovnici* $\forall V.\, t_1 \approx t_2$, když pro všechna $\rho : V \longrightarrow X$ platí rovnost $\rho^\sharp(t_1) = \rho^\sharp(t_2)$.

Tento fakt značíme $\mathbb{X} \models \forall V.\, t_1 \approx t_2$.

Podobně jako v jednosortovém případě můžeme sestrojit volnou algebru splňující S-sortovou množinu rovnic. To je základ *iniciální sémantiky* rovnicových algebraických specifikací.

6.4.10 Definice Finitární typ Ω nad konečnou množinou sort S je *silně finitární*, když obsahuje pouze konečně mnoho neprázdných konečných množin $\Omega_{(w,s)}$.

6.4.11 Definice *Rovnicová algebraická specifikace* nad S je dvojice $\mathcal{S} = (\Omega, E)$, kde Ω je silně finitární typ nad S a E je S-sortová množina formálních rovnic taková, že E obsahuje pouze konečně mnoho neprázdných konečných množin.

Zamýšlený model specifikace $\mathcal{S} = (\Omega, E)$ je volná algebra nad prázdnou S-sortovou množinou proměnných, která splňuje všechny rovnice z S-sortové množiny E. Pro podrobnosti odkazujeme na knihu

☞ W. Wechler, *Universal Algebra for Computer Scientists*, Springer-Verlag, Berlin, 1992

6.4.12 Poznámka Určitě jste si všimli podobnosti mezi jednosortovým a vícesortovým případem. Jednotící pohled dává opět teorie kategorií:

1. V jednosortovém případě pracujeme nad kategorií Set všech množin a zobrazení. To znamená, že nosné objekty našich algeber jsou množiny a že homomorfismy jsou neseny zobrazeními.
2. V obecném S-sortovém případě pracujeme nad kategorií Set^S všech S-sortových množin a S-sortových zobrazení. To znamená, že nosné objekty našich algeber jsou S-sortové množiny a že homomorfismy jsou neseny S-sortovými zobrazeními.

V jiných oblastech computer science (například v sémantice programovacích jazyků) potřebujeme, aby algebry byly neseny (například) určitými posety nebo metrickými prostory. To si vyžaduje změnu kategorie, nad kterou pracujeme, viz například knihy

☞ M. Barr a C. Wells, *Category Theory for Computing Science*, Prentice Hall, 1990

☞ R. D. Tennent, *Semantics of Programming Languages*, C.A.R. Hoare Series, Prentice Hall, London 1991

6.5 Cvičení

6.5.1 Cvičení Vysvětlete, proč největší společný dělitel přirozených čísel a, b lze chápat jako infimum v uspořádání podle dělitelnosti. Návod: porovnejte definice 2.1.7 a 6.1.2.

6.5.2 Cvičení Existuje uspořádaná množina, kde každá dvojice *různých* prvků má infimum, ale žádná dvojice *různých* prvků nemá supremum?

6.5.3 Cvičení Ať $\langle X, \sqsubseteq \rangle$ je poset. Dokažte, že $\langle X, \sqsubseteq^{op} \rangle$ je opět poset, kde

$$x \sqsubseteq^{op} y \quad \text{právě tehdy, když} \quad y \sqsubseteq x$$

Jaký je vztah Hasseho diagramů posetů $\langle X, \sqsubseteq \rangle$ a $\langle X, \sqsubseteq^{op} \rangle$? Jak se počítají infima a suprema v posetu $\langle X, \sqsubseteq^{op} \rangle$?

6.5.4 Cvičení Ať $\langle X, \wedge, \vee \rangle$ je svaz. Definujte binární operace $\sqcap$ a $\sqcup$ následujícím způsobem:

$$\begin{aligned} x \sqcap y &= x \vee y \\ x \sqcup y &= x \wedge y \end{aligned}$$

Dokažte, že $\langle X, \sqcap, \sqcup \rangle$ je opět svaz. Jaký je vztah Hasseho diagramů svazů $\langle X, \wedge, \vee \rangle$ a $\langle X, \sqcap, \sqcup \rangle$?

6.5.5 Cvičení Definujte přesně pojmy: *homomorfismus svazů* a *podsvaz*.

6.5.6 Cvičení Označte jako X množinu všech kladných dělitelů čísla 40. Na množině X definujte binární relaci $\sqsubseteq$ takto:

$$x \sqsubseteq y \quad \text{právě tehdy, když číslo } y \text{ je dělitelné číslem } x$$

Ukažte, že $\langle X, \sqsubseteq \rangle$ je svaz (ve smyslu věty 6.1.8). Jak lze popsat operace průseku a spojení v tomto svazu? Jde o distributivní svaz? Jde o Booleovu algebru?

6.5.7 Cvičení Ukažte, že v libovolném svazu $\langle X, \wedge, \vee \rangle$ jsou následující podmínky ekvivalentní:

1. Každá podmnožina množiny X je podsvaz svazu $\langle X, \wedge, \vee \rangle$.
2. Uspořádání $\sqsubseteq$ na množině X (viz věta 6.1.9) je *lineární*, tj. pro libovolné prvky $x, y \in X$ platí $x \sqsubseteq y$ nebo $y \sqsubseteq x$.

6.5.8 Cvičení Dokažte, že ve svazu $\langle X, \wedge, \vee \rangle$ pro všechna x, y, z platí *distributivní nerovnost*

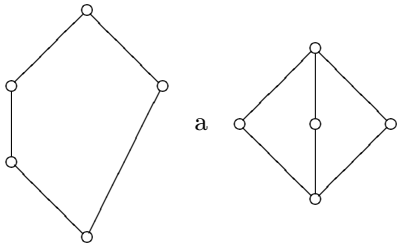
$$(x \wedge y) \vee (x \wedge z) \sqsubseteq x \wedge (y \vee z)$$

kde $\sqsubseteq$ je uspořádání z věty 6.1.9.

Tím je dokončen důkaz faktu, že každý svaz je „téměř distributivní“, viz příklad 6.1.11.

6.5.9 Cvičení Dokažte, že 3-diamant (viz větu 6.2.5) není distributivní svaz.

6.5.10 Cvičení V Hasseho diagramu z příkladu 6.2.8 nalezněte všechny „podobrázky“ tvaru



a dokažte, že nejde o podsvazy.

6.5.11 Cvičení Ukažte, že v každé Booleově algebře platí *de Morganovy zákony* $(x \wedge y)' = x' \vee y'$ a $(x \vee y)' = x' \wedge y'$.

6.5.12 Cvičení V libovolné Booleově algebře $\langle X, \wedge, \vee, \top, \bot, (-)' \rangle$ definujte binární operaci $\Rightarrow$ následovně:

$$(x \Rightarrow y) = (x' \vee y)$$

Dokažte, že v každé Booleově algebře platí *věta o dedukci*: pro libovolné prvky x, y, z

$$z \sqsubseteq (x \Rightarrow y) \quad \text{právě tehdy, když} \quad (z \wedge x) \sqsubseteq y$$

Proč se tomuto tvrzení říká věta o dedukci?

6.5.13 Cvičení Ukažte, že neexistuje Booleova algebra, která by měla přesně tři prvky. Návod: použijte příklad 6.3.2.

6.5.14 Cvičení Pro libovolnou množinu M označme jako X množinu všech podmnožin množiny M:

$$X = \{A \mid A \subseteq M\}$$

Potom $\langle X, \cap, \cup, M, \emptyset, M \setminus - \rangle$ je Booleova algebra. Návod: zobecněte úvahy z příkladu 6.1.4.

Promyslete, jak vypadají Hasseho diagramy těchto Booleových algeber pro konečné množiny M. Pro množiny $M = \emptyset$, $M = \{a\}$, $M = \{a, b\}$ a $M = \{a, b, c\}$ tyto Hasseho diagramy nakreslete. Vidíte nějakou souvislost? Jak byste nazvali Hasseho diagram pro množinu $M = \{a, b, c, d\}$?

6.5.15 Cvičení Na Booleově algebře $\langle X, \wedge, \vee, \top, \bot, (-)' \rangle$ definujte binární operaci $\oplus$ takto:

$$x \oplus y = (x \wedge y') \vee (x' \wedge y)$$

Ukažte, že rovnost $x \oplus y = \bot$ platí právě tehdy, když platí $x = y$.

Co znamená operace $\oplus$ v Booleových algebrách ze cvičení 6.5.14?

6.5.16 Cvičení Označte jako F množinu všech formulí výrokové logiky nad množinou At atomických formulí. Z přednášky z logiky si připomeňte, že relace $\models\!\models$ sémantické ekvivalence je relace ekvivalence.

Ukažte, že na faktorové množině $L = F/\models\!\models$ vzniká Booleova algebra. Této Booleově algebře se říká *Lindenbaumova algebra*.

6.5.17 Cvičení Vypište explicitně, co znamená pojem *homomorfismus Booleových algeber*.

6.5.18 Cvičení Dejte iniciální sémantiku specifikace (6.1) `BASIC_TRUTH_VALUES`. (Návod: lze definovat na prázdné množině strukturu svazu?)

6.5.19 Cvičení Uvažujte o jednosortovém případu, tj. $S = \{s\}$. Řekněte, co jsou S-sortové množiny a S-sortová zobrazení. Co je arita nad S? Je exponenciální značení pro slova nad S v rozporu s exponenciálním značením z odstavce 5.4?

6.5.20 Cvičení Dejte iniciální sémantiku specifikace (6.6) `STACK`.

Revize kapitoly

Dozvěděli jsme se:

- ✔ Svazy jsou modely základního chování spojek `and` a `or`. Tyto modely si můžeme představit jako posety, kde existují infima všech dvojic (ta modelují `and`) a suprema všech dvojic (ta modelují `or`).
- ✔ Požadujeme-li pro `and` a `or` navíc platnost distributivních zákonů, musíme pracovat v distributivním svazu. Hasseho diagram distributivního svazu nesmí obsahovat zakázané podsvazy 3-diamant a pentagon.
- ✔ Chceme-li modelovat klasickou logiku, musíme pracovat v Booleově algebře. Konečná Booleova algebra musí mít 2^n prvků (pravdivostních hodnot).
- ✔ Použijeme-li kategoriální techniku, pak se pojmy vícesortové teorie neliší od jednosortových.

Pro přípravu na zkoušku zkuste zodpovědět následující otázky:

- ✔ Jakou logiku popisuje tříprvkový svaz?
- ✔ Konstrukci `if _ then _ else _ fi` programovacího jazyka považujte za operaci. Popište její (vícesortovou) aritu a dejte příklad jedné rovnice, kterou by tato operace měla splňovat. U této rovnice popište její sortu.
- ✔ U všech rovnic specifikace `STACK` popište jejich sortu a dejte iniciální sémantiku této specifikace.
- ✔ Navrhněte rovnicovou specifikaci datového typu `QUEUE` (fronta, tj. paradigma *first-in-first-out*). Daří se rovnicemi zachytit vše, co byste si přáli?

Doplňující literatura

Seznam literatury je stejný jako v kapitole 5. Lze jej doplnit o knihu

- ☞ B. A. Davey a H. A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, 1990

kde je kromě svazů a Booelových algeber vyložena i aplikace svazů v *teorii konceptů*.

Příloha A

Aplikace řetězových zlomků

> Matematik může připomínat oděvního návrháře, který se vůbec nestará o bytosti, jimž mají jeho oděvy padnout. Jeho umění sice vzniklo na základě potřeby tyto bytosti oblékat, ale to už bylo dávno. Dnes se příležitostně vyskytne někdo, komu šat padne, jako by byl pro něj ušit. Překvapení a radost pak nebere konce.
>
> Tobias Danzig

Řetězové zlomky tvoří klasické téma teorie čísel. V této kapitole ukážeme jednu jejich standardní aplikaci: výpočet periody zatmění Slunce a Měsíce. V odstavci A.3 ukážeme ale také aplikaci zcela moderní: využití řetězových zlomků pro *útok na protokol RSA*.

A.1 Řetězové zlomky

V tomto odstavci ukážeme, jak použít Eukleidův algoritmus k výpočtu řetězového zlomku racionálního čísla.

Porovnejte následující tvrzení s definicí 1.2.2.

A.1.1 Tvrzení *Každý zlomek* $\frac{a}{b}$, *kde* a, b *jsou kladná přirozená čísla, lze zapsat ve tvaru*

$$\frac{a}{b} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \ldots \cfrac{1}{q_n}}} \tag{A.1}$$

kde $q_1, \ldots, q_n$ *jsou přirozená čísla,* $q_n \neq 0$.

Přenecháme (jednoduchý) důkaz jako cvičení a ukážeme na příkladu, jak požadovaný zápis získat pomocí Eukleidova algoritmu.

A.1.2 Příklad Vyjádříme v požadované formě zlomek $\frac{221}{42}$. Spustíme (obyčejný) Eukleidův algoritmus:

$$\begin{aligned} 221 &= 5 \cdot 42 + 11 \\ 42 &= 3 \cdot 11 + 9 \\ 11 &= 1 \cdot 9 + 3 \\ 9 &= 3 \cdot 3 + 0 \end{aligned}$$

Tvrdíme, že *posloupnost kvocientů* 5, 3, 1, 3 je hledaná posloupnost $q_1, \ldots, q_n$. Budeme totiž zlomek postupně přepisovat pomocí jednotlivých rovnic z Eukleidova algoritmu:

$$\begin{aligned}
\frac{221}{42} &= 5 + \frac{11}{42} && \text{(první rovnice v Eukleidově algoritmu)}\\
&= 5 + \frac{11}{3\cdot 11 + 9} && \text{(druhá rovnice v Eukleidově algoritmu)}\\
&= 5 + \cfrac{1}{3 + \frac{9}{11}} && \text{(krácení číslem 11)}\\
&= 5 + \cfrac{1}{3 + \cfrac{9}{1\cdot 9 + 3}} && \text{(třetí rovnice v Eukleidově algoritmu)}\\
&= 5 + \cfrac{1}{3 + \cfrac{1}{1 + \frac{3}{9}}} && \text{(krácení číslem 9)}\\
&= 5 + \cfrac{1}{3 + \cfrac{1}{1 + \frac{1}{3}}} && \text{(poslední rovnice v Eukleidově algoritmu)}
\end{aligned}$$

Převedeme na požadovaný tvar zlomek $\frac{4}{235}$. V tomto případě si uvědomme, že musí nutně platit $q_1 = 0$ a další hodnoty $q_2, \ldots, q_n$ nalezneme jako kvocienty v Eukleidově algoritmu pro zlomek $\frac{235}{4}$:

$$\begin{aligned}
235 &= 58\cdot 4 + 3\\
4 &= 1\cdot 3 + 1\\
3 &= 3\cdot 1 + 0
\end{aligned}$$

a proto platí

$$\frac{4}{235} = 0 + \cfrac{1}{58 + \cfrac{1}{1 + \frac{1}{3}}}$$

A.1.3 Definice Zlomku na pravé straně rovnosti (A.1) říkáme *řetězový zlomek* čísla $\frac{a}{b}$ a budeme jej značit $[q_1; q_2, \ldots, q_n]$.

Obráceně, je-li zadán řetězový zlomek na pravé straně rovnosti (A.1), je snadné najít příslušná přirozená čísla a, b: spustíme Eukleidův algoritmus „pozpátku“.

A.1.4 Příklad Pro řetězový zlomek $[6; 8, 7, 5]$ nalezněte příslušná a a b.

Zápis $[6; 8, 7, 5]$ je zkratkou za zlomek

$$6 + \cfrac{1}{8 + \cfrac{1}{7 + \frac{1}{5}}}$$

a tento zlomek zjednodušíme zpětným chodem Eukleidova algoritmu. Stačí totiž pozpátku doplňovat do rovnic

Eukleidova algoritmu, kde posloupnost 6, 8, 7, 5 musí tvořit kvocienty (vyplňujeme odspodu):

$$\begin{aligned} 1\,794 &= 6\cdot 293 + 36 \\ 293 &= 8\cdot 36 + 5 \\ 36 &= 7\cdot 5 + 1 \\ 5 &= 5\cdot 1 + 0 \end{aligned}$$

Proto má daný řetězový zlomek hodnotu $\frac{1\,794}{293}$. Povšimněme si, že tento způsob nalezení čísel a, b vede nutně ke dvojici *nesoudělných* čísel a, b, protože jsme poslední rovnici v Eukleidově algoritmu volili ve tvaru $5 = 5\cdot 1 + 0$. Kdybychom začali (například) rovnicí $10 = 5\cdot 2 + 0$, skončili bychom s dvojicí čísel a, b, pro které platí $\gcd(a,b) = 2$.

A.1.5 Poznámka I když jsme řetězové zlomky definovali pouze pro racionální čísla, nic zřejmě nebrání tomu, definovat řetězový zlomek pro libovolné kladné *reálné* číslo. Lze dokázat, že takový zlomek (obecně nekonečný) vždy existuje. Toto je například řetězový zlomek čísla $e = 2,718\,281\,828\,459\,045\,235\,36\ldots$ (základ přirozených logaritmů):

$$e = 2 + \cfrac{1}{1 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{1 + \ldots}}}}$$

nebo (budeme-li psát pouze posloupnost q_1, q_2, $\ldots$)

$$e = [2; 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, 1, 12, 1, 1, 14, \ldots]$$

Není to fascinující? Číslo e je *transcendentní* (viz cvičení 4.7.18), přesto má jeho řetězový zlomek velmi přehlednou strukturu! Zkuste najít řetězový zlomek pro číslo $\sqrt{2}$.

Řetězový zlomek reálného čísla ovšem obecně přehlednou strukturu mít nemusí, například

$$\pi = [3; 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, \ldots]$$

Zajímavě se o strukturách nejrůznějších nekonečných posloupností píše v knize

☞ D. R. Hofstadter, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, Penguin Books, 1998

a více se o řetězových zlomcích lze dočíst například v knize

☞ H. Davenport, *The Higher Arithmetic*, Cambridge University Press, 1999

A.2 Zatmění Slunce a Měsíce

Následující příklad je v knize

☞ O. Hlad a J. Pavlousek, *Přehled astronomie*, SNTL, Praha, 1984

pouze naznačen, my jej spočteme podrobněji.

A.2.1 Příklad Dostanou-li se Země, Měsíc a Slunce do jedné přímky, nastává zatmění Slunce nebo zatmění Měsíce. Aby nastalo zatmění, musí nastat současně tyto dvě podmínky:

1. Měsíc musí být v novu (zatmění Slunce) nebo v úplňku (zatmění Měsíce).
   Době mezi dvěma stejnými fázemi Měsíce se říká *synodický měsíc* a trvá $s = 29,530\,588\,18$ dne.
2. Měsíc musí být v uzlu své dráhy (tj. v průsečíku své dráhy s rovinou ekliptiky).
   Době oběhu Měsíce od uzlu k témuž uzlu se říká *drakonický měsíc* a trvá $d = 27,212\,219\,97$ dne.

Spočítáme nyní periodičnost obou dějů, tj. hledáme přirozená čísla a, b tak, aby platilo

$$\frac{s}{d} = \frac{29,530\,588\,18}{27,212\,219\,97} \doteq \frac{a}{b}$$

Samozřejmě, je snadné najít čísla a, b tak, aby poslední rovnost platila *přesně* — stačí zlomek $\frac{s}{d}$ rozšířit číslem 10^8:

$$\frac{s}{d} = \frac{2\,953\,058\,818}{2\,721\,221\,997}$$

neboli

$$2\,721\,221\,997 \cdot s = 2\,953\,058\,818 \cdot d$$

To znamená, že po 80 359 286 140 dnech, tj. po 22 001 173 letech a 175 dnech se zatmění budou opakovat (pokud se za tu dobu nezmění hodnoty synodického a drakonického měsíce).

My se budeme snažit zlomek $\frac{s}{d}$ *aproximovat* zlomkem tvaru $\frac{a}{b}$ a využijeme k tomu řetězové zlomky:

$$\begin{aligned}
2\,953\,058\,818 &= 1 \cdot 2\,721\,221\,997 + 231\,836\,821\\
2\,721\,221\,997 &= 11 \cdot 231\,836\,821 + 171\,016\,966\\
231\,836\,821 &= 1 \cdot 171\,016\,966 + 60\,819\,855\\
171\,016\,966 &= 2 \cdot 60\,819\,855 + 49\,377\,256\\
60\,819\,855 &= 1 \cdot 49\,377\,256 + 11\,442\,599\\
49\,377\,256 &= 4 \cdot 11\,442\,599 + 3\,606\,860\\
11\,442\,599 &= 3 \cdot 3\,606\,860 + 622\,019\\
3\,606\,860 &= 5 \cdot 622\,019 + 496\,765\\
622\,019 &= 1 \cdot 496\,765 + 125\,254\\
496\,765 &= 3 \cdot 125\,254 + 121\,003\\
125\,254 &= 1 \cdot 121\,003 + 4\,251\\
121\,003 &= 28 \cdot 4\,251 + 1\,975\\
4\,251 &= 2 \cdot 1\,975 + 301\\
1\,975 &= 6 \cdot 301 + 169\\
301 &= 1 \cdot 169 + 132\\
169 &= 1 \cdot 132 + 37\\
132 &= 3 \cdot 37 + 21\\
37 &= 1 \cdot 21 + 16\\
21 &= 1 \cdot 16 + 5\\
16 &= 3 \cdot 5 + 1\\
5 &= 5 \cdot 1 + 0
\end{aligned}$$

Získali jsme tedy řetězový zlomek

$$\frac{s}{d} = [1; 11, 1, 2, 1, 4, 3, 5, 1, 3, 1, 28, 2, 6, 1, 1, 3, 1, 1, 3, 5]$$

Budeme nyní podíl $\frac{s}{d}$ *aproximovat* zlomky, které z onoho řetězového zlomku dostáváme. Například s použitím prvních dvou členů platí přibližná rovnost

$$\frac{s}{d} \doteq [1; 11] = 1 + \frac{1}{11} = \frac{12}{11}$$

s chybou $1,7\,102$ dne. To znamená, že zhruba po 11 synodických měsících, tj. po cca $324,8$ dnech bude Měsíc opět v uzlu a v novu nebo úplňku (s poměrně velikou chybou).

Chceme-li chybu zmenšit, musíme k aproximaci použít delší řetězový zlomek. Nahradíme-li řetězovým zlomkem o šesti členech, dostaneme z Eukleidova algoritmu

$$\begin{aligned} 242 &= 1\cdot 233+19\\ 223 &= 11\cdot 19+14\\ 19 &= 1\cdot 14+5\\ 14 &= 2\cdot 5+4\\ 5 &= 1\cdot 4+1\\ 4 &= 4\cdot 1+0 \end{aligned}$$

aproximaci

$$\frac{s}{d}\doteq[1;11,1,2,1,4]=\frac{242}{233}$$

s chybou $0,0361$ dne, tj. po 18 letech a $10\frac{1}{3}$ dnech se zatmění opakují (třetina dne posune zatmění o 120 stupňů zeměpisné délky). Tato perioda zatmění byla známa chaldejským astronomům již řadu století před naším letopočtem pod názvem *Saros*. Viz také cvičení A.4.3.

A.3 Wienerův útok na RSA

Myšlenku aproximace zlomku řetězovými zlomky mistrně využívá útok M. Wienera, popsaný v článku

☞ M. J. Wiener, Cryptanalysis of Short RSA Exponents, *IEEE Trans. Inform. Theory* 36 (1990), 553–558

Ukazuje se, že soukromý exponent d účastníka RSA by neměl být příliš malý v porovnání s modulem RSA.

A.3.1 Definice Ať $\alpha=[q_1;q_2,q_3,\ldots]$ je (obecně nekonečný) řetězový zlomek kladného reálného čísla α. Posloupnost racionálních čísel $\alpha_i=[q_1;q_2,q_3,\ldots,q_i]$, $i\geq 1$, nazveme *posloupností konvergent čísla* α a číslo α_i nazveme *i-tou konvergentou čísla* α.

A.3.2 Příklad V poznámce A.1.5 jsme uvedli nekonečný řetězový zlomek

$$\pi=[3;7,15,1,292,1,1,1,2,1,3,\ldots]$$

Spočteme první tři konvergenty čísla π

$$\pi_1=[3]\qquad \pi_2=[3;7]\qquad \pi_3=[3;7,15]$$

metodou příkladu A.1.4. Dostáváme[1]

$$\pi_1=3\qquad \pi_2=\frac{22}{7}\qquad \pi_3=\frac{333}{106}$$

Z názvu je patrné, že konvergenty kladného reálného čísla α by měly být aproximacemi čísla α. Přesný výsledek dává následující věta (která má elementární, ale velmi dlouhý důkaz).

A.3.3 Věta *Posloupnost konvergent nekonečného řetězového zlomku má vždy limitu. Jde-li o řetězový zlomek kladného reálného čísla α, pak limita posloupnosti konvergent je toto číslo α.*

[1] Konvergenta $\pi_1=3$ je zmíněna jako aproximace čísla π v Bibli (První Královská, 7, verš 23): *Odlil také moře o průměru deseti loket, okrouhlé, pět loket vysoké; dalo se obepnout měřící šňůrou dlouhou třicet loket.* Samozřejmě: konvergenta π_1 je velmi nepřesná aproximace čísla π. Druhá konvergenta $\pi_2=\frac{22}{7}$ byla jako přiblížení čísla π známa starořeckým matematikům. Více se o historii čísla π můžete dozvědět v zajímavé knize P. Beckmann, *A History of Pi*, St. Martin's Press, New York, 1976. Bizarní je text písně π z alba *Aerial* (2005) anglické zpěvačky Kate Bush, ve kterém je prvních 78 cifer decimálního rozvoje čísla π správně, pak je 22 cifer špatně, a nakonec opět 37 cifer správně.

Konvergenty řetězového zlomku jsou tedy dobrými aproximacemi daného kladného reálného čísla. Platí i obrácená věta, jejíž důkaz opět neuvedeme:

A.3.4 Věta *Ať α je kladné reálné číslo. Jestliže pro nesoudělná přirozená čísla a a b platí nerovnost*

$$\left|\alpha - \frac{a}{b}\right| \leq \frac{1}{2b^2}$$

pak je zlomek $\frac{a}{b}$ nějaká konvergenta čísla α.

Než budeme moci popsat Wienerův útok na RSA, musíme provést jemnou analýzu modulu n protokolu RSA. Budeme jako vždy přepokládat, že platí $n = p \cdot q$, kde $p < q$. Pak zřejmě existuje nejmenší přirozené číslo c takové, že platí nerovnosti

$$p < q < cp$$

Tato analýza dovolí zformulovat technické tvrzení, které je základem Wienerova útoku:

A.3.5 Tvrzení *Označme jako c nejmenší přirozené číslo takové, že pro dvě prvočísla p, q platí nerovnost $p < q < cq$.*

Ať (n, e) a (n, d) jsou veřejný a soukromý klíč jednoho účastníka protokolu RSA s modulem $n = pq$. Označme jako k přirozené číslo, které splňuje rovnost $ed - k\varphi(n) = 1$ (toto přirozené číslo k existuje díky definici čísel e a d).

Pak platí následující dvě tvrzení:

1. *Platí $k < d$ a $\gcd(d, k) = 1$.*
2. *Jestliže platí nerovnost*

$$d \leq \frac{1}{c+1}\sqrt[4]{n}$$

platí i nerovnost

$$\left|\frac{e}{n} - \frac{k}{d}\right| < \frac{1}{2d^2}$$

Důkaz. Nejprve ukážeme, že platí $\gcd(d, k) = 1$. Z definice čísla k totiž plyne rovnost

$$ed - k\varphi(n) = 1$$

a proto musí být číslo 1 dělitelné číslem $\gcd(d, k)$. To není možné jinak, než že platí $\gcd(d, k) = 1$.

Kdyby nerovnost $k < d$ neplatila, musí platit nerovnost $d < k$, protože případ $k = d$ vztah $\gcd(d, k) = 1$ vylučuje (předpokládáme samozřejmě, že d, jakožto soukromý exponent, není rovno 1). Protože platí $e < \varphi(n)$, máme nerovnost

$$1 = ed - k\varphi(n) < \varphi(n)k - k\varphi(n) = 0$$

což je spor: musí platit $k < d$.

Důkaz druhého tvrzení: Nejprve dokážeme nerovnost

$$n - \varphi(n) < (c+1)\sqrt{n} \tag{A.2}$$

kterou dostaneme z toho, že platí nerovnosti $p < \sqrt{n}$ a $q < cp$:

$$n - \varphi(n) = p \cdot q - (p-1) \cdot (q-1) = p + q - 1 < p + q < p + cp < \sqrt{n} + c\sqrt{n} = (c+1)\sqrt{n}$$

Jako další nerovnost dokážeme

$$\left|\frac{e}{n} - \frac{k}{d}\right| < \frac{k(c+1)\sqrt{n}}{nd} \tag{A.3}$$

Upravujeme levou část nerovnosti (A.3) a využijme při tom rovnost $ed - k\varphi(n) = 1$ a nerovnost (A.2):

$$\begin{aligned}\left|\frac{e}{n} - \frac{k}{d}\right| &= \left|\frac{ed - kn}{nd}\right| \\ &= \left|\frac{ed - k\varphi(n) - kn + k\varphi(n)}{nd}\right| \\ &= \left|\frac{1 - k(n - \varphi(n))}{nd}\right| \\ &= \frac{k(n - \varphi(n)) - 1}{nd} \\ &< \frac{k(n - \varphi(n))}{nd} \\ &< \frac{k(c+1)\sqrt{n}}{nd}\end{aligned}$$

čímž je nerovnost (A.3) dokázána.

Pokud ukážeme platnost nerovnosti

$$\frac{k(c+1)\sqrt{n}}{nd} \leq \frac{1}{2d^2} \tag{A.4}$$

bude důkaz hotov: spojíme dohromady nerovnosti (A.3) a (A.4).

Podle předpokladu platí nerovnost

$$d \leq \frac{1}{c+1}\sqrt[4]{n} \tag{A.5}$$

a protože platí $k < d$ (viz první část důkazu), platí i nerovnosti

$$\frac{k(c+1)\sqrt{n}}{nd} < \frac{d(c+1)\sqrt{n}}{nd} \leq \frac{\frac{\sqrt[4]{n}}{c+1}(c+1)\sqrt{n}}{nd} = \frac{1}{d\sqrt[4]{n}} < \frac{1}{2d^2}$$

kde jsme pro poslední nerovnost využili faktu

$$d < \frac{\sqrt[4]{n}}{c+1} < \frac{\sqrt[4]{n}}{2}$$

protože platí $c \geq 2$.

Ukázali jsme, že nerovnost (A.4) platí, a tím je důkaz ukončen. ■

Tvrzení A.3.5 nám říká, že pokud zvolíme soukromý exponent d tak, že je splněna nerovnost

$$d \leq \frac{1}{c+1}\sqrt[4]{n}$$

pak zlomek $\frac{k}{d}$ je nějaká konvergenta zlomku $\frac{e}{n}$. Volba malého soukromého exponentu tak není odolná na následující útok:

A.3.6 Wienerův útok na RSA Eve spočítá konvergenty $\left(\frac{e}{n}\right)_i$ zlomku $\frac{e}{n}$. Jmenovatele i-té konvergenty označí jako d_i a každého z nich testuje na rovnost

$$(z^e)^{d_i} = z \quad \text{v } \mathbb{Z}_n$$

pro náhodné z. Pokud rovnost pro d_i platí, je d_i soukromý exponent d.

A.3.7 Poznámka Namísto testování náhodné zprávy se může Eve při Wienerově útoku pokusit pomocí hodnot d_i a e faktorizovat modul n efektivním algoritmem z důkazu lemmatu 3.6.2.

Wienerův útok ukážeme na školním příkladu.

A.3.8 Příklad Předpokládejme, že známe veřejný klíč $(n, e) = (574\,339, 52\,075)$ účastníka protokolu RSA. Spočítáme řetězový zlomek $\frac{e}{n} = \frac{52\,075}{574\,339}$:

$$\frac{e}{n} = [0; 10, 1, 52\,074]$$

a jeho jednotlivé (netriviální) konvergenty:

$$[0; 10] = \frac{1}{10} \qquad [0; 10, 1] = \frac{1}{11}$$

Máme dva kandidáty na soukromý exponent: $d_1 = 10$ a $d_2 = 11$. Zvolme náhodnou zprávu z, například $z = 3$. Potom

$$(z^e)^{d_1} = (3^{52\,075})^{10} = 77\,547 \quad \text{v } \mathbb{Z}_{574\,339}$$

takže d_1 nemůže být soukromý exponent. Vyzkoušíme d_2:

$$(z^e)^{d_2} = (3^{52\,075})^{11} = 1 \quad \text{v } \mathbb{Z}_{574\,339}$$

takže hledaný soukromý exponent je $d_2 = 11$.

A.4 Cvičení

A.4.1 Cvičení Nalezněte řetězové zlomky následujících racionálních čísel:

$$\frac{18}{13} \qquad \frac{33}{17} \qquad \frac{18}{29} \qquad \frac{187}{136}$$

A.4.2 Cvičení Pro řetězové zlomky

$$[2; 12] \qquad [0; 5, 7] \qquad [3; 13, 5, 7]$$

nalezněte příslušné zlomky (s nesoudělným čitatelem a jmenovatelem).

A.4.3 Cvičení Trojnásobek periody zatmění Slunce a Měsíce *Saros* (viz příklad A.2.1) nese název *Exeligmos*, najděte jej pomocí aproximací z řetězového zlomku

$$\frac{s}{d} = [1; 11, 1, 2, 1, 4, 3, 5, 1, 3, 1, 28, 2, 6, 1, 1, 3, 1, 1, 3, 5]$$

pro synodický a drakonický měsíc.

A.4.4 Cvičení Nalezněte racionální čísla odpovídající řetězovým zlomkům

$$z_2 = [1; 1] \qquad z_3 = [1; 1, 1] \qquad z_4 = [1; 1, 1, 1]$$

Definujte posloupnost řetězových zlomků z_n, $n \geq 2$, zřejmým způsobem. Dovedete sestavit rekurentní rovnici pro tuto posloupnost racionálních čísel? Dovedete odhadnout limitu této posloupnosti? Svůj odhad dokažte.

A.4.5 Cvičení Spočítejte čtvrtou a pátou konvergentu čísla π.

A.4.6 Cvičení Nalezněte prvních pět konvergent čísla e.

A.4.7 Cvičení Navrhněte algoritmus pro tvorbu konvergent kladných reálných čísel.

A.4.8 Cvičení Proveďte Wienerův útok na uživatele RSA s veřejným klíčem

1. $(n, e) = (2\,856\,109, 1\,037\,347)$.
2. $(n, e) = (14\,452\,147, 4\,248\,393)$.
3. $(n, e) = (9\,449\,868\,410\,449, 6\,792\,605\,526\,025)$.

Revize kapitoly

Dozvěděli jsme se:

- ✔ Konvergenty řetězových zlomků jsou výbornou aproximací kladných reálných čísel.
- ✔ Soukromý exponent d při provozování protokolu RSA s modulem $n = pq$ bychom měli volit tak, aby platila nerovnost
$$\frac{1}{c+1}\sqrt[4]{n} < d$$
kde c je nejmenší přirozené číslo splňující nerovnosti $p < q < cq$, jinak se vystavujeme nebezpečí Wienerova útoku. (Původní Wienerův útok je popsán pro případ $c = 2$, my jsme jej v textu trochu zobecnili.)

Doplňující literatura

Vynikajícím zdrojem informací o řetězových zlomcích je kniha

- ☞ K. Rosen, *Elementary Number Theory and Its Applications*, Addison-Wesley, New York, 2005

kde je popsán i faktorizační algoritmus využívající řetězové zlomky.

Příloha B

Něco o prvočíslech

> Jmenuji se Christopher Francis Boone. Znám všechny země na světě i jejich hlavní města a všechna prvočísla do 7 507.
>
> Mark Haddon, *Podivný případ se psem*

V této kapitole ukážeme některá hlubší tvrzení o prvočíslech. Zejména nás budou zajímat odhady počtu prvočísel v daném intervalu, „hustota“ rozložení prvočísel v množině všech přirozených čísel. V posledním odstavci též uvedeme některé testy prvočíselnosti daného přirozeného čísla.

B.1 Rozložení prvočísel mezi přirozenými čísly

Nekonečnou (viz lemma 2.1.3) množinu všech prvočísel budeme značit $\mathbb{P}$. Protože platí $\mathbb{P} \subseteq \mathbb{N}$, je množina $\mathbb{P}$ spočetná, a tudíž lze všechna prvočísla seřadit do rostoucí nekonečné posloupnosti p_0, p_1,...

B.1.1 Poznámka V souvislosti s důkazem lemmatu 2.1.3 si můžeme položit tuto otázku:

Je každé číslo tvaru $p_0 \cdot p_1 \cdot \ldots \cdot p_n + 1$ prvočíslo?

Odpověď je záporná. Pro prvočíslo p označme součin všech prvočísel $q \leq p$ jako $p\#$. Pro podobnost s faktoriálem se tomuto součinu říká *primoriál prvočísla p*. V článku

☞ J. P. Buhler, R. E. Crandall a M. A. Penk, Primes of the Form $n! \pm 1$ and $2 \cdot 3 \cdot \ldots \cdot p \pm 1$, *Math. Comp.* 38 (1982), 639–643

bylo například dokázáno, že

$$p\# + 1 \text{ je } \begin{cases} \text{prvočíslo pro } p = 2, 3, 5, 7, 11, 31, 379, 1\,019, 1\,021, 2\,657 \\ \\ \text{složené pro všechna další prvočísla } p,\ p < 11\,213. \end{cases}$$

Nevyřešeným problémem zůstává, zda existuje nekonečně mnoho prvočísel p, pro která je $p\# + 1$ složené číslo a zda existuje nekonečně mnoho prvočísel q, pro která je $q\# + 1$ opět prvočíslo.

Nyní ukážeme, že v množině všech přirozených čísel jsou libovolně dlouhé úseky, neobsahující žádné prvočíslo.

B.1.2 Tvrzení *Ať $n \geq 1$ je přirozené číslo. Potom existuje n po sobě jdoucích složených přirozených čísel.*

Důkaz. Definujme $a_1 = (n+1)! + 2$, $a_2 = (n+1)! + 3, \ldots$, $a_n = (n+1)! + (n+1)$. Je zřejmé, že čísla a_1, a_2, ..., a_n tvoří posloupnost délky n po sobě jdoucích přirozených čísel. Dále je zřejmé, že číslo a_1 je dělitelné číslem 2, číslo a_2 je dělitelné číslem 3, ..., číslo a_n je dělitelné číslem $n+1$. Žádné z čísel a_1, ..., a_n tedy není prvočíslo. ■

B.1.3 Poznámka Předchozí tvrzení samozřejmě neříká, že pro každé n existuje přesně n po sobě jdoucích složených čísel. Lze snadno dokázat následující tvrzení:

Je-li $r \geq 1$, pak neexistuje přesně $2r$ po sobě jdoucích složených čísel.

Při důkazu lze postupovat sporem: přepokládejme, že po prvočíslu p_k následuje přesně $2r$ po sobě jdoucích složených čísel. Protože $p_0 = 2$ a $p_1 = 3$, musí být prvočíslo p_k liché. To znamená, že další prvočíslo v řadě, číslo p_{k+1}, splňuje rovnost $p_{k+1} = p_k + (2r + 1)$. To ale znamená, že p_{k+1} je sudé číslo. To je spor, jediné sudé prvočíslo je $p_0 = 2$.

Následující jednoduché tvrzení nám dává (velmi hrubý) horní odhad velikosti n-tého prvočísla.

B.1.4 Tvrzení *Ať p_0, p_1,... je rostoucí posloupnost všech prvočísel. Potom pro každé $n \geq 0$ platí nerovnost $p_n \leq 2^{2^n}$.*

Důkaz. Budeme postupovat silným principem matematické indukce podle n.

Základní krok: $n = 0$. Protože $p_0 = 2$, je tvrzení triviální.

Indukční předpoklad: předpokládejme, že platí $p_k \leq 2^{2^k}$ pro všechna $0 \leq k \leq n$.

Uvažujme $(n+1)$-ní prvočíslo p_{n+1}. Musí platit nerovnost

$$p_{n+1} \leq p_0 \cdot p_1 \cdot \ldots \cdot p_n + 1,$$

protože přirozené číslo nalevo není dělitelné ani jedním z prvočísel $p_0, p_1, \ldots, p_n$. Podle indukčního předpokladu ale platí

$$\begin{aligned} p_0 \cdot p_1 \cdot \ldots \cdot p_n + 1 &\leq 2^{2^0} \cdot 2^{2^1} \cdot \ldots \cdot 2^{2^n} + 1 \\ &= 2^{2^0+2^1+\ldots+2^n} + 1 = 2^{2^{n+1}-1} + 1 \\ &= \frac{1}{2} \cdot 2^{2^{n+1}} + 1 \leq 2^{2^{n+1}}. \end{aligned}$$

(Poslední nerovnost plyne z toho, že $2 \leq 2^{2^{n+1}}$ a tudíž $1 \leq \frac{1}{2} \cdot 2^{2^{n+1}}$.)

Ukázali jsme tedy, že $p_{n+1} \leq 2^{2^{n+1}}$.

Podle principu matematické indukce je tvrzení dokázáno. ■

Povšimněme si, že důkaz předchozího tvrzení využíval triku podobného jako důkaz lemmatu 2.1.3.

Hlavním výsledkem o rozložení prvočísel mezi všemi přirozenými čísly je takzvaná Čebyševova[1] nerovnost (věta B.1.6), který tvrdí, že prvočísel menších než dané přirozené číslo n je „přibližně tolik“ jako hodnota zlomku $\frac{n}{\ln(n)}$, kde $\ln(n)$ je přirozený logaritmus čísla n. I když lze důkaz Čebyševovy nerovnosti provést elementárními prostředky, je technicky poměrně náročný. Proto jej neuvádíme a zájemce odkazujeme například na kapitolu 35 knihy

☞ A. A. Buchštab, *Teorija čisel*, Učpedgiz, Moskva, 1960.

V tomto textu se zaměříme spíše na některé bezprostřední důsledky Čebyševovy nerovnosti.

B.1.5 Definice Pro přirozené číslo n definujeme $\pi(n)$ jako počet všech prvočísel v množině $\{0, \ldots, n\}$.

B.1.6 Věta (Čebyševova nerovnost) *Existují kladná reálná čísla A, B taková, že $0 < A < 1$ a $B > 1$ a platí nerovnost*

$$A \cdot \frac{n}{\ln(n)} \leq \pi(n) \leq B \cdot \frac{n}{\ln(n)}$$

pro všechna $n \geq 2$.

[1] Pafnutij Lvovič Čebyšev (1821–1894) byl velmi významným ruským matematikem, který (kromě řady výsledků z teorie čísel) pracoval v teorii pravděpodobnosti a matematické analýzy.

B.1.7 Poznámka Čebyšev ve svém důkazu z roku 1848 ukázal, že je možné volit $A = 0,921$ a $B = 1,106$. V pozdějších pracech bylo ukázáno, že konstanty A a B lze volit ještě blíže číslu 1.

Důsledkem Čebyševovy nerovnosti je například přesnější odhad velikosti k-tého prvočísla p_k. I když lze důkaz následujícího tvrzení provést elementárními prostředky, neuvádíme jej.

B.1.8 Věta *Existují dvě kladné reálné konstanty C, D takové, že platí nerovnost*

$$C \cdot k \cdot \ln(k) < p_k < D \cdot k \cdot \ln(k)$$

pro všechna $k \geq 2$.

Jednoduchým důsledkem Čebyševovy nerovnosti je fakt, že v každém „dlouhém intervalu“ přirozených čísel leží alespoň jedno prvočíslo.

B.1.9 Věta *Existuje kladná konstanta K taková, že mezi čísly n, $n \geq 2$ a Kn vždy leží alespoň jedno prvočíslo.*

B.1.10 Poznámka Silnější formulací předchozí věty je *Bertrandova*[2] *věta* (dokázaná v roce 1852 Čebyševem):

Jestliže $n > 1$, pak mezi n a $2n$ leží vždy alespoň jedno prvočíslo.

Následující tvrzení lze považovat za asymptotický tvar Čebyševovy nerovnosti. Důkaz tohoto tvrzení však používá metody komplexní analýzy a vybudování tohoto aparátu by vyžadovalo delší čas. Zájemce odkazujeme například na kapitolu 19 knihy

☞ J. Bak a D. J. Newman, *Complex Analysis*, Springer-Verlag, New York, 1991.

B.1.11 Věta (O asymptotickém rozložení prvočísel) *Platí*

$$\lim_{n \longrightarrow \infty} \frac{\pi(n) \cdot \ln(n)}{n} = 1.$$

B.1.12 Poznámka Přeformulování věty pomocí definice limity dává: pro každé reálné $\varepsilon > 0$ existuje n_0 takové, že platí

$$1 - \varepsilon < \frac{\pi(n) \cdot \ln(n)}{n} < 1 + \varepsilon$$

pro všechna $n \geq n_0$.

Pro velká n lze tedy $\pi(n)$ přibližně nahradit výrazem $\dfrac{n}{\ln(n)}$. Přesnější nahrazení dává věta, kterou dokázali v roce 1896 Jacques Hadamard (1865–1963) a Charles Jean Gustave Nicholas de la Valleé-Poussin (1866–1962):

Pro velká n lze $\pi(n)$ přibližně nahradit hodnotou integrálu $\displaystyle\int_2^n \frac{dt}{\ln(t)}$.

Další metodou pro porovnání rozložení nějaké množiny A kladných přirozených čísel mezi všemi přirozenými čísly je porovnání nekonečné řady $\displaystyle\sum_{a \in A} \frac{1}{a}$ s divergentní harmonickou řadou $\displaystyle\sum_{n=1}^{\infty} \frac{1}{n}$. Lze říci, že množina A je „velká“, pokud řada $\displaystyle\sum_{a \in A} \frac{1}{a}$ také diverguje. V tomto smyslu je například množina $A = \{n^2 \mid n \geq 1\}$ malá, protože řada $\displaystyle\sum_{n=1}^{\infty} \frac{1}{n^2}$ je konvergentní. Ukážeme, že množina všech prvočísel $\mathbb{P}$ je velká.

B.1.13 Věta *Řada*

$$\sum_{p \in \mathbb{P}} \frac{1}{p}$$

je divergentní.

[2] Joseph Louis François Bertrand (1822–1900) pracoval v teorii čísel, diferenciální geometrii a teorii pravděpodobnosti (známý je *Bertrandův paradox*).

Důkaz. Podle integrálního kritéria konvergence řad je nekonečná řada $\sum_{k=2}^{\infty} \frac{1}{k \cdot \ln(k)}$ divergentní.

Podle věty B.1.8 existuje konstanta D taková, že $\frac{1}{p_k} > \frac{1}{D} \cdot \frac{1}{k \cdot \ln(k)}$ pro $k \geq 2$. Tudíž podle srovnávacího kritéria pro konvergenci řad je řada $\sum_{p \in \mathbb{P}} \frac{1}{p}$ divergentní. ■

B.2 Prvočísla speciálního tvaru

Jednoduchá modifikace důkazu lemmatu 2.1.3 dává následující tvrzení:

B.2.1 Lemma *Existuje nekonečně mnoho prvočísel tvaru* $4n + 3$.

Důkaz. Nejprve si uvědomme následující jednoduché tvrzení:

(*) Každé prvočíslo větší než 2 je tvaru $4t + 1$ nebo $4t + 3$.

Předpokládejme, že všech prvočísel tvaru $4n + 3$ je pouze konečně mnoho a ať $p_1, \ldots, p_k$ je jejich úplný výčet. Označme $x = p_1 \cdot \ldots \cdot p_k$.

Číslo $4x - 1$ je větší než 1 a musí mít ve svém prvočíselném rozkladu alespoň jedno prvočíslo p tvaru $4t + 3$. (Uvažujte v $\mathbb{Z}_4$.) Tudíž prvočíslo p je jedním z čísel $p_1, \ldots, p_k$, speciálně $p \geq 3$.

Číslo p tedy dělí číslo x a současně dělí číslo $4x - 1$. Existují tedy přirozená čísla a a b taková, že $x = p \cdot a$ a $4x - 1 = p \cdot b$. Odečtením čtyřnásobku první rovnosti od druhé dostáváme $-1 = p \cdot (b - 4a)$. To je spor s tím, že $p \geq 3$. ■

B.2.2 Poznámka Předchozí lemma samozřejmě *neříká*, že všechna čísla tvaru $4n + 3$ jsou prvočísla. Zvolte například $n = 3$, potom $4n + 3 = 15$, což je složené číslo.

Lemma je speciálním případem *Dirichletovy*[3] *věty* z teorie čísel: jsou-li a, b nesoudělná přirozená čísla, pak existuje nekonečně mnoho prvočísel tvaru $a \cdot n + b$. Obecný důkaz Dirichletovy věty je technicky velmi náročný, a proto jej neuvádíme.

Podobným způsobem jako v lemmatu B.2.1 lze však ukázat další speciální případ Dirichletovy věty: *existuje nekonečně mnoho prvočísel tvaru* $6n + 5$. (Návod: Definujte číslo x podobně jako v důkazu lemmatu B.2.1. Ukažte, že existuje prvočíslo tvaru $6n + 5$ takové, že p dělí čísla x a $6x - 1$.)

Následující lemma ukazuje, že například žádné číslo tvaru $100^n - 1$ není prvočíslo.

B.2.3 Lemma *Ať a, n jsou přirozená čísla, $n \geq 2$. Je-li číslo tvaru $a^n - 1$ prvočíslo, pak $a = 2$ a n je prvočíslo.*

Důkaz. Protože $n \geq 2$, můžeme psát $n = k + 1$, $k \geq 1$. Dále platí (lze snadno dokázat indukcí)

$$a^n - 1 = a^{k+1} - 1^{k+1} = (a - 1) \cdot (a^k + a^{k-1} + \cdot \ldots \cdot + a^1 + 1).$$

Protože předpokládáme, že $a^n - 1$ je prvočíslo, musí být $a - 1 = 1$, neboli $a = 2$.

Předpokládejme, že n není prvočíslo, tj. existují přirozená čísla u, v taková, že $n = u \cdot v$ a $1 < u < n$, $1 < v < n$. Potom platí

$$2^n - 1 = 2^{u \cdot v} - 1^{u \cdot v} = (2^u)^v - (1^u)^v = (2^u - 1^u) \cdot ((2^u)^{v-1} + (2^u)^{v-2} + \cdot + 2^u + 1).$$

Protože předpokládáme, že $2^n - 1$ je prvočíslo, musí být $2^u - 1^u = 1$, neboli $u = 1$. To je spor s naší volbou čísla u. ■

[3] O Gustavu Lejeune Dirichletovi (1805–1859) se říká, že byl první, kdo dokonale porozuměl knize *Disquisitiones Arithmeticae* od Karla Friedricha Gausse. Kromě teorie čísel pracoval Dirichlet v analýze a po něm pojmenovaný *Dirichletův princip* se používá v kombinatorice.

B.2.4 Poznámka Čísla tvaru $2^p - 1$, kde p je prvočíslo, se nazývají *Mersennova*[4] *čísla*. I když řada Mersennových čísel jsou opravdu prvočísla, není tomu tak pro každé Mersennovo číslo. Například pro $p = 2, 3, 5, 7$ jsou příslušná Mersennova čísla prvočísla. Číslo $2^{11} - 1$ je však číslo složené.

Největší dosud[5] známé Mersennovo prvočíslo je číslo

$$2^{32\,582\,657} - 1$$

Toto Mersennovo prvočíslo má 9 808 358 cifer a jde dokonce o *největší dosud známé prvočíslo* (ke dni 16. 6. 2007).

Číslům tvaru $2^n + 1$ se říká *Fermatova čísla*. Pro některé hodnoty n má $2^n + 1$ prvočíselnou hodnotu. O takových prvočíslech lze dokázat následující tvrzení.

B.2.5 Lemma *Ať číslo $2^n + 1$ je prvočíslo. Potom je buď $n = 0$ nebo $n = 2^k$ pro nějaké $k \geq 0$.*

Důkaz. Předpokládejme, že $n \geq 2$. Stačí ukázat, že každý dělitel čísla n je sudý.

Předpokládejme, že číslo n má lichého dělitele $d > 1$. Potom platí

$$2^n + 1 = (2^{\frac{n}{d}} + 1) \cdot (2^{\frac{n}{d}(d-1)} - 2^{\frac{n}{d}(d-2)} + \ldots - 2^{\frac{n}{d}} + 1)$$

Protože $n \geq 3$ (předpokládáme existenci lichého dělitele) a protože $d > 1$, je číslo $2^n + 1$ složené. ■

Pierre de Fermat zformuloval hypotézu, že každé číslo tvaru $2^{2^k} + 1$ je prvočíslo. Tuto hypotézu ovšem vyvrátil v roce 1739 Leonard Euler: ukázal, že číslo $2^{2^5} + 1$ je složené. Dále ukázal, že každý dělitel čísla $2^n + 1$ musí být tvaru $2^{(k+1)t} + 1$ pro nějaké t.

Fermatova prvočísla souvisí s úlohou konstrukce pravidelných mnohoúhelníků s použitím kružítka a pravítka. Karl Friedrich Gauss dokázal, že pravidelný n-úhelník lze sestrojit pomocí pravítka a kružítka právě tehdy, když n je tvaru

$$2^a \cdot p_1 \cdot \ldots \cdot p_r$$

kde $p_1, \ldots, p_r$ jsou navzájem různá Fermatova prvočísla. Nelze tedy, například, pomocí kružítka a pravítka sestrojit pravidelný sedmiúhelník. Mezi prvními tisíci přirozenými čísly existuje pouze 54 čísel n, pro která lze kružítkem a pravítkem sestrojit pravidelný n-úhelník.

Problémy okolo Mersennových a Fermatových prvočísel nastolují otázku, zda existují funkce, jejichž hodnoty jsou vždy prvočísla. Následující tvrzení ukazuje, že žádný polynom takovou funkcí být nemůže.

B.2.6 Tvrzení *Předpokládejme, že polynom $P(x)$ má celočíselné koeficienty. Potom existuje celé číslo a takové, že $P(a)$ je složené číslo.*

Důkaz. Označme $P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0 +$, kde čísla $a_n, \ldots, a_0$ jsou celá, $a_n \neq 0$. Pokud je každá hodnota polynomu $P(x)$ složené číslo, není co dokazovat. Předpokládejme tedy, že pro nějaké celé číslo k je hodnota $P(k)$ prvočíslo.

Potom existuje přirozené číslo $t > 1$ takové, že $k + tp > 1$ a platí $P(k + tp) \neq p$. To vyplývá z toho, že polynom stupně n může mít hodnotu p nanejvýš v n různých bodech. Rozviňme nyní $P(k + tp)$ do Taylorovy řady

$$P(k + tp) = P(k) + c_1 tp + c_2 (tp)^2 + \ldots + c_n (tp)^n$$

Potom p dělí $P(k + tp)$ a tudíž číslo $a = k + tp$ je složené. ■

Existují funkce, jejichž hodnoty jsou pouze prvočísla, předchozí věta jen naznačuje, že jde o složité funkce. Například v článku

☞ W. H. Mills, A Prime-representing function, *Bulletin AMS* 53 (1947), 604

[4] Marin Mersenne (1588–1648), člen řádu *Minimů*, je autorem řady knih o matematice, fyzice, hudbě a akustice. Mersenne čile korespondoval s řadou evropských vědců své doby a je znám i pro svou obhajobu René Descarta a Galileo Galilea před náboženskými kritiky.

[5] Objeveno 4. září 2006 Stevenem R. Boonem a Curtisem Cooperem.

je ukázána existence reálného čísla A takového, že číslo

$$[A^{3^n}]$$

je prvočíslo pro všechna přirozená čísla $n \geq 1$. W. H. Mills ukázal pouze existenci, nikoli hodnotu, čísla A. Navíc, konstanta A není jednoznačně určena: pokud vyhovuje hodnota A, potom i hodnoty A^3, A^9, ... vyhovují Millsovu výsledku. V současnosti panuje domněnka, že nejmenší hodnotou čísla A je číslo

$$1,306\,377\,883\,863\,080\ldots,$$

která je označována jako *Millsovo číslo*. Paradoxem je, že pro určení hodnoty Millsova čísla je zapotřebí znát dostatečné množství prvočísel, takže pro praktické účely je Millsův výsledek nepoužitelný.

Snad nejdůležitějšími aplikacemi vlastností prvočísel jsou některé algoritmy teorie šifrování. Pro tvorbu bezpečných šifrovacích klíčů je nutné znát velká prvočísla a při kryptoanalýze pak je pak důležité umět zjistit prvočíselný rozklad velkého čísla.

Testům prvočíselnosti a faktorizačním algoritmům jsou věnovány další dva odstavce.

B.3 Elementární testy prvočíselnosti

Zaměříme se na otázku zjištění prvočíselnosti. První test je jednoduchý a dobře známý, jde o test prvočíselnosti hrubou silou (viz například příklad 3.6.1).

B.3.1 Věta *Jestliže přirozené číslo $n > 1$ není dělitelné ani jedním prvočíslem menším nebo rovným $\sqrt{n}$, potom n je prvočíslo.*

Důkaz. Předpokládejme, že n je složené číslo. Existují tedy přirozená čísla a a b taková, že $n = a \cdot b$ a $1 < a < n$, $1 < b < n$.

Čísla a, b nemohou být obě současně větší než $\sqrt{n}$, protože pak by jejich součin byl větší než n. Předpokládejme tedy, že platí například $a \leq \sqrt{n}$.

Protože $a > 1$, existuje prvočíslo p takové, že a je dělitelné číslem p. Potom i číslo n je dělitelné číslem p. Protože zřejmě $p \leq a \leq \sqrt{n}$, dostáváme spor s předpokladem věty. ■

Modifikace předchozí věty je známa jako Eratosthenovo[6] síto; jde o algoritmus, který pro pevné přirozené číslo $n \geq 2$ nalezne postupně všechna prvočísla mezi čísly 2 a n.

B.3.2 Věta (Eratosthenovo síto) *Označme jako p_1, p_2, ... rostoucí posloupnost všech prvočísel. Ať $n \geq 2$ je přirozené číslo. Potom platí:*

1. *Ať $r \geq 1$ je přirozené číslo. Jestliže v množině $\{2, 3, \ldots, n\}$ zaškrtneme všechny násobky prvních r prvočísel $p_1, \ldots, p_r$, potom nejmenší nezaškrtnuté číslo je prvočíslo.*
2. *Ať $r \geq 1$ je přirozené číslo takové, že platí $p_r \leq \sqrt{n} < p_{r+1}$. Jestliže v množině $\{2, 3, \ldots, n\}$ zaškrtneme všechny násobky prvních r prvočísel $p_1, \ldots p_r$, potom všechna nezaškrtnutá čísla tvoří přesně množinu všech prvočísel p takových, že $\sqrt{n} < p \leq n$.*

Důkaz. Zřejmě platí následující dvě podmínky:

1. Každé složené číslo a v množině $\{2, 3, \ldots, n\}$ je dělitelné některým prvočíslem menším než a. Jestliže číslo a není dělitelné žádným prvočíslem menším než a, pak a je prvočíslo.
2. Každé složené číslo a takové, že $\sqrt{n} < a \leq n$ je podle věty B.3.1 dělitelné alespoň jedním prvočíslem p, pro které je $p \leq \sqrt{a}$. Toto prvočíslo p je však jedním z prvočísel $p_1, \ldots, p_r$. Číslo a tedy bude vyškrtnuto. Prvočísla p taková, že $p > \sqrt{n}$ nejsou dělitelná ani jedním z prvočísel $p_1, \ldots, p_r$ — tudíž nebudou vyškrtnuta.

[6] Eratosthenes (276–194) byl učitelem syna Ptolemaia II. v Alexandrii a později se stal hlavním knihovníkem ve známé alexandrijské knihovně. Je znám i svým měřením velikosti obvodu země. Podle Eratosthena má země obvod dlouhý 250 000 stadionů, což je přibližně 40 000 km (to je vynikající aproximace skutečné hodnoty). Eratostenův způsob měření je však zatížen mnoha chybami, snad nejzajímavějším faktem je to, že měření se opírá i o odhadovanou průměrnou rychlost karavany velbloudů. O úspěších starověké astronomie píše velmi zajímavě Thomas S. Kuhn (1923–1996) v knize *The Copernican Revolution — Planetary Astronomy in the Development of Western Thought*, Harvard University Press, 1977.

Proto je postup Eratosthenova síta korektním algoritmem. ■

Elementární testy prvočíselnosti, které v dalším zmíníme, jsou založeny na malé Fermatově větě (věta 3.4.7). Pro pohodlí připomeňme její znění:

B.3.3 Věta (malá Fermatova věta) *Ať p je prvočíslo. Potom pro libovolné celé číslo a nesoudělné s p platí*

$$a^{p-1} = 1 \quad \text{v } \mathbb{Z}_p$$

Eulerovo zobecnění malé Fermatovy věty říká, že pro všechna čísla a nesoudělná s n platí rovnost $a^{\varphi(n)} = 1$ v $\mathbb{Z}_n$. Exponent $\varphi(n)$ ale nemusí být nejmenší číslo s, pro který rovnost $a^s = 1$ v $\mathbb{Z}_n$ platí.

Nejmenšímu takovému exponentu říkáme *řád a v $\mathbb{Z}_n$* a značíme jej $ord(a, n)$. Je zřejmé, že $ord(a, n)|\varphi(n)$.

Následující tvrzení dokázal François Edouard Anatole Lucas (1842–1891)[7] v roce 1876. Lze jej použít jako test prvočíselnosti.

B.3.4 Tvrzení *Ať $n > 1$. Předpokládejme, že existuje přirozené číslo $1 < a < n - 1$ takové, že platí*

1. $a^{n-1} = 1$ v $\mathbb{Z}_n$.
2. $a^s \neq 1$ v $\mathbb{Z}_n$ *pro všechna* $s \in \{1, 2, \ldots, n-2\}$.

Potom n je prvočíslo.

Důkaz. Stačí ukázat, že $\varphi(n) = n - 1$. K tomu stačí ukázat, že v $\mathbb{Z}_n$ existuje číslo a, $0 \neq a \neq 1$, řádu $n - 1$. Existenci takového čísla ale předpokládáme. ■

V roce 1891 Lucas předchozí test vylepšil. Následující tvrzení má stejný důkaz, jako tvrzení B.3.4.

B.3.5 Tvrzení *Ať $n > 1$. Předpokládejme, že existuje přirozené číslo $1 < a < n - 1$ takové, že platí*

1. $a^{n-1} = 1$ v $\mathbb{Z}_n$.
2. $a^s \neq 1$ v $\mathbb{Z}_n$ *pro všechna* $s \in \{1, 2, \ldots, n-2\}$, *která dělí číslo* $n - 1$.

Potom n je prvočíslo.

Následující algoritmus je jedním z nejlepších způsobů, kterým lze rozhodnout, zda velké liché číslo n je složené.

Protože n je liché můžeme psát

$$n - 1 = 2^s \cdot t$$

kde $t \geq 1$ je liché přirozené číslo. Zvolme nyní náhodně číslo a takové, že platí $1 < a < n-1$. Nejprve spočítáme mocninu $z = a^t$ v $\mathbb{Z}_n$ (algoritmem opakovaných čtverců — viz příklad 3.4.16). Potom opakovaně mocníme z na druhou, dokud nedostaneme $z^{2^s} = a^{n-1}$ v $\mathbb{Z}_n$:

$$z,\ z^2,\ z^4,\ \ldots,\ z^{2^s} = a^{n-1} \tag{B.1}$$

Připomeňme, že pokud n skutečně *je* prvočíslo, pak platí:

1. $z^{2^s} = a^{n-1} = 1$ v $\mathbb{Z}_n$. To je tvrzení malé Fermatovy věty. (Protože platí $1 < a < n - 1$, jsou jistě n a a navzájem nesoudělná čísla.)
2. Pokud všechny mocniny v (B.1) nejsou v $\mathbb{Z}_n$ rovny 1, pak první hodnota 1 v (B.1) je předcházena hodnotou $n - 1$. To plyne okamžitě z následujícího tvrzení:

   B.3.6 Tvrzení *Ať p je liché prvočíslo. Pak rovnice $x^2 = 1$ má v $\mathbb{Z}_p$ pouze dva (různé!) kořeny: $x_1 = 1$ a $x_2 = -1 = p - 1$.*

[7] Kromě výsledků z teorie čísel (v roce 1876 například dokázal, že $2^{127} - 1$ je prvočíslo) je Lucas znám i svými příspěvky k teorii algoritmů, snad nejznámějším je jeho analýza problému *Hanojských věží*, viz cvičení 1.6.21. Lucas zemřel na infekci způsobenou bizarním incidentem: na banketu francouzské asociace pro pokrok ve vědách byl pořezán upuštěným talířem.

Důkaz. Předpokládejme, že pro b v $\mathbb{Z}_p$ platí $b^2 = 1$. Potom $0 = b^2 - 1 = (b-1)\cdot(b+1)$, tudíž platí buď $b = 1$ nebo $b = -1 = p - 1$, protože Z_p je těleso. ■

Pokud číslo n splňuje obě výše uvedené podmínky, řekneme, že n *splňuje silný Fermatův test při základu* a.

B.3.7 Věta *Pokud číslo n nesplňuje silný Fermatův test při základu a, jde určitě o složené číslo. Pokud číslo n tento test splňuje, pak jde o prvočíslo s pravděpodobností větší než 75%.*

Podle této věty tedy platí:

B.3.8 Millerův-Rabinův test prvočíselnosti Pokud číslo n splňuje silný Fermatův test pro k náhodně zvolených čísel a, potom je pravděpodobnost toho, že n je prvočíslo, větší než $1 - \frac{1}{4^k}$.

Například pro $k = 20$ je $1 - \frac{1}{4^k}$ (tedy očekávaný počet složených čísel, která algoritmus jako složená čísla nerozpozná) jedno číslo z 10^{12}.

B.4 Elementární faktorizační algoritmy

V tomto odstavci zmíníme tři „pravěké“ faktorizační algoritmy: *postupné dělení*, *rozdíl čtverců* a *Pollardovu $p-1$ metodu*. Moderní používané faktorizační algoritmy používají matematický aparát, který je značně mimo rozsah těchto poznámek, odkazujeme na seznam doplňující literatury na konci kapitoly.

Nejjednodušším (a nejpomalejším) faktorizačním algoritmem je *postupné dělení*. Je založen na větě B.3.1.

B.4.1 Postupné dělení Chceme-li faktorizovat číslo n, dělme jej postupně přirozenými čísly p splňujícími $2 \le p \le \sqrt{n}$ a spočtěme zbytek při dělení.

Přestože má algoritmus postupného dělení exponenciální složitost, ukazuje se být poměrně rozumnou volbou pro čísla menší než 10^{12}.

Dalším, velmi starým faktorizačním trikem je nalezení dvou čísel a a b splňujících rovnost

$$a^2 = b^2 \quad \text{v } \mathbb{Z}_n$$

protože potom platí rovnost

$$a^2 - b^2 = (a-b)\cdot(a+b) = 0 \quad \text{v } \mathbb{Z}_n$$

a my můžeme (pro modul protokolu RSA) postupovat následovně:

B.4.2 Faktorizace rozdílem čtverců Budeme předpokládat, že n je součin dvou neznámých různých prvočísel p a q a že jsme nalezli čísla a a b splňující rovnost $a^2 = b^2$ v $\mathbb{Z}_n$. Pak spočtením $d = \gcd(a-b, n)$ nalezneme jedno z prvočísel p, q, s pravděpodobností $\frac{1}{2}$.

Všimněme si, že faktorizace rozdílem čtverců je *pravděpodobnostní* algoritmus. Důvodem je to, že s pravděpodobností $\frac{1}{4}$ nastane právě jedna ze čtyř možností:

1. p dělí $a-b$ a q dělí $a+b$.
2. p dělí $a+b$ a q dělí $a-b$.
3. p i q dělí $a-b$, ale ani p ani q nedělí $a+b$.
4. p i q dělí $a+b$, ale ani p ani q nedělí $a-b$.

Spočítáme-li $d = \gcd(a-b, n)$, potom výše uvedené případy přejdou na případy

1. $d = p$.
2. $d = q$.

3. $d = n$.

4. $d = 1$.

a tudíž s pravděpodobností $\frac{1}{2}$ nacházíme jedno z prvočísel p, q.

Problémem je samozřejmě najít čísla a a b, které faktorizace rozdílem čtverců vyžaduje (srovnejte s důkazem lemmatu 3.6.2). Viz také cvičení B.5.6.

Pollardova $p-1$ metoda je vylepšením metody postupného dělení. Funguje však pouze pro čísla určitého typu a my ji zformulujeme opět pouze pro moduly protokolu RSA, tedy pro čísla, která jsou součinem dvou různých prvočísel.

Nejprve zavedeme následující technický pojem:

B.4.3 Definice Ať b je přirozené číslo. Řekneme, že přirozené číslo n je

1. *b-hladké*,[8] když každé prvočíslo z prvočíselného rozkladu n je menší než b.

2. *b-superhladké*,[9] když každá mocnina prvočísla, která dělí n, je menší než b.

Zřejmě je každé b-superhladké číslo také b-hladké. Uveďme příklad:

B.4.4 Příklad Číslo

$$n = 3\,361\,494\,409\,216 = 2^{13} \cdot 17^7$$

je 18-hladké, protože každé prvočíslo z jeho prvočíselného rozkladu je menší než 18. Číslo n je $410\,338\,674$-superhladké, protože platí $2^{13} = 8\,192$ a $17^7 = 410\,338\,673$, takže všechny mocniny 2 a všechny mocniny 17, které dělí n, jsou menší než $410\,338\,674$.

Jednoduchým pozorováním, které plyne z definice, je následující:

B.4.5 Lemma *Číslo n je b-superhladké právě tehdy když číslo n dělí faktoriál čísla b.*

Důkaz. Ať $n = p_1^{n_1} \cdot \ldots \cdot p_r^{n_r}$ je prvočíselný rozklad čísla n. Pak platí $p_i^{n_i} < (b+1)$ pro všechna $i = 1, \ldots, r$ právě tehdy, když n dělí součin všech čísel od 1 do b. ■

B.4.6 Pollardova $p-1$ metoda Chceme faktorizovat číslo n, které je součinem neznámých různých prvočísel p a q. Předpokládejme, že víme, že $p-1$ je b-superhladké číslo a že $q-1$ není b-superhladké číslo pro nějaké (nám známé) číslo b.

Spočteme $a = 2^{b!}$ v $\mathbb{Z}_n$ a $d = \gcd(a-1, n)$. Pokud neplatí ani $d = 1$ ani $d = n$, je $d = p$.

Důkaz. Podle lemmatu B.4.5 existuje k tak, že $b! = (p-1) \cdot k$. Z malé Fermatovy věty tedy dostáváme rovnost $a = 2^{b!} = (2^{p-1})^k = 1$ v $\mathbb{Z}_p$.

Dále platí $a \neq 1$ v $\mathbb{Z}_q$: to ale plyne okamžitě z toho, že $q-1$ není b-superhladké.

Ukázali jsme tedy, že p dělí $a-1$ a současně q nedělí $a-1$. Proto pro $d = \gcd(a-1, n)$ platí buď $d = 1$ nebo $d = n$ nebo $d = p$. ■

Pollardovu $p-1$ metodu ukážeme na školním příkladu.

B.4.7 Příklad Faktorizujme modul RSA $n = 812\,423$ a předpokládejme, že číslo b je 26.

Spočítáme $a = 2^{26!} = 84\,081$ v $\mathbb{Z}_{812\,423}$ a $d = \gcd(a-1, n) = \gcd(84\,080, 812\,423) = 1\,051$.

To nám dává hledanou faktorizaci

$$812\,423 = 1\,051 \cdot 773$$

Nevýhodou Pollardovy $p-1$ metody je nutnost odhadnout číslo b tak, aby byly splněny předpoklady této metody. V aplikacích se často volí $b \approx (\log n)^k$ pro nějaké k.

[8] V angličtině *b-smooth*.
[9] V angličtině *b-power smooth*.

B.4.8 Poznámka Díky Pollardově $p-1$ metodě se často doporučuje, aby prvočísla p, q, použitá při tvorbě klíče účastníka RSA, splňovala rovnosti

$$p-1=2p_1 \quad \text{a} \quad q-1=2q_1$$

kde p_1 i q_1 jsou prvočísla.

Součin takových prvočísel p, q je totiž odolný proti Pollardově $p-1$ metodě: splňují-li (bez újmy na obecnosti) prvočísla p_1 a p_2 nerovnost $p_1 < q_1$, pak je nutné volit b alespoň tak velké, jako p_1+1 a pro velká prvočísla p, q se tak Pollardova $p-1$ metoda stává velmi zdlouhavou.

B.5 Cvičení

B.5.1 Cvičení Spočtěte $13\#+1$ (pro definici primoriálu $p\#$ viz stranu 183) a výslednou hodnotu označte m. Ukažte, že m je složené číslo.

B.5.2 Cvičení Spočtěte (numerickou metodou) integrál $\int_2^{100} \frac{dt}{\ln(t)}$ a hodnotu $\frac{100}{\ln(100)}$ a porovnejte výsledky s hodnotou $\pi(100)=24$.

B.5.3 Cvičení Ukažte, že Mersennovo číslo $2^{11}-1$ je složené.

B.5.4 Cvičení Ukažte, že Fermatovo číslo $2^{2^5}+1$ je složené.

B.5.5 Cvičení Ukažte, že číslo 2 047 splňuje Fermatův test při základu 2. Je 2 047 prvočíslo?

B.5.6 Cvičení Ukažte, že liché číslo n lze faktorizovat rozdílem čtverců následujícím způsobem: prohledejte (konečnou!) posloupnost čísel

$$t^2-n, \quad (t+1)^2-n, \quad (t+2)^2-n, \quad (t+3)^2-n, \quad \ldots, \left(\frac{n+1}{2}\right)^2-n$$

kde t je nejmenší přirozené číslo větší než $\sqrt{n}$ a hledejte v této posloupnosti číslo tvaru b^2. Pokud platí rovnost $(t+i)^2-n=b^2$, můžete faktorizovat n rozdílem čtverců.

Tento algoritmus vždy skončí, protože platí rovnost

$$\left(\frac{n+1}{2}\right)^2-n=\left(\frac{n-1}{2}\right)^2$$

ze které obdržíte triviální faktorizaci $n=1\cdot n$.

B.5.7 Cvičení Ukažte, že v $\mathbb{Z}_{6\,077}$ platí rovnost $81^2=22^2$ a využijte ji k faktorizaci čísla 6 077.

B.5.8 Cvičení Předpokládejte, že $n=p\cdot q$, kde p a q jsou různá prvočísla. Definujte v $\mathbb{Z}_n$ rekurentně posloupnost $a_0=2$, $a_{k+1}=a_k^2+1$.

1. Ukažte, že pokud pro různá k, l platí rovnost $a_k=a_l$ v $\mathbb{Z}_p$, pak lze n (s vysokou pravděpodobností) faktorizovat spočítáním $\gcd(a_k-a_l,n)$.
2. Ukažte, že různá k, l, pro která platí rovnost $a_k=a_l$ v $\mathbb{Z}_p$, musí existovat.

Na základě těchto faktů navrhněte faktorizační algoritmus. (Říká se mu *Pollardova ρ metoda*.)

Revize kapitoly

Dozvěděli jsme se:

- ✔ Prvočísla jsou mezi přirozenými čísly rozložena poměrně hustě (Bertrandova věta). Dovedeme také (asymptoticky) určit počet prvočísel menších než dané přirozené číslo (Čebyševova nerovnost).
- ✔ Testy prvočíselnosti a faktorizační algoritmy, které jsme zatím schopni vysvětlit, jsou poměrně těžkopádné. Pro moderní algoritmy je třeba znát více teorie!

Doplňující literatura

Studiem vlastností prvočísel (a problémy týkající se přirozených čísel) se zabývá matematická disciplína nazvaná *teorie čísel*. Existuje řada knih o teorii čísel, ve kterých lze předchozí výsledky nalézt. Elementárním způsobem o základech teorie čísel pojednává například kniha

☞ K. Rosen, *Elementary Number Theory and Its Applications*, Addison-Wesley, New York, 2005

Vlastnostem prvočísel, velkým prvočíslům a jejich některým aplikacím je věnována například stránka

☞ http://www.utm.edu/research/primes

ze které jsme čerpali informace o největších dosud známých prvočíslech.

Obsáhlým a vynikajícím tištěným zdrojem informací o prvočíslech je kniha

☞ P. Ribenboim, *The New Book of Prime Number Records*, Springer-Verlag, New York, 1996

V současnosti je atraktivní aplikací vlastností prvočísel tvorba šifrovacích protokolů. O možné doplňující literatuře k šifrování se informujte v kapitole 3.

O moderních a rychlých testech prvočíselnosti a faktorizačních algoritmech (například *Number Field Sieve*) se lze dočíst například v knihách

☞ N. Smart, *Cryptography: An Introduction*, McGraw-Hill, 2003

☞ K. Rosen, *Elementary Number Theory and Its Applications*, Addison-Wesley, New York, 2005

Rejstřík