Úvod do matematické logiky

Jan Starý

9. května 2025

Obsah

Rostoucí studijní text k přednáškám *Matematická logika* a *Vyčíslitelnost* konaným na Fakultě informačních technologií ČVUT v Praze v letech 2012–2025. Případné chyby a připomínky prosím zasílejte na `jan.stary@fit.cvut.cz`.

Úvod

V tomto textu budeme zkoumat matematickou logiku jakožto vyjadřovací a důkazový aparát matematiky, potažmo informatiky. Základem tohoto aparátu je dosti jednoduchý formální jazyk, ve kterém se formuluje vše potřebné pro matematickou praxi. Chceme čtenáře nejprve přesvědčit, že prozkoumání základů matematických metod, jazykem počínaje, je užitečné i potřebné.

Jazyk matematiky Předně, proč vůbec nějaký formální jazyk zavádět? Většinou mluvíme přirozeným jazykem (třeba česky), a zjevně s tím dobře vystačíme. Proč by tomu v matematice mělo být jinak?

Relativně nedávno bylo běžné zavést matematický pojem třeba takto:

> *Je-li dána nějaká posloupnost reálných čísel* $a_1, a_2, \ldots, a_n, a_{n+1}, \ldots$*, uvažme soubor všech reálných čísel* x *s tou vlastností, že zvolíme-li jakékoli reálné číslo od uvažovaného čísla* x *svou velikostí se byť jen nepatrně lišící, pak za každým číslem z dané posloupnosti, libovolně daleko se vyskytujícím, najde se v dané posloupnosti ještě nějaké pozdější, které se bude od uvažovaného čísla* x *lišiti ještě méně.*

To je definice hromadných bodů posloupnosti. Čtení takového textu je ale možná náročnější než jeho matematický obsah. To je jeden z dobrých důvodů, které vedly ke vzniku formálního jazyka matematiky: úspornost vyjádření. Vskutku, současným *epsilon-delta* jazykem analýzy a teorie množin vyjádříme totéž výrazem $\{x \in \mathbb{R}; (\forall \varepsilon > 0)(\forall m \in \mathbb{N})(\exists n > m)|a_n - x| < \varepsilon\}$.

Přirozený jazyk je bohatý a mnohoznačný. To však může být i překážkou, pokud se naopak potřebujeme vyslovit jednoznačně, aby nemohlo být žádných pochyb o tom, co přesně jsme měli na mysli. Přirozenému jazyku jsou zároveň vlastní různé nepravidelnosti a výjimky, od kterých je formální jazyk oproštěn.

Nejpodstatnější důvod, proč zavést pro účely matematického vyjadřování nějaký jiný jazyk než ten, kterým běžně mluvíme, je ale to, že sám jazyk nás při neopatrném zacházení může svést na scestí. Připomeňme známý *Berryho paradox*, ve kterém se definuje *nejmenší přirozené číslo, které nelze definovat méně než jedenácti slovy*, ale právě jsme je „definovali“ deseti slovy. K takovým paradoxům vede použití jazyka, který může „mluvit sám o sobě.“ Tentýž jazyk, ve kterém je „definice“ vedena, vystupuje zároveň jako *metajazyk*, který hovoří *o* definicích, když používá takové pojmy jako „nelze definovat.“ Od jazyka matematiky očekáváme užitečný prostředek vyjadřování, nikoli nástroj paradoxních tvrzení o sobě. Přirozený jazyk však podobné úskoky umožňuje.

Matematika používá mnohem jednodušší formální jazyk, ve kterém lze podávat definice, formulovat teorie, vést důkazy, atd. Přirozený jazyk zůstane v roli *metajazyka*, kterým budeme neformálně mluvit *o* definicích, důkazech, atd. Popis tohoto jazyka rozdělíme do dvou tradičních kroků: nejprve popíšeme jazyk *výrokových spojek*, který zkoumá *výroková logika*, a později tento jazyk zjemníme zavedením *kvantifikátorů* a *relačních symbolů* při studiu *predikátové logiky*. Tvrzení vyjádřená v tomto formálním jazyce se nazývají *formule*.

Co je to důkaz? Co přesně obnáší *dokázat* nějaké (matematické) tvrzení? Lze pojem důkazu zavést přesně, abychom korektní důkazy efektivně rozpoznali a mohli je podrobně zkoumat matematickými prostředky?

Čtenář jistě má intuitivní představu o tom, co by měl „důkaz“ být: vychází z nějakých očividně pravdivých nebo výslovně přijatých předpokladů, každý jeho krok je očividně korektní, a po konečně mnoha takových krocích dospěje k závěrečnému tvrzení, které je tím prokázáno nade všechnu pochybnost. Jako příklad nechť laskavý čtenář posoudí následující argument — je to „důkaz“?

> Buď $<$ binární relace taková, že (i) při $x < y$ a $y < z$ je také $x < z$; (ii) pro žádné x není $x < x$. Potom pro žádné $x < y$ není $y < x$.
>
> Buď totiž pro nějaké x, y zároveň $x < y$ a $y < x$. Pak ale díky (i) je také $x < x$. To však není možné, díky podmínce (ii). To znamená, že taková x, y nemohou existovat.

Matematická logika zavádí pojem *formálního důkazu*: je to konečná posloupnost formulí, ve které každá formule je buďto předem výslovně uvedeným *axiomem*, nebo ji lze z nějakých dříve již dokázaných formulí *odvodit* pomocí některého z předem výslovně daných *odvozovacích pravidel*. Je samozřejmě otázka, které axiomy a jaká pravidla by to měla být. Popíšeme *Hilbertův systém* pro predikátovou logiku, který je zavedeným standardem.

Výše uvedený argument není formálním důkazem v tomto smyslu, ostatně není to vůbec posloupnost formulí. Je to tzv. *neformální důkaz*, jaký matematik běžně předkládá. S trochou úsilí jej však lze do formálního důkazu přepsat.

Je důležité si uvědomit, že ve formálním důkaze nehraje „význam“ symbolu $<$ žádnou roli. Jedná se o pouhou manipulaci se symboly, čistě syntaktický postup, který není závislý na tom, jakou přesně relaci symbol $<$ označuje a co tedy „znamenají“ podmínky (i) a (ii). Čtenář obeznámený s pojmem uspořádané množiny si jistě všimne, že taková relace je ostrým uspořádáním, a právě jsme dokázali, že je antisymetrická. Korektnost formálního důkazu ale nestojí na tomto (ani žádném jiném) porozumění, může být ověřena zcela mechanicky.

Můžeme se zároveň ptát, zda je možné rozhodnout o *dokazatelnosti* ještě předtím, než začneme nějaký důkaz hledat. Uvidíme, že ve výrokové logice je to možné, v predikátové logice nikoli. Víme-li nicméně o nějaké dané formuli předem, že je dokazatelná, můžeme nějaký její důkaz efektivně nalézt.

Syntax a sémantika Jazyk predikátové logiky má — tak jako každý jazyk, přirozený či umělý — svou *syntax* a svou *sémantiku*. Syntax stanovuje gramatická pravidla: které výrazy považujeme vůbec za slova a věty (*termy* a *formule*), a jakými způsoby můžeme z jednoduchých výrazů skládat složitější, tak

jako v přirozeném jazyce skládáme slova do vět a věty do souvětí. Syntaktické otázky jsou čistě formální: zkoumají výrazy jazyka jen jako řetězce symbolů. Specielně formální důkazy, jakožto jisté posloupnosti formulí, jsou takovými syntaktickými útvary. Sémantika dává výrazům jazyka *význam*, a táže se po *pravdivosti* formulí. To je styčný bod logiky s filozofií: v jistém jazyce (*logos*) se snažíme popsat objekty matematiky, a nad tvrzeními tohoto jazyka se tážeme: Je to pravda? Dá se to dokázat? Je mezi těmito dvěma věcmi nějaký vztah?

Ukážeme, že Hilbertův systém je *korektní* a *úplný*. To znamená, že každá v něm dokazatelná formule je pravdivá, a naopak ke každé pravdivé formuli poskytuje důkaz. Tedy pravdivost a dokazatelnost si odpovídají nejlepším možným způsobem. Takový systém je dobrým formálním rámcem matematiky.

Logika jako metamatematika Každý obor má nějaké své objekty zájmu a nějaký jazyk, kterým o nich mluví. Analýza se zabývá reálnými čísly, posloupnostmi, limitami, a vyjadřuje se o nich známým *epsilon-delta* formalismem. Lineární algebra se zabývá vektorovými prostory, lineárními operátory, maticemi, a používá k tomu opět jistý formální jazyk, dosti odlišný od jazyka analýzy. Čím se tedy zabývá matematická logika, jakožto samostatná disciplína?

Samotným jazykem a formálním aparátem matematiky. Samotné vyjadřovací a dokazovací prostředky jsou nyní předmětem zájmu. Formule, teorie, definice, věty, důkazy, modely, tedy běžné nástroje matematiky, stávají se v logice zkoumanými objekty. Budeme se například zajímat o vztah důsledku mezi formulemi, tak jako se třeba aritmetika zajímá o vztah dělitelnosti mezi čísly; budeme studovat důkazy, tak jako třeba algebra studuje polynomy. V tomto smyslu je matematická logika *metamatematikou*.

Zároveň je logika sama částí matematiky, a zkoumá objekty svého zájmu matematickými prostředky: bohatě čerpá z algebry, teorie množin, teoretické informatiky a topologie. Ostatní matematiku naopak obohacuje zkoumáním *úplnosti* či *rozhodnutelnosti* různých algebraických teorií, *bezespornosti* rozličných množinových či topologických principů, *složitosti* rozhodovacích a jiných algoritmů, atd. Přínos je oboustranný, vedl k mnoha hlubokým výsledkům, a zároveň obnažil i těžké, dosud nezodpovězené otázky.

Logika a teorie množin Popíšeme jazyk predikátové logiky *prvního řádu*, který umožňuje kvantifikovat jednotlivé objekty („každé prvočíslo větší než 2“), ale nikoli množiny objektů („každá omezená podmnožina“, „každá komutativní podgrupa“) či jejich vlastnosti („každá unární relace“). To je možné až v logikách vyšších řádů, jejichž jazyk obsahuje speciální symboly pro množiny nebo přirozená čísla. V jazyce vyššího řádu lze potom kvantifikovat systémy množin, systémy takových systémů, atd. Predikátová logika prvního řádu přesto plně postačuje k vybudování „běžné“ matematiky.

Je to možné prostřednictvím *teorie množin*, která vznikala ve stejné době jako formální logika. I její postavení v matematice je podobně dvojaké: je samostatnou disciplínou, která má svá vlastní témata a problémy, ale zároveň má i metamatematický význam. Brzy se totiž ukázalo, že na elementárním pojmu náležení do množiny lze „uvnitř“ teorie množin vybudovat ostatní pojmy běžné matematiky, jako *číslo*, *relace*, *funkce*, vystavět algebru jako studium relací a funkcí na množinách, matematickou analýzu, obecnou topologii jako studium

jistých systémů množin a funkcí na nich, funkcionální analýzu jako topologii na množinách funkcí, atd. Na objekty zkoumané v matematice (čísla, funkce, prostory, …), potažmo informatice (grafy, stromy, jazyky, databáze, …), můžeme pak hledět jako na množiny opatřené nějakou strukturou. Jazyk matematiky založené na teorii množin lze potom redukovat právě do jazyka predikátové logiky prvního řádu: kvantifikace jednotlivých objektů, totiž množin, umožňuje zároveň kvantifikaci množin objektů, které samy jsou opět jednotlivými objekty, totiž množinami. Základy teorie množin popíšeme v kapitole 3.

Co vynecháváme Nebudeme se zaobírat filozofickými kořeny logiky ani jejím historickým vývojem. Pomineme aristotelské sylogismy, stoickou školu starého Řecka i středověkou scholastiku. Náš zájem o logiku začíná na přelomu devatenáctého a dvacátého století, kdy se *matematická* logika spolu se souběžně vznikající teorií množin stává základem moderní matematiky.

Pomineme i všechny *neklasické logiky*: modální logiku, logiky s více než dvěma pravdivostními hodnotami, jazyky s nekonečně dlouhými formulemi či nestandardními kvantifikátory, fuzzy logiku, atd.

Kapitola 1

Výroková logika

Výroková logika zkoumá jazyk matematiky jen na úrovni *výrokových spojek*: $\neg$ negace, $\wedge$ konjunkce, $\vee$ disjunkce, $\to$ implikace, $\leftrightarrow$ ekvivalence. Výrokovými spojkami chceme ve formálním jazyce matematiky zachytit obraty přirozeného jazyka, vytvořené pomocí spojek *ne*, *a*, *nebo*, *pokud—pak*, *právě když*. Vnitřní strukturu jednotlivých promluv, které takto spojujeme, přitom výroková logika dále nezkoumá. V analogii s přirozeným jazykem můžeme takový pohled chápat jako analýzu souvětí, při které dále nerozebíráme jednotlivé věty.

1.1 Formule výrokové logiky

1.1.1 Definice. Buď $\mathcal{A}$ nějaká neprázdná množina, jejíž prvky nazveme *prvotní formule* nebo *výrokové proměnné*. Potom *výroková formule nad* $\mathcal{A}$ je každý řetězec symbolů získaný aplikací následujících pravidel v konečně mnoha krocích. (i) Každá prvotní formule z $\mathcal{A}$ je formule. (ii) Jsou-li φ, ψ formule, pak také $(\neg\varphi)$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$, $(\varphi \leftrightarrow \psi)$ jsou formule. Každý podřetězec dané formule, který je sám formulí, je její *podformule*.

Formule vytvořené pomocí výrokových spojek jako výše čteme postupně: „neplatí φ“, „φ a ψ“, „φ nebo ψ“, „pokud φ, pak ψ“ (případně „z φ plyne ψ“ či „φ implikuje ψ“) a „φ právě když ψ“ (též „φ je ekvivalentní s ψ“).

Povahou prvotních formulí se *výroková* logika nezabývá. Vycházíme z představy, že prvotní formule jsou nějaká základní (pravdivá či nepravdivá) tvrzení našeho jazyka, jako „všechna prvočísla jsou lichá,“ nebo nějakého formálního jazyka, jako $(\forall x)(\forall y)(xy = yx)$. Jejich vnitřní strukturu ale nebudeme zkoumat. Zajímáme se prozatím jen o způsob, jakým se pomocí výrokových spojek skládají do výrokových formulí. Budeme tedy jako prvotní formule používat třeba písmena $A, B, C \ldots, P, Q, R, \ldots$, případně s indexy $A_1, A_2, A_3, \ldots$ apod. Při studiu *predikátové* logiky, která zkoumá jazyk matematiky podrobněji, pak budeme zkoumat i vnitřní strukturu těchto prozatím nerozborných výrazů.

1.1.2 Příklad. $((A \wedge (\neg B)) \to (((\neg C) \vee D) \leftrightarrow (\neg E)))$ je výroková formule: A je prvotní formule; B je prvotní formule, takže $(\neg B)$ je formule, načež $(A \wedge (\neg B))$ je formule; zároveň C je prvotní formule, takže $(\neg C)$ je formule, načež $((\neg C) \vee D)$ je formule; dále E je prvotní formule, tedy $(\neg E)$ je formule, a

$(((\neg C) \vee D) \leftrightarrow (\neg E))$ je formule; tedy $((A \wedge (\neg B)) \rightarrow (((\neg C) \vee D) \leftrightarrow (\neg E)))$ je formule. Všechny předchozí formule jsou její podformule, zatímco řetězec $\rightarrow (((\neg C$ formulí není.

Všimněme si zásadní *konečnosti* formulí: jedná se o konečný řetězec, ve kterém se vyskytuje konečně mnoho prvotních formulí a konečně mnoho spojek. To je důležitá vlastnost klasické logiky, kterou se odlišuje od jiných možných logik, ve kterých se zkoumají nekonečně dlouhé konjunkce apod.

1.1.3 Cvičení. (a) Definice formule vyžaduje striktní uzávorkování; přísně vzato, $A \wedge B$ není formule, tou je teprve $(A \wedge B)$. Je ovšem běžnou praxí některé závorky vynechávat. Pravidla, podle kterých se některé spojky *váží* ke svým operandům silněji než jiné, a není proto nutné je závorkovat, jsou analogií *operator precedence* různých programovacích jazyků nebo aritmetických operací. Například $7*3+5$ obvykle čteme jako $(7*3)+5$, nikoli jako $7*(3+5)$. Zformulujte co nejpohodlnější pravidla takové *precedence* pro výrokové spojky.

(b) Přijměme obvyklou konvenci, podle které vazební síla výrokových spojek postupně klesá v pořadí $\neg, \{\wedge, \vee\}, \rightarrow, \leftrightarrow$; tedy konjunkce a disjunkce mají stejnou vazební sílu. Doplňte při této konvenci závorky do $A \wedge \neg B \rightarrow C \leftrightarrow D$. Ve formuli $((A \vee (B \wedge C)) \leftrightarrow (((\neg A) \wedge B) \vee ((\neg C) \rightarrow D)))$ naopak vynechte všechny závorky, které vynechat lze.

1.1.4 Definice. Je-li φ výroková formule, ve které se vyskytují jen výrokové proměnné $A_1, \ldots, A_n$, budeme někdy obšírněji psát $\varphi(A_1, \ldots, A_n)$. Jsou-li pak $\psi_1, \ldots, \psi_n$ libovolné jiné formule, označíme výrazem $\varphi(\psi_1, \ldots, \psi_n)$ formuli, která vznikne z formule φ nahrazením všech výskytů proměnné A_i formulí ψ_i, pro všechna $i \leq n$. Formuli $\varphi(\psi_1, \ldots, \psi_n)$ pak nazýváme *instancí* formule $\varphi(A_1, \ldots, A_n)$, která vznikla *substitucí* formulí ψ_i za výrokové proměnné A_i.

1.1.5 Cvičení. Jsou následující formule instancemi $(\neg Z \rightarrow Y) \vee (X \leftrightarrow Z)$? $(\neg A \rightarrow A) \vee (A \leftrightarrow A)$, $(\neg A \rightarrow Y) \vee (X \leftrightarrow A)$, $(\neg A \rightarrow Y) \vee (X \leftrightarrow \neg\neg A)$, $(\neg A \rightarrow Y) \vee (X \leftrightarrow Z)$, $(\neg\neg A \rightarrow B) \vee (C \leftrightarrow \neg A)$, $(A \rightarrow B) \vee (C \leftrightarrow \neg A)$, $(\neg Z \rightarrow Y) \wedge (X \leftrightarrow Z)$, $(\neg(A \vee B) \rightarrow (B \leftrightarrow C)) \vee ((B \wedge \neg A) \leftrightarrow (A \vee B))$, $(\neg(A \rightarrow B) \rightarrow (B \leftrightarrow C)) \vee ((B \wedge \neg A) \leftrightarrow (\neg A \vee B))$.

1.1.6 Cvičení. Syntax výrokových formulí zavedená v 1.1.1 se nazývá *infixní* — spojka je „uvnitř“ mezi operandy. Analogicky můžeme zavést *prefixní* a *postfixní* syntax, ve které jsou formulemi například výrazy $\vee\neg AB$ resp. $A\neg B\vee$.

(a) Zformulujte rekurzivní definici formule v prefixní a postfixní notaci.[1] Všimněte si, že prefix a postfix nepotřebuje žádné závorky. (b) Zapište formuli $((A \rightarrow B)\wedge(\neg((A\vee B) \leftrightarrow C)))$ v prefixu, formuli $\wedge \rightarrow AB\neg \leftrightarrow \vee ABC$ v postfixu a formuli $AB \rightarrow AB \vee C \leftrightarrow \neg\wedge$ v infixu.

1.1.7 Cvičení. (a) Syntaktický tvar výrokové formule lze zachytit *binárním stromem*. Nakreslete takový strom pro předchozí formuli. (b) Implementujte *parser* výrokových formulí, tj. program, který čte výrokové formule a rozeznává jejich strukturu; specielně tedy rozpozná, zda je vstup výrokovou formulí či nikoli. Pro jednoduchost použijte symboly `-` pro negaci, `+` pro disjunkci, `.` pro konjunkci, `>` pro implikaci, `=` pro ekvivalenci; jako proměnné rozeznávejte třeba latinské kapitálky. Například `(-((A>(B+C))=((A.B)>C)))` je potom zápisem formule $(\neg((A \rightarrow (B \vee C)) \leftrightarrow ((A \wedge B) \rightarrow C)))$. (c) Implementujte metody,

[1] Prefixní notace, zavedená J. Łukasiewiczem ([T], str. 39), se též nazývá „polská“.

které čtou a píší formule v prefixu, infixu a postfixu. (d) Implementujte metodu, která pro dvě dané formule rozpozná, zda první je instancí druhé.

1.2 Sémantika výrokové logiky

Výrokové formule jsou syntaktické útvary: řetězce jistého tvaru. Nyní popíšeme *sémantiku* výrokové logiky, která přiřazuje výrokovým formulím *pravdivostní hodnoty*. Ukážeme, jak se pravdivost formule odvíjí z pravdivosti podformulí, a zavedeme vztah *důsledku* mezi formulemi.

Pravdivostní ohodnocení Prvotní formule jsou z pohledu výrokové logiky nějaká blíže neurčená základní tvrzení. O každém z nich předpokládáme, že je buďto pravdivé nebo nepravdivé, ale nemáme žádný záměr (ani možnost) o jejich pravdivosti rozhodovat — musí tedy být dána „zvenčí".

1.2.1 Definice. Zobrazení v z množiny výrokových formulí do množiny $\{0,1\}$ se nazývá *pravdivostní ohodnocení*, pokud pro každé formule φ a ψ platí:[2]

$$\begin{aligned}
v(\neg\varphi) = 1 &\text{ právě když } v(\varphi) = 0\\
v(\varphi \wedge \psi) = 1 &\text{ právě když } v(\varphi) = 1 \text{ a } v(\psi) = 1\\
v(\varphi \vee \psi) = 1 &\text{ právě když } v(\varphi) = 1 \text{ nebo } v(\psi) = 1\\
v(\varphi \to \psi) = 1 &\text{ právě když } v(\varphi) = 0 \text{ nebo } v(\psi) = 1\\
v(\varphi \leftrightarrow \psi) = 1 &\text{ právě když } v(\varphi) = v(\psi)
\end{aligned}$$

Pro formuli φ je $v(\varphi)$ její *pravdivostní hodnota* při ohodnocení v. Je-li $v(\varphi) = 1$, řekneme, že φ je *splněna při ohodnocení* v, nebo že v *splňuje* φ.

Pravdivostní hodnota výrokové formule zřejmě závisí pouze na ohodnocení těch výrokových proměnných, které se v ní vyskytují. Toto očividné tvrzení nyní dokážeme, abychom předvedli *důkaz indukcí podle složitosti formule*.

1.2.2 Lemma. *Buď φ výroková formule, buďte $A_1, A_2, \ldots, A_n$ prvotní formule, které se ve φ vyskytují. Buďte v a w dvě pravdivostní ohodnocení, která se shodují na $A_i, i \le n$, to jest $v(A_i) = w(A_i)$ pro $i \le n$. Potom je $v(\varphi) = w(\varphi)$.*

Důkaz. (i) Pro prvotní formuli φ je tvrzení triviální. (ii) Je-li φ tvaru $\neg\psi$ a tvrzení platí pro ψ, pak $v(\varphi) = v(\neg\psi) = 1 - v(\psi) = 1 - w(\psi) = w(\neg\psi) = w(\varphi)$. (iii) Je-li φ tvaru $\psi \wedge \vartheta$ a tvrzení platí pro ψ a ϑ, pak $v(\varphi) = v(\psi \wedge \vartheta) = 1$ právě tehdy, když $v(\psi) = 1 = v(\vartheta)$, což je právě tehdy, když $w(\psi) = 1 = w(\vartheta)$, což je právě tehdy, když $w(\psi \wedge \vartheta) = w(\varphi) = 1$. (iv) Je-li φ tvaru $\psi \vee \vartheta$ a tvrzení platí pro ψ a ϑ, pak $v(\varphi) = v(\psi \vee \vartheta) = 1$ právě tehdy, když $v(\psi) = 1$ nebo $v(\vartheta) = 1$, což je právě tehdy, když $w(\psi) = 1$ nebo $w(\vartheta) = 1$, což je právě tehdy, když $w(\psi \vee \vartheta) = w(\varphi) = 1$. Zbylé případy pro formuli φ tvaru (v) implikace $\psi \to \vartheta$ a (vi) ekvivalence $\psi \leftrightarrow \vartheta$ přenecháváme čtenáři. □

[2] Definice pravdivostních hodnot v závislosti na syntaktickém tvaru formule je samozřejmě volena tak, aby zachycovala naše přirozené porozumění spojkám „a", „nebo" a dalším, tak jak je používáme v běžném jazyce. Disjunkci používáme v obvyklém „nevylučovacím" smyslu, tedy formule $A \vee B$ je splněna, pokud je splněna formule A nebo formule B, včetně případu, kdy jsou splněny obě zároveň. Sémantika spojky $\to$ se někdy nazývá *materiální implikace*: pravdivost formule $A \to B$ při daném ohodnocení znamená právě a jen tolik, že pokud je pravda A, je pravda i B — tím se netvrdí, že je mezi nimi nějaká „příčinná souvislost".

Skoro stejně se dokáže, že každé ohodnocení prvotních formulí se právě jedním způsobem rozšiřuje na ohodnocení všech formulí. To jest, pro každé zobrazení u z množiny prvotních formulí do $\{0,1\}$ existuje právě jedno pravdivostní ohodnocení, které se shoduje se zobrazením u na prvotních formulích.

Pravdivostní tabulky Právě zavedené pravdivostní ohodnocení výrokových spojek lze kompaktně vyjádřit pomocí následující *pravdivostní tabulky*.

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Podle 1.2.2 skutečně záleží jen na ohodnocení těch výrokových proměnných, které se v dané formuli vyskytují. Takových je konečně mnoho, jelikož formule je konečný výraz; stačí tedy uvážit konečně mnoho ohodnocení. Pravdivostní tabulku lze tedy rekurzivně sestavit pro každou výrokovou formuli.

1.2.3 Cvičení. Napište tabulku pravdivostních hodnot pro výrokovou formuli $(A \wedge \neg B) \to (\neg C \vee D)$. Kolik ohodnocení je potřeba vyšetřit?

1.2.4 Cvičení. Komise má tři členy, A, B, C. Napište formuli, která platí právě tehdy, když většina hlasuje pro.

1.2.5 Cvičení. Ukažte, že každá pravdivostní tabulka (o 2^n řádcích) je pravdivostní tabulkou nějaké výrokové formule nad n výrokovými proměnnými.

1.2.6 Cvičení. S pomocí svého výrokového parseru napište program, který vypíše pravdivostní tabulku dané formule.

Výrokové spojky odpovídají jistým *bitovým operací*: na vstupech 0 nebo 1 vrací hodnotu 0 nebo 1. Někteří přímo píší `~A`, `A&B`, `A|B` místo $\neg A$, $A \wedge B$, $A \vee B$. Zavedením těchto operací je na množinu $\{0,1\}$ uvalena jistá algebraická struktura. Ve skutečnosti jsme jednoduché vlastnosti této struktury již použili, když jsme v důkaze 1.2.2 psali stručně $v(\neg\psi) = 1 - v(\psi)$. Algebraickými souvislostmi výrokové logiky se budeme zabývat při studiu *Booleových algeber*.

Tautologie Pravdivost výrokové formule závisí na ohodnocení prvotních formulí, které se v ní vyskytují. U některých formulí však na ohodnocení nezáleží.

1.2.7 Definice. Výroková formule φ je (i) *tautologie*, je-li splněna při všech ohodnoceních; (ii) *splnitelná*, je-li splněna při nějakém ohodnocení; (iii) *kontradikce*, není-li splněna při žádném ohodnocení.

Například $A \to A$ je tautologie a $B \wedge \neg B$ je kontradikce; $A \to B$ je splnitelná, ale není tautologií ani kontradikcí. Každá tautologie je splnitelná, a kontradikce jsou právě nesplnitelné formule; negace tautologie je kontradikce a naopak.

Výrokové tautologie jsou pravdivé bez ohledu na to, o čem mluví: jsou pravdivé díky svému tvaru, neříkají nic specifického. Například *je-li každá posloupnost konvergentní, pak každá posloupnost je konvergentní* je jistě pravdivé tvrzení, totiž tautologie tvaru $A \to A$, ale o konvergenci neříká ve skutečnosti nic.

1.2.8 Cvičení. Ověřte, že následující ekvivalence (tzv. *deMorganovy zákony*) jsou tautologie: $\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$, $\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$.

1.2.9 Cvičení. Zjistěte, které z následujících formulí jsou tautologie, kontradikce, a splnitelné formule. $\neg A \to (A \to B)$; $A \to (A \to \neg A)$; $A \to (B \to \neg A)$; $\neg(A \to B) \to A$; $(A \to B) \vee (B \to A)$; $\neg A \wedge (B \to A)$; $(A \leftrightarrow B) \wedge (B \to \neg A)$; $((A \to B) \wedge (B \to C) \wedge (C \to D)) \to (A \to D)$.

1.2.10 Cvičení. Ověřte, že následující ekvivalence jsou tautologické.
$\neg\neg A \leftrightarrow A$; $(A \wedge A) \leftrightarrow A$; $(A \vee A) \leftrightarrow A$; $(A \wedge B) \leftrightarrow (B \wedge A)$; $(A \vee B) \leftrightarrow (B \vee A)$; $(A \wedge B) \wedge C \leftrightarrow A \wedge (B \wedge C)$; $(A \vee B) \vee C \leftrightarrow A \vee (B \vee C)$; $A \wedge (A \vee B) \leftrightarrow A$; $A \vee (A \wedge B) \leftrightarrow A$; $A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$; $A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$; $(A \to B) \leftrightarrow (\neg A \vee B)$; $A \to (B \wedge \neg B) \leftrightarrow \neg A$; $A \to (B \to C) \leftrightarrow (A \wedge B) \to C$; $(A \leftrightarrow (B \leftrightarrow C)) \leftrightarrow ((A \leftrightarrow B) \leftrightarrow C)$.

V dalším budeme běžně psát jen $(A \wedge B \wedge C \wedge D)$ místo formálně správného $(A \wedge (B \wedge (C \wedge D)))$ a podobně — podle předchozího cvičení na uzávorkování nezáleží. Říkáme, že spojky $\wedge$ a $\vee$ jsou *asociativní*. Spojka $\leftrightarrow$ je rovněž asociativní, nicméně zkratkou $A \leftrightarrow B \leftrightarrow C$ se obvykle myslí nikoli $A \leftrightarrow (B \leftrightarrow C)$ resp. $(A \leftrightarrow B) \leftrightarrow C$, nýbrž $(A \leftrightarrow B) \wedge (B \leftrightarrow C)$, a podobně pro delší řetězce.[3]

1.2.11 Cvičení. Které z následujících formulí jsou tautologie? $A \to (B \to A)$, $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$, $(\neg B \to \neg A) \to (A \to B)$.

1.2.12 Cvičení. Buď φ_1 formule $(A \to B) \to A$, a buď φ_{n+1} formule $\varphi_n \to A$. Pro která $n \in \mathbb{N}$ je φ_n tautologie?

1.2.13 Příklad. Pro některé formule lze o tautologičnosti rozhodnout efektivněji než vyšetřením všech možných ohodnocení.

(a) Formule $((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$ je sestavena jen z implikací. Tautologičnost takové formule lze ověřit vyšetřením „nejhoršího možného případu“. Totiž ohodnocení v, při kterém tato implikace neplatí, musí nutně splňovat $v(A \to (B \to C)) = 1$ a $v((A \to B) \to (A \to C)) = 0$. Tedy $v(A \to B) = 1$ a $v(A \to C) = 0$; a tedy $v(A) = 1$ a $v(C) = 0$; a tedy $v(B) = 1$. Při tomto ohodnocení je ale $v(A \to (B \to C)) = 0$, tedy celá formule je splněna.

(b) Formule sestavená z výrokových proměnných jen pomocí spojky $\leftrightarrow$ je tautologií právě tehdy, když počet výskytů každé výrokové proměnné je sudý.

1.2.14 Definice. Jsou-li φ, ψ dvě výrokové formule, řekneme, že ψ je *logickým důsledkem* φ, nebo že ψ *vyplývá* z φ, pokud každé ohodnocení, které splňuje formuli φ, splňuje také formuli ψ. V takovém případě píšeme[4] $\varphi \models \psi$. Pokud platí zároveň $\varphi \models \psi$ i $\psi \models \varphi$, řekneme, že formule φ a ψ jsou *logicky ekvivalentní*, a píšeme v takovém případě $\varphi \models\!\mid \psi$.

Pokud z formule φ vyplývá formule ψ, říkáme také, že φ je *postačující podmínkou* pro ψ, a naopak že ψ je *nutnou podmínkou* pro φ.

Snadno nahlédneme základní vlastnosti vztahu vyplývání: (i) $\varphi \models \psi$ právě tehdy, když $\varphi \to \psi$ je tautologie. (ii) $\varphi \models\!\mid \psi$ právě tehdy, když $\varphi \leftrightarrow \psi$ je tautologie. (iii) Každé dvě tautologie (i každé dvě kontradikce) jsou ekvivalentní. (iv) Je-li ϑ tautologie, pak pro každou formuli φ je $\varphi \models\!\mid (\varphi \wedge \vartheta)$. (v) Je-li ξ kontradikce, pak pro každou formuli φ je $\varphi \models\!\mid (\varphi \vee \xi)$.

[3] Například v analýze běžně říkáme, že $x > 0$ právě když $x^3 > 0$ právě když $x^5 > 0$.

[4] Specielně pro tautologii ψ odpovídá notace $\models \psi$ tomu, že ψ platí při každém ohodnocení.

1.2.15 Cvičení. (a) Je formule $B \vee C$ důsledkem $(A \vee B) \wedge (\neg A \vee C)$?
(b) Je $(A \to B) \wedge (B \to C) \wedge (C \to A)$ ekvivalentní s $A \leftrightarrow C$?

1.2.16 Cvičení. Pro každou dvojici formulí z následujících souborů rozhodněte, zda jedna je logickým důsledkem druhé, případně naopak:
(a) $(A \wedge B) \to C, (A \vee B) \to C, (A \to C) \wedge (B \to C), (A \to C) \vee (B \to C)$
(b) $A \to (B \wedge C), A \to (B \vee C), (A \to B) \wedge (A \to C), (A \to B) \vee (A \to C)$

1.2.17 Cvičení. Buďte φ a ψ výrokové formule, buď ϑ nějaká tautologie a ξ nějaká kontradikce. Pak $\varphi \models \varphi \vee \psi$, $\psi \models \varphi \vee \psi$, $\varphi \wedge \psi \models \varphi$, $\varphi \wedge \psi \models \psi$, $\models \xi \to \varphi$, $\models \varphi \to \vartheta$, $\models \varphi \wedge \vartheta \leftrightarrow \varphi$, $\models \varphi \vee \vartheta \leftrightarrow \vartheta$, $\models \varphi \wedge \xi \leftrightarrow \xi$, $\models \varphi \vee \xi \leftrightarrow \varphi$, $\models \vartheta \leftrightarrow \neg\xi$.

1.2.18 Cvičení. Kolik vzájemně neekvivalentních formulí existuje nad konečnou množinou výrokových proměnných $\{A_1, \ldots, A_n\}$? Návod: použijte 1.2.5.

1.2.19 Cvičení. Kolik existuje vzájemně neekvivalentních formulí φ v jazyce $\{A, B, C\}$, pro které je $\neg A \models \varphi \vee B$.

1.2.20 Cvičení. Buďte φ_0 a ψ_0 dvě logicky ekvivalentní formule. Je-li φ_0 podformule formule φ, a formule ψ vznikne z formule φ nahrazením všech výskytů podformule φ_0 ekvivalentní formulí ψ_0, jsou formule φ a ψ opět ekvivalentní.

1.2.21 Příklad. Je-li φ výroková tautologie, pak i každá její instance je tautologie. Je-li φ kontradikce, pak i každá její instance je kontradikce. Pokud φ není ani tautologií ani kontradikcí, pak pro každou pravdivostní tabulku existuje nějaká instance formule φ s touto pravdivostní tabulkou. (Tím je zesíleno tvrzení z 1.2.5.) Speciálně tedy nějaká instance formule φ je tautologií a nějaká jiná instance je kontradikcí.

Pokud totiž $\varphi(A_1, \ldots, A_n)$ není tautologií ani kontradikcí, pak pro nějaké ohodnocení f je $f(\varphi) = 0$ a pro nějaké ohodnocení t je $t(\varphi) = 1$. Pro každé $i \leq n$ buď $\psi_i(X)$ nějaká formule, která při hodnocení $v(X) = 0$ má hodnotu $v(\psi_i(X)) = f(A_i)$ a při ohodnocení $w(X) = 1$ má hodnotu $w(\psi_i(X)) = t(A_i)$. Potom instance $\varphi(\psi_1(X), \ldots, \psi_n(X))$ formule φ je ekvivalentní s formulí X. Je-li nyní dána libovolná pravdivostní tabulka, buď ϑ nějaká formule s předepsanými hodnotami (podle 1.2.5 taková formule existuje). Potom formule $\varphi(\psi_1(\vartheta), \ldots, \psi_n(\vartheta))$ je instancí formule φ a má předepsanout tabulku.

1.2.22 Cvičení. Najděte instanci formule $A_1 \to (A_2 \vee \neg A_3)$, která (i) je tautologií; (ii) je kontradikcí; (iii) má pravdivostní tabulku 00:1, 01:0, 10:0, 11:1.

1.3 Normální tvar

V tomto oddílu prozkoumáme vyjadřovací sílu jednotlivých výrokových spojek. Ukážeme, že jazyk výrokové logiky lze různými způsoby redukovat, a že každou výrokovou formuli lze ekvivalentně vyjádřit v tzv. *normálním tvaru*.

Vyjadřovací síla spojek Jazyk výrokové logiky od začátku budujeme pomocí spojek $\neg, \wedge, \vee, \to$ a $\leftrightarrow$. Tyto spojky zachycují nejběžnější obraty jazyka, kterým mluvíme, a chceme je tedy zachytit i ve formálním jazyce matematiky.

Další obraty přirozeného jazyka jsme se zatím nepokoušeli ve formálním jazyce zachytit. Připomeňme alespoň známou *vylučovací disjunkci*, tedy obrat „jedno nebo druhé, ale ne oboje.“ Pro ten můžeme zavést spojku $A \triangle B$ (*exclusive or* neboli *xor*) s hodnotami definovanými jako u formule $(A \wedge \neg B) \vee (B \wedge \neg A)$. Proč jsme ji do jazyka logiky nepřijali hned na začátku?

Takový jazyk by rozhodně byl redundantní: spojku $\triangle$ lze ekvivalentně vyjádřit pomocí ostatních spojek, a není tedy třeba ji zařazovat do základního jazyka — můžeme ji považovat za užitečnou zkratku, ale obejdeme se i bez ní. Podobně ovšem můžeme $A \leftrightarrow B$ považovat za zkratku pro $(A \to B) \wedge (B \to A)$.

Stejnou otázku si nyní můžeme položit i u ostatních spojek. Přirozený požadavek na úspornost jazyka nás vede ke zjištění, že některé spojky lze vyjádřit pomocí ostatních, a základní jazyk výrokové logiky lze tedy redukovat. Všechny klasické výrokové spojky lze například ekvivalentně vyjádřit jen spojkami $\neg$ a $\wedge$, neboť $(A \vee B) \leftrightarrow \neg(\neg A \wedge \neg B)$, $(A \to B) \leftrightarrow \neg(A \wedge \neg B)$ a $(A \leftrightarrow B) \leftrightarrow (\neg(A \wedge \neg B) \wedge \neg(B \wedge \neg A))$ jsou tautologie.

Řekneme, že množina $\mathcal{C}$ výrokových spojek je *univerzální*, pokud ke každé výrokové formuli existuje ekvivalentní formule obsahující výhradně spojky z $\mathcal{C}$. Právě jsme tedy ukázali, že $\{\neg, \wedge\}$ je univerzální množina spojek.

1.3.1 Cvičení. (a) Ukažte, že množiny spojek $\{\neg, \vee\}$ a $\{\neg, \to\}$ jsou univerzální. Redukcí jazyka výrokové logiky na spojky $\neg$ a $\to$ později zahájíme výstavbu formálního odvozovacího systému. (b) Uvažte binární výrokovou spojku $\bot$ (*false*), pro kterou pravdivostní hodnota formule $A \bot B$ je 0 při každém ohodnocení. Ukažte, že i množina spojek $\{\bot, \to\}$ je univerzální. (c) Napište formuli $(A \wedge \neg B) \to ((\neg C \vee D) \leftrightarrow \neg E)$ ekvivalentně v jazycích $\{\neg, \wedge\}, \{\neg, \vee\}$ a $\{\neg, \to\}$. (d) Ukažte, že $\to$ nelze ekvivalentně vyjádřit spojkami $\neg$ a $\leftrightarrow$. Tedy množina $\{\neg, \leftrightarrow\}$ není univerzální. (e) Ukažte, že výroková formule, která vznikne jen použitím $\wedge$ a $\vee$, není ani tautologie ani kontradikce. Tedy množina spojek $\{\wedge, \vee\}$ není univerzální. (f) Ukažte, že ani $\{\wedge, \vee, \to, \leftrightarrow\}$ není univerzální.

1.3.2 Cvičení. Krajním případem univerzální množiny spojek je situace, kdy všechny formule lze vyjádřit pomocí jediné *univerzální spojky*. (a) Ukažte, že spojky $A \uparrow B$ (*nand*) a $A \downarrow B$ (*nor*), s hodnotami definovanými jako u formulí $\neg(A \wedge B)$ a $\neg(A \vee B)$, jsou univerzální. (b) Pro která ohodnocení je splněna formule $(((((((A \uparrow B) \downarrow C) \uparrow D) \downarrow E) \uparrow F) \downarrow G) \uparrow H)$?

1.3.3 Lemma. *$\uparrow$ a $\downarrow$ jsou jediné univerzální spojky.*

Důkaz. Buď $A \diamond B$ univerzální spojka. Pak při ohodnocení $u(A) = 1 = u(B)$ musí být $u(A \diamond B) = 0$: kdyby totiž $u(A \diamond B) = 1$, pak by už každá formule sestavená z A, B jen pomocí $\diamond$ měla při tomto ohodnocení hodnotu 1 (což se snadno ukáže indukcí); pak by ovšem $\diamond$ nemohla být univerzální. Podobně se ukáže, že při ohodnocení $v(A) = 0 = v(B)$ je nutně $v(A \diamond B) = 1$. Všimněme si, že obě univerzální spojky $\uparrow$ a $\downarrow$ tyto vlastnosti skutečně mají. Zbývá vyšetřit hodnotu $A \diamond B$ při ohodnoceních $w(A) = 0, w(B) = 1$ a $z(A) = 1, z(B) = 0$.

V úvahu přicházejí čtyři možnosti. Jejich probírkou zjistíme, že spojka $A \diamond B$ se celkem chová buďto jako $\uparrow$ nebo $\downarrow$, čímž je důkaz hotov, nebo jako $\neg A$ či $\neg B$, o kterých se ale snadno přesvědčíme, že nejsou univerzální. □

Jako důsledek získáváme, že univerzální množiny $\{\neg, \wedge\}$, $\{\neg, \vee\}$, $\{\neg, \to\}$, $\{\bot, \to\}$ z předchozího jsou navíc *minimální*, tj. nelze je dále redukovat.

1.3.4 Cvičení. Po zavedení spojek $\triangle, \uparrow, \downarrow, \perp$ se můžeme ptát, co všechno máme považovat za spojku. Abstraktně lze na binární výrokovou spojku pohlížet jako na nějaké zobrazení z $\{0,1\} \times \{0,1\}$ do $\{0,1\}$. Takových „spojek" potom existuje tolik, kolik je všech zobrazení z 2^2 do 2, tedy $2^{2^2} = 16$. Napište jejich pravdivostní tabulky a popište je pomocí dosavadních spojek.

Normální tvar formulí Viděli jsme, že každou výrokovou formuli lze ekvivalentně vyjádřit v různě redukovaných jazycích. Nyní budeme navíc požadovat, aby se povolené spojky uplatnily při výstavbě formule ve zvoleném pořadí.

1.3.5 Definice. Výroková formule je

(i) *literál*, pokud je prvotní formulí nebo negací prvotní formule;

(ii) *implikant*, pokud je konjunkcí literálů;

(iii) *klauzule*, pokud je disjunkcí literálů;

(iv) v *disjunktivním normálním tvaru* (DNT), pokud je disjunkcí implikantů;

(v) v *konjunktivním normálním tvaru* (KNT), pokud je konjunkcí klauzulí.

(vi) v *úplném* disjunktivním (konjunktivním) tvaru, pokud se všech jeho implikantech (klauzulích) vyskytují všechny použité výrokové proměnné. Takové implikanty (klauzule) jsou potom jeho *mintermy* (*maxtermy*).

Tedy například $\neg A, B, \neg C, \neg D$ jsou literály, $(A \wedge \neg B \wedge \neg C)$ je implikant, $(B \vee \neg C \vee D)$ je klauzule, $(A \wedge \neg B) \vee (\neg A \wedge C)$ je v disjunktivním normálním tvaru, a $(B \vee \neg C) \wedge (A \vee C)$ je v konjunktivním tvaru; formule $(A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge C)$ je v úplném disjunktivním tvaru, který sestává ze dvou mintermů.[5]

Bez újmy na obecnosti můžeme zároveň požadovat, aby normální tvar neobsahoval žádné tautologie či kontradikce, jako třeba $(A \vee \neg B) \wedge (C \vee \neg C)$ nebo $(A \wedge \neg B) \vee (C \wedge \neg C)$, žádné duplicitní implikanty či klauzule, a žádné duplicitní literály jako třeba $A \wedge \neg B \wedge A$ nebo $B \vee B \vee \neg C$.

Směřujeme k existenční větě, podle které má každá výroková formule ekvivalentní vyjádření v úplném normálním tvaru. Popíšeme nejprve několik standardních obratů, pomocí kterých lze k takovému tvaru dospět.

1.3.6 Cvičení. (a) Každou výrokovou formuli lze ekvivalentně vyjádřit v takovém tvaru, kde se negace vyskytují pouze v literálech. To lze dokázat rekurzivním použitím těchto tautologií: $\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$, $\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$, $\neg(A \to B) \leftrightarrow (A \wedge \neg B)$, $\neg(A \leftrightarrow B) \leftrightarrow (A \wedge \neg B) \vee (B \wedge \neg A)$, $\neg\neg A \leftrightarrow A$. Přepisu formule do tohoto tvaru budeme v dalším stručně říkat *propagace negace* k literálům. (b) Propagujte negaci k literálům ve formulích $\neg(A \to (B \to C))$; $\neg(A \leftrightarrow (B \wedge (C \to D)))$; $\neg(A \vee (B \to (C \wedge D)))$.

Pro zkrácení některých dalších zápisů přijmeme následující úmluvu: zápisem $\varphi \rightleftharpoons (\psi \wedge \vartheta)$ myslíme, že φ je formule tvaru $(\psi \wedge \vartheta)$. Tedy $\rightleftharpoons$ je symbol *metajazyka*, kterým zde hovoříme *o* formulích, nikoli nový symbol jazyka výrokové logiky.

[5] Názvy *minterm* a *maxterm* používáme proto, že takové formule odpovídají minimálním a maximálním prvkům jistých uspořádání, jak uvidíme v kapitole o Booleových algebrách.

V několika předchozích situacích figurovaly spojky $\wedge$ a $\vee$ ve velmi podobných, navzájem duálních rolích. Předvedeme nyní několik projevů této duality, včetně duality mezi konjunktivním a disjunktivním tvarem.

1.3.7 Lemma (princip duality)**.** *Pro výrokovou formuli φ, která obsahuje pouze spojky $\neg, \wedge, \vee$, buď φ^* formule, která vznikne z φ nahrazením každého výskytu spojky $\wedge$ spojkou $\vee$, nahrazením každého výskytu spojky $\vee$ spojkou $\wedge$, a nahrazením každého literálu opačným literálem. Pak φ^* je ekvivalentní $\neg\varphi$.*

Důkaz. Je-li φ sama literálem, je tvrzení triviální. Pokud tvrzení platí pro formule ψ a ϑ, pak pro formule z nich složené máme: $(\neg\psi)^* \rightleftharpoons \neg(\psi^*) \models \neg(\neg\psi)$ pro negaci, $(\psi \wedge \vartheta)^* \rightleftharpoons (\psi^* \vee \vartheta^*) \models (\neg\psi \vee \neg\vartheta) \models \neg(\psi \wedge \vartheta)$ pro konjunkci a $(\psi \vee \vartheta)^* \rightleftharpoons (\psi^* \wedge \vartheta^*) \models (\neg\psi \wedge \neg\vartheta) \models \neg(\psi \vee \vartheta)$ pro disjunkci. □

1.3.8 Cvičení. Buď φ výroková formule, a buďte φ_d a φ_k formule v disjunktivním resp. konjunktivním tvaru, pro které $\varphi \models \varphi_d \models \varphi_k$. Potom $(\varphi_d)^*$ a $(\varphi_k)^*$ je v konjunktivním resp. disjunktivním normálním tvaru, a $\neg\varphi \models (\varphi_d)^* \models (\varphi_k)^*$.

1.3.9 Cvičení. Ukažte indukcí, že pro formuli v (úplném) konjunktivním tvaru vede důsledná distribuce jednotlivých klauzulí (maxtermů) na ekvivalentní formuli v (úplném) disjunktivním tvaru. Například z formule $(A \vee \neg B) \wedge (\neg C \vee D)$ takto vznikne ekvivalentní formule $(A \wedge \neg C) \vee (A \wedge D) \vee (\neg B \wedge \neg C) \vee (\neg B \wedge D)$. Analogické tvrzení platí pro distribuci (úplného) disjunktního tvaru.

1.3.10 Věta (o normálním tvaru)**.** *Každou výrokovou formuli lze ekvivalentně vyjádřit v úplném disjunktivním a v úplném konjunktivním normálním tvaru. To jest, pro formuli φ existuje formule φ_d v úplném disjunktivním tvaru a formule φ_k v úplném konjunktivním tvaru tak, že $\varphi \models \varphi_d$ a $\varphi \models \varphi_k$.*

Důkaz. Je-li formule φ literálem, je v úplném normálním tvaru. Je-li složena z formulí ψ a ϑ, najdeme její disjunktivní normální tvar φ_d indukcí za předpokladu, že již známe $\psi_d, \psi_k, \vartheta_d, \vartheta_k$. Postup nalezení φ_k je díky dualitě analogický.

(¬) Pro $\varphi \rightleftharpoons (\neg\psi)$ je $\varphi \models (\neg\psi)_k \models (\psi_k)^* \rightleftharpoons \varphi_d$ podle 1.3.8.

(∨) Pro $\varphi \rightleftharpoons (\psi \vee \vartheta)$ je $\varphi \models (\psi_d \vee \vartheta_d) \rightleftharpoons \varphi_d$.

(∧) Pro $\varphi \rightleftharpoons (\psi \wedge \vartheta)$ získáme φ_d distribucí $\psi_k \wedge \vartheta_k$ podle 1.3.9.

(→) Pro $\varphi \rightleftharpoons (\psi \to \vartheta)$ je $\varphi \models \neg\psi \vee \vartheta \models (\psi_k)^* \vee \vartheta_d \rightleftharpoons \varphi_d$ podle (¬) a (∨).

(↔) Pro $\varphi \rightleftharpoons (\psi \leftrightarrow \vartheta)$ je $\varphi \models (\psi \wedge \vartheta)_d \vee (\neg\psi \wedge \neg\vartheta)_d \rightleftharpoons \varphi_d$ dle (¬), (∨), (∧).

Od normálního tvaru k úplnému normálnímu tvaru pak dojdeme použitím tautologií $A \leftrightarrow (A \wedge X) \vee (A \wedge \neg X)$ a $A \leftrightarrow (A \vee X) \wedge (A \vee \neg X)$, pokud v některém z implikantů či klauzulí chybí proměnná X. □

1.3.11 Příklad. Právě podaný důkaz je *konstruktivní*, tj. dává přímo návod, jak normální tvar rekurzivně nalézt. Předvedeme tento postup na následující formuli. Ze vznikající disjunkce průběžně odstraňujeme kontradikce a duplicity; nakonec doplníme chybějící literály.

$(A \wedge \neg(B \to C)) \leftrightarrow (D \to C)$
$((A \wedge \neg(B \to C)) \wedge (D \to C)) \vee (\neg(A \wedge \neg(B \to C)) \wedge \neg(D \to C))$
$((A \wedge B \wedge \neg C) \wedge (\neg D \vee C)) \vee ((\neg A \vee \neg B \vee C) \wedge (D \wedge \neg C))$
$(A \wedge B \wedge \neg C \wedge \neg D) \vee (\neg A \wedge D \wedge \neg C) \vee (\neg B \wedge D \wedge \neg C)$
$(A \wedge B \wedge \neg C \wedge \neg D) \vee (\neg A \wedge B \wedge D \wedge \neg C) \vee (\neg A \wedge \neg B \wedge D \wedge \neg C) \vee (A \wedge \neg B \wedge D \wedge \neg C)$

1.3.12 Cvičení. (a) Najděte úplný disjunktivní a úplný konjunktivní tvar pro $A \to (B \wedge C)$, $A \to (B \vee C)$, $(A \wedge B) \to C$, $(A \vee B) \to C$, $(A \to B) \wedge (A \to C)$, $(A \to B) \vee (A \to C)$, $(A \to C) \wedge (B \to C)$, $(A \to C) \vee (B \to C)$. (b) Napište pravdivostní tabulky těchto formulí: mintermy úplného disjunktivního tvaru odpovídají pravdivostním ohodnocením, při kterých je formule splněna. To se snadno nahlédne přímo z definice splňování pro disjunkci, konjunkci a negaci. Úplný disjunktivní normální tvar dané formule tedy nese stejnou informaci, jako její pravdivostní tabulka.[6] Kterým ohodnocením odpovídají maxtermy úplného konjunktivního tvaru? (d) Úplný disjunktivní i konjunktivní tvar dané formule je jednoznačný až na pořadí mintermů/maxtermů a literálů.

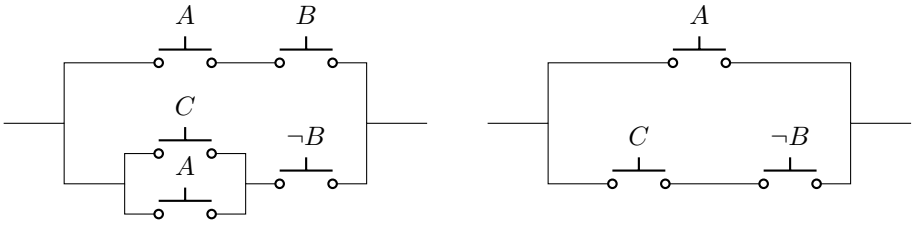
1.3.13 Cvičení. (a) Buďte φ a ψ výrokové formule, buďte φ_d a ψ_d jejich úplné disjunktivní tvary. Potom $\varphi \models \psi$ platí právě tehdy, když každý minterm z φ_d je obsažen i v ψ_d. Zformulujte a dokažte duální tvrzení pro konjunktivní tvary. (b) Je $A \to (\neg B \wedge C) \models B \to (A \to C)$, $\neg(A \to (B \vee \neg C)) \models \neg((A \vee B) \to \neg C)$, $(\neg(E \to D)) \wedge A \models (A \to (D \vee \neg E)) \to (C \wedge \neg(A \to B))$?

Minimalizace Popsali jsme způsob, jak dojít k úplnému normálnímu tvaru výrokové formule. Nyní ukážeme na příkladech, jak normální tvar naopak minimalizovat. Minimální normální tvar je užitečný v technických aplikacích.

1.3.14 Příklad. Následující formule je v úplném disjunktivním tvaru: $(A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C)$. Některé dvojice mintermů se navzájem liší právě v jednom literálu, například $(A \wedge \neg B \wedge \neg C)$ a $(\neg A \wedge \neg B \wedge \neg C)$. Každý takový pár lze s použitím distribučního zákona zapsat jako jeden kratší implikant, v tomto případě $(\neg B \wedge \neg C)$. Podobně místo $(A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C)$ můžeme ekvivalentně psát $(B \wedge \neg C)$. Přepíšeme-li potom $(\neg B \wedge \neg C) \vee (B \wedge \neg C)$ na $\neg C$, dojdeme nakonec k formuli $(A \wedge B) \vee \neg C$, kterou už nelze popsaným způsobem dále zjednodušit.

Mintermy s opačnými literály můžeme ovšem sjednocovat i jinak: sloučíme-li například první dva skrze $A, \neg A$ a druhé dva skrze $C, \neg C$, dojdeme k formuli $(\neg B \wedge \neg C) \vee (A \wedge B) \vee (\neg A \wedge B \wedge \neg C)$. Ani tu už nelze popsaným způsobem dále zjednodušit, ale předchozí varianta je stručnější: obsahuje dva implikanty místo tří, navíc kratší. Záleží tedy na tom, jak přesně mintermy slučujeme.

1.3.15 Příklad. Elektrický obvod lze zachytit následujícím diagramem.



Obvod je sestaven jen z přepínačů,[7] a každým z nich teče proud právě když je splněna příslušná formule. Obvodem vlevo pak teče proud právě když platí $(A \wedge B) \vee ((A \vee C) \wedge \neg B)$. Tu lze podobně jako výše zkrátit na formuli $A \vee (C \wedge \neg B)$, která odpovídá obvodu vpravo; ten je jednodušší, přitom funkčně ekvivalentní.

[6] Tím se trivializuje 1.2.5

[7] C. E. Shannon, *A Symbolic Analysis of Relay and Switching Circuits*, Trans. AIEE 57:12 (1938), 713–723

Řekneme, že disjunktivní normální tvar je *minimální*, pokud žádný ekvivalentní normální tvar nemá méně implikantů nebo méně literálů.

Výroková formule může mít více minimálních tvarů, a všechny lze odhalit hrubou silou: normálních tvarů nad konečně mnoha prvotními formulemi je konečně mnoho, a systematickou probírkou všech můžeme zjistit, které z nich jsou minimální. Takový postup by ale zřejmě nebyl příliš efektivní.

1.3.16 Příklad (Quine-McCluskey)**.** Předvedeme algoritmus,[8] který slučuje mintermy úplného disjunktivního tvaru všemi možnými způsoby; tím odhalí nejkratší možné implikanty, a z nich sestaví minimální disjunktivní tvar.

Místo se samotnými mintermy pracuje algoritmus s jejich bitovými kódy, při nějaké zvolené korespondenci mezi bitovými pozicemi a symboly pro prvotních formule. Například `1101` kóduje $A \wedge B \wedge \neg C \wedge D$. Úplný disjunktivní tvar lze potom zadat výčtem přítomných mintermů, resp. jejich dekadických kódů; např. výraz $\sum m(0, 2, 5, 6, 7, 8, 10, 12, 13, 14, 15)$ určuje úplný normání disjunktivní tvar formule se čtyřmi proměnnými (řekněme A, B, C, D), který obsahuje minterm $\neg A \wedge \neg B \wedge \neg C \wedge \neg D$, kódovaný jako `0000`=0, a deset dalších mintermů. Předvedeme Quine-McCluskey algoritmus na této formuli.

Najít všechny páry mintermů, které se liší právě jedním opačným literálem, znamená najít všechny páry odpovídajících čtyřbitových kódů, které se liší právě na jedné pozici. K tomu nejprve seskup kódy mintermů podle počtu pozitivních bitů — takové páry potom vzejdou vždy ze sousedních skupin. To je provedeno ve druhém sloupci tabulky níže.[9]

Nyní slučuj mintermy všemi možnými způsoby. Při hledání kandidátů na párování s daným kódem stačí vždy projít bezprostředně následující skupinu. Například jedinými kandidáty na sloučení s `0000` jsou `0010` a `1000`. Označ takto vzniklé dvojice kódem se znakem – na příslušné bitové pozici. Například sloučením `0000` a `0010` je `00-0`; v normálním disjunktivním tvaru se disjunkce $(\neg A \wedge \neg B \wedge \neg C \wedge \neg D) \vee (\neg A \wedge \neg B \wedge C \wedge \neg D)$ zkracuje na ekvivalentní podformuli $(\neg A \wedge \neg B \wedge \neg D)$. Všechny takové páry jsou sepsány v dalším sloupci. Nadále platí, že kódy z každé skupiny mají předepsaný počet pozitivních bitů.

[8] E. J. McCluskey, *Minimization of Boolean Functions*, BSTJ 35 (1956), 1417–1444

[9] Uvádíme pro názornost i dekadické hodnoty, bez kterých se samotný algoritmus obejde.

0	0000 (0)	00-0 (0,2)	-0-0 (0,2,8,10)
		-000 (0,8)	-0-0 (0,8,2,10)
1	0010 (2)	-010 (2,10)	--10 (2,10,6,14)
		0-10 (2,6)	--10 (2,6,10,14)
	1000 (8)	1-00 (8,12)	1--0 (8,12,10,14)
		10-0 (8,10)	1--0 (8,10,12,14)
2	0101 (5)	-101 (5,13)	-1-1 (5,13,7,15)
		01-1 (5,7)	-1-1 (5,7,13,15)
	0110 (6)	-110 (6,14)	-11- (6,14,7,15)
		011- (6,7)	-11- (6,7,14,15)
	1010 (10)	1-10 (10,14)	
	1100 (12)	11-0 (12,14)	11-- (12,14,13,15)
		110- (12,13)	11-- (12,13,14,15)
3	0111 (7)	-111 (7,15)	
	1101 (13)	11-1 (13,15)	
	1110 (14)	111- (14,15)	
4	1111 (15)		

Stejně potom slučuj dvojice do čtveřic, například `00-0` a `10-0` do `-0-0`; disjunkce $(\neg A \wedge \neg B \wedge \neg D) \vee (A \wedge \neg B \wedge \neg D)$ se tak zkrátí na $(\neg B \wedge \neg D)$. Některé takové implikanty mohou vzniknout více způsoby, například `-000` a `-010` se rovněž slučují na `-0-0`. Takové duplicity však můžeme ignorovat: v normálním tvaru odpovídají sloučení týchž implikantů v jiném pořadí.

Pokračuj stejným způsobem dále a sdružuj čtveřice do osmic atd., dokud nedojdeš ke kódům, které už dále slučovat nelze. To jsou *minimální implikanty* zadané formule.[10] Její minimální tvar bude sestávat výhradně z takových: snadno se nahlédne, že jinak by nebyl minimální.

Tím končí první fáze algoritmu. Ve druhé fázi se rozhodne, které z minimálních implikantů zařadit do výsledného minimálním tvaru. Disjunkce všech je jistě disjunktivní tvar ekvivalentní se zadanou formulí, ale není nutně minimální: některé implikanty *pokrývají* stejné mintermy.

	0	2	5	6	7	8	10	12	13	14	15
-0-0	*	*				*	*				
--10		*		*			*			*	
1--0						*	*	*		*	
-1-1			*		*				*		*
-11-				*	*					*	*
11--								*	*	*	*

Některé mintermy pokrývá jediný implikant, například `-0-0` jako jediný pokrývá `0=0000`, a `-1-1` jako jediný pokrývá `5=0101`. Takové implikanty jsou *esenciální*: ve hledaném minimálním tvaru musí figurovat. To znamená, že každý minimální tvar bude obsahovat $(\neg B \wedge \neg D) \vee (B \wedge D)$. Tím jsou pokryty mintermy `0,2,5,7,8,10,13,15`. Zbývá najít nějaké minimální pokrytí ostatních.

[10] V našem příkladě vznikly všechny minimální implikanty sloučením právě čtyř původních mintermů — obecně mohou být sloučením různých podmnožin různých velikostí 2^k.

	6	12	14
--10	*		*
1--0		*	*
-11-	*		*
11--		*	*

Pokrytí zbylých mintermů nejsou navzájem nezávislá: každý implikant, který pokrývá 6 nebo 12, pokrývá zároveň 14. Říkáme, že mintermy 6 a 12 dominují minterm 14. Při hledání minimálního pokrytí to znamená, že minterm 14 pokrývat nemusíme, stačí pokrýt 6 a 12. Přitom každý z mintermů 6 a 12, který je pokryt implikantem --10, je zároveň pokryt implikantem -11-, a také naopak. Stejný vztah platí i mezi implikanty 1--0 a 11--. Říkáme, že takové implikanty se dominují navzájem. Při hledání minimálního pokrytí stačí z každé množiny vzájemně se dominujících implikantů zvolit jeden; zvolme třeba první. Odpovídajícím minimálním tvarem je pak $(\neg B \wedge \neg D) \vee (B \wedge D) \vee (C \wedge \neg D) \vee (A \wedge \neg D)$.

V krajním případě, kdy všechny implikanty jsou esenciální, je minimální tvar určen jednoznačně. Zde naopak volbami vzájemně se dominujících implikantů dojdeme k minimálním tvarům $(\neg B \wedge \neg D) \vee (B \wedge D) \vee (C \wedge \neg D) \vee (A \wedge B)$, $(\neg B \wedge \neg D) \vee (B \wedge D) \vee (B \wedge C) \vee (A \wedge \neg D)$, $(\neg B \wedge \neg D) \vee (B \wedge D) \vee (B \wedge C) \vee (A \wedge B)$.

1.3.17 Cvičení (Karnaughovy mapy)**.** Úplný disjunktivní tvar lze zachytit bitmapou, a je-li vhodně indexována bitovými posloupnostmi jako výše, lze ji použít k minimalizaci normálního tvaru.[11] Vhodná indexace spočívá v tom, že indexy sousedních polí se liší právě v jednom bitu. To je pro čtyři proměnné možné například takto:

0000	0001	0011	0010
0100	0101	0111	0110
1100	1101	1111	1110
1000	1001	1011	1010

Pozice vyplněné hodnotou **1** pak odpovídají mintermům úplného disjunktivního tvaru, tedy splňujícím ohodnocením. Karnaughova mapa tedy nese stejnou informaci jako pravdivostní tabulka. Například formule z 1.3.16 má při zvolené indexaci tuto mapu:

1	**0**	**0**	**1**
0	**1**	**1**	**1**
1	**1**	**1**	**1**
1	**0**	**0**	**1**

Slučování mintermů, které jsme prováděli v 1.3.16, odpovídá v mapě slučování sousedních pozic. Přitom „sousední“ jsou i pozice 0=0000 a 2=0010 nebo 8=1000 a 10=1010, totiž liší se právě v jednom bitu. Implikanty odhalené předchozím algoritmem odpovídají potom v mapě maximálním blokům sousedních pozitivních polí velikosti 2^k; například implikantu --10 odpovídá pravý sloupec a implikantu -0-0 čtveřice rohových polí. Minimální tvary pak odpovídají minimálním pokrytím pomocí takových maximálních bloků.

(a) Odhalte v mapě výše implikanty formule z 1.3.16. Všimněte si zvláště pozice eseciálních implikantů. (b) Nakreslete Karnaughovu mapu formule z

[11] M. Karnaugh, *The map method for synthesis of combinatorial logical circuits*, Trans. AIEE 72 (1953), 593–598

příkladu 1.3.14. Všimněte si, že implikanty mají různé velikosti. Napište minimální tvar. (c) Najděte všechny minimální disjunktivní tvary výrokové formule s mapou $\sum m(0,1,5,7,8,10,14,15)$. (d) Popište nějakou indexaci Karnaughovy mapy pro pět proměnných, a obecně pro daný konečný počet proměnných.

1.3.18 Cvičení. Je-li φ minimální disjunktivní tvar formule ψ, je φ^* minimální konjunktivní tvar formule $\neg\psi$, a podobně naopak.

1.4 Splnitelnost

V tomto oddíle se budeme zabývat splnitelností výrokových formulí a výrokových teorií. Otázka splnitelnosti formule je pro nás příležitostí ilustrovat souvislost matematické logiky a teoretické informatiky pomocí známého SAT problému. Popíšeme rezoluční metodu, která efektivně rozhoduje o splnitelnosti konečných výrokových teorií, a dokážeme větu o kompaktnosti, která popisuje splnitelnost nekonečných teorií.

SAT Problem Vyplnit pravdivostní tabulku je procedura,[12] která rozhoduje o splnitelnosti výrokových formulí. Pravdivostní tabulka pro formuli s n výrokovými proměnnými však má 2^n řádků, takže metoda pravdivostních tabulek není příliš efektivní: složitost výpočtu roste exponenciálně rychle vůči velikosti vstupu. Je přirozené se ptát, zda existuje nějaký efektivnější způsob.

Rozhodnout o každé dané výrokové formuli, zda je splnitelná, je rozhodovací úloha známá jako SAT; algoritmus, který tuto úlohu řeší, je pak SAT *solver*. Zatím známe dva: vyplnění tabulky a nalezení úplného normálního tvaru. Nyní se ptáme, jak efektivní může SAT solver být. Je zde patrný přesun zájmu: zatímco *řešitelnost* otázky po splnitelnosti výrokové formule je triviální, otázka po *složitosti řešení* je zajímavá.

Dá se ukázat, že SAT je NP-*úplný* problém.[13] Třídu složitosti NP tvoří úlohy, které lze v polynomiálním čase řešit nedeterministickým Turingovým strojem. Cookova věta říká, že každou takovou úlohu lze pomocí deterministického Turingova stroje v polynomiálním čase *redukovat* na SAT; řešení SAT pak dává i řešení původní úlohy. Sám SAT tedy musí být výpočetně velice náročná úloha: alespoň tak náročná, jako kterákoli úloha z NP.

Třídu složitosti P tvoří úlohy, které lze algoritmicky řešit pomocí deterministického Turingova stroje v polynomiálním čase. Je tedy $P \subseteq NP$. Otázka po rovnosti $P = NP$ je známa pod názvem *PNP Problem*, a je všeobecně považována za jednu z nejdůležitějších otevřených otázek informatiky.

Důsledkem Cookovy věty je, že *pokud* existuje deterministický, polynomiálně složitý SAT solver (tj. pokud SAT patří do P), pak už takové řešení existuje i pro všechny ostatní úlohy z NP, a tedy $P = NP$. To znamená, že PNP Problem lze redukovat na existenci deterministického polynomiálního SAT solveru.

Rezoluční metoda Rozšíříme nyní základní pojmy výrokové logiky z jednotlivých formulí i na množiny formulí, tj. výrokové *teorie*, a popíšeme algoritmus,

[12] Odvoláváme se zatím jen na vágní představu *procedury* či *algoritmu*; zaujatý čtenář takový algoritmus implementoval ve cvičení 1.2.6.

[13] S. A. Cook, *The complexity of theorem-proving procedures*, Proc. of the Third ACM Symposium on Theory of Computing (1971), 151–158

který rozhoduje o splnitelnosti konečných výrokových teorií. Přitom splnit konečnou teorii $\varphi_1, \ldots, \varphi_n$ znamená právě tolik, jako splnit konjunkci $\varphi_1 \wedge \cdots \wedge \varphi_n$. Rezoluční metoda je tedy další SAT solver.

1.4.1 Definice. *Výroková teorie* je jakákoli množina výrokových formulí. Teorie T je ***splněna*** při ohodnocení v, pokud v splňuje všechny formule z T. Pokud takové ohodnocení existuje, řekneme, že T je ***splnitelná***.

1.4.2 Definice. Buď T výroková teorie a φ výroková formule. Řekneme, že formule φ ***vyplývá*** z teorie T, nebo že je ***důsledkem*** teorie T, a píšeme $T \models \varphi$, pokud každé ohodnocení splňující T splňuje také formuli φ. Obecněji, jsou-li S a T dvě výrokové teorie, řekneme, že T ***vyplývá*** z S, a píšeme $S \models T$, pokud každé ohodnocení splňující teorii S splňuje také teorii T. Pokud platí zároveň $S \models T$ i $T \models S$, řekneme, že teorie S a T jsou ***ekvivalentní***, a píšeme $S \models\!\mid T$.

Pro teorii T a formuli φ je zřejmě $T \models \varphi$ právě tehdy, když $T \cup \{\neg\varphi\}$ není splnitelná. Teorie S a T jsou ekvivalentní právě tehdy, když pro každou formuli φ je $T \models \varphi$ právě když $S \models \varphi$. Jinými slovy, ekvivalentní jsou takové teorie, které mají tytéž důsledky.

1.4.3 Cvičení. Jsou výrokové teorie $\{A \vee \neg B, C \vee \neg A, A\}$ a $\{C, B \to C, A \vee \neg C\}$ ekvivalentní? Jsou teorie $\{A \vee B, \neg A \vee C\}$ a $\{A \to C, B \vee C\}$ ekvivalentní?

1.4.4 Cvičení. Je-li $T \models A \vee B$, je nutně $T \models A$ nebo $T \models B$?

1.4.5 Cvičení. Výroková teorie T je ***nezávislá***, pokud žádná formule φ z T není důsledkem teorie $T \setminus \{\varphi\}$; jinými slovy, pokud pro každou φ z T je teorie $T, \neg\varphi$ splnitelná. Ukažte, že (a) každá konečná teorie má nezávislou ekvivalentní podmnožinu; (b) nekonečná teorie nemusí mít nezávislou ekvivalentní podmnožinu; (c) ke každé teorii existuje ekvivalentní nezávislá teorie.

Rezoluční metoda rozšiřuje danou konečnou teorii T do jisté ekvivalentní teorie $\mathcal{R}(T)$, o jejíž splnitelnosti lze rozhodnout triviálně. Z oddílu o normálních tvarech víme, že každou výrokovou formuli, a tedy i konečnou výrokovou teorii, lze ekvivalentně vyjádřit v konjunktivním normálním tvaru. Bez újmy na obecnosti hledíme tedy na předloženou výrokovou teorii jako na množinu klauzulí, a na tyto klauzule jako na množiny literálů.

Jsou-li $(A \vee B_1 \vee \cdots \vee B_n)$ a $(\neg A \vee C_1 \vee \cdots \vee C_m)$ dvě klauzule, pak formule $(B_1 \vee \cdots \vee B_n \vee C_1 \vee \cdots \vee C_m)$ je jejich ***rezolventa***. Rezolventa může být i prázdná, například klauzule A a $\neg A$ mají prázdnou rezolventu; tu budeme značit $\bot$ a nazývat ji ***kontradikce***, jak je zvykem. Snadno se ověří, že rezolventa je logickým důsledkem klauzulí, ze kterých vzešla.

1.4.6 Lemma. *Pokud nějaké pravdivostní ohodnocení splňuje výrokové formule* $(A \vee B_1 \vee \cdots \vee B_n)$ *a* $(\neg A \vee C_1 \vee \cdots \vee C_m)$*, pak splňuje i* $(B_1 \vee \cdots \vee B_n \vee C_1 \vee \cdots \vee C_m)$.

Je-li T množina klauzulí, označme jako $R(T)$ sjednocení T s množinou všech možných rezolvent dvojic klauzulí z T. Zřejmě $T \subseteq R(T)$, přitom $R(T)$ obsahuje výhradně důsledky teorie T, takže teorie T a $R(T)$ jsou ekvivalentní. Navíc pro T konečnou je $R(T)$ konečná. Položme $R^0(T) = T$ a $R^{n+1}(T) = R(R^n(T))$; pak

$$T = R^0(T) \subseteq R^1(T) \subseteq \ldots \subseteq R^n(T) \subseteq R^{n+1}(T) \subseteq \ldots$$

je rostoucí posloupnost konečných teorií. Přitom nad konečně mnoha literály z T existuje jen konečně mnoho navzájem různých klauzulí. Pro nějaké $n \in \mathbb{N}$ je tedy $R^n(T) = R^{n+1}(T)$. Tuto množinu klauzulí budeme nazývat *rezoluční obal* teorie T značit ji $\mathcal{R}(T)$. Teorie T a $\mathcal{R}(T)$ jsou opět ekvivalentní; specielně T je splnitelná právě když $\mathcal{R}(T)$ je splnitelná.

1.4.7 Příklad. Pro konečnou $T = \{A \vee B, B \to C, C \to D, D \to E\}$ je $R^0(T) = \{A \vee B, \neg B \vee C, \neg C \vee D, \neg D \vee E\}$, $R^1(T) = R^0(T) \cup \{A \vee C, \neg B \vee D, \neg C \vee E\}$ a $R^2(T) = R^1(T) \cup \{A \vee D, \neg B \vee E, A \vee E\}$. Systematickou probírkou všech dvojic klauzulí se snadno ověří, že $R^2(T) = R^3(T) = \mathcal{R}(T)$.

1.4.8 Věta (o rezoluci). *Konečná množina klauzulí T je splnitelná právě tehdy, když její rezoluční obal $\mathcal{R}(T)$ neobsahuje kontradikci.*

Důkaz. Jeden směr je triviální: pokud $\mathcal{R}(T)$ obsahuje kontradikci, pak je nesplnitelná, a ekvivalentní teorie T rovněž. V opačném směru ukážeme, že pokud $\mathcal{R}(T)$ neobsahuje kontradikci, pak je splnitelná, takže i $T \subseteq \mathcal{R}(T)$ je splnitelná.

Buď $A_1, \ldots, A_k$ jazyk teorie T, tedy seznam všech prvotních formulí, které se vyskytují v klauzulích z T. Indukcí zkonstruujeme ohodnocení v těchto výrokových proměnných, které splní teorii $\mathcal{R}(T)$. Je-li A_j první dosud neohodnocená výroková proměnná (ve zvoleném očíslování), definujeme $v(A_j)$ takto: pokud nějaká klauzule z $\mathcal{R}(T)$ sestává výhradně z $\neg A_j$ a z literálů ohodnocených opačně než při dosavadním ohodnocení, buď $v(A_j) = 0$; jinak buď $v(A_j) = 1$.

Pokud takové ohodnocení v nesplňuje nějakou klauzuli φ z $\mathcal{R}(T)$, znamená to, že φ obsahuje výhradně literály ohodnocené opačně než při ohodnocení v; v takovém případě buď $j \leq k$ nejmenší možné číslo, pro které jsou všechny výrokové proměnné z nějaké takové klauzule φ obsaženy mezi $A_1, \ldots, A_j$. To neznamená, že se ve φ vyskytují nutně všechny, ale proměnná A_j ve φ figurovat musí, jinak zvolené j nebylo nejmenší možné. Popíšeme případ, kdy φ obsahuje literál A_j — opačný případ, kdy φ obsahuje $\neg A_j$, je analogický.

Máme tedy $v(A_j) = 0$, jinak je klauzule φ splněna. Podle definice ohodnocení v to znamená, že nějaká klauzule ψ z $\mathcal{R}(T)$ sestává výhradně z $\neg A_j$ a literálů ohodnocených opačně než proměnné $A_1, \ldots, A_{j-1}$. Přitom proměnná A_j se v ψ musí vyskytovat, jinak zvolené $j \leq k$ nebylo nejmenší možné; tedy ψ obsahuje literál $\neg A_j$. Pak ale rezolventa klauzulí φ a ψ, přítomná v obalu $\mathcal{R}(T)$, obsahuje výhradně literály ohodnocené opačně než proměnné $A_1, \ldots, A_{j-1}$. To je spor s minimalitou zvoleného $j \leq k$. Zbývá jen možnost, kdy tato rezolventa je prázdná. Podle předpokladu ale $\mathcal{R}(T)$ kontradikci neobsahuje. □

1.4.9 Příklad. Je teorie $\{P \wedge Q \to R, \neg R \wedge P, \neg Q \vee \neg R\}$ splnitelná? Rezoluce se stabilizuje bez kontradikce, navíc mezi rezolventami je $\neg Q$, tedy mezi důsledky jsou již všechny literály: $P, \neg Q, \neg R$. To je jediné splňující ohodnocení.

1.4.10 Cvičení. (a) Je teorie $T = \{A \to (B \vee C), E \to (C \vee D), \neg C\}$ splnitelná? Je formule $\varphi \rightleftharpoons (\neg B \wedge \neg D) \to (\neg A \wedge \neg E)$ jejím důsledkem? Jako v každém konečném případě lze odpovědět s pomocí pravdivostních tabulek — to však znamená vyšetřit 2^5 různých ohodnocení pro čtyři různé formule. Ptejme se ekvivalentně, zda je teorie $T, \neg\varphi$ splnitelná.

1.4.11 Cvičení. Je $\{B \wedge D \to E, B \wedge C \to F, E \vee F \to A, \neg C \to D, B\} \models A$? Je $\{B \wedge D \to E, B \wedge C \to F, E \vee F \to A, C \to D, B\} \models A$?

1.4.12 Cvičení. Politická strana *Práce a mír* potřebuje vysekat svého ministra z žaloby pro korupci. To vyžaduje buďto zastrašit svědka A nebo podplatit soudce B. Na zastrašení svědka A je potřeba uvěznit osobu C. Pro uplacení soudce B je potřeba obsadit firmu F a získat pro ni zakázku E. Uvěznění osoby C i převzetí firmy F vyžaduje zabít osobu D. Potřebuje *Práce a mír* zabít D?

1.4.13 Cvičení (Davis–Putnam)**.** Rozhodněte o splnitelnosti resp. důsledku v předchozích cvičeních pomocí následujícího algoritmu. Vstupem je konečná výroková teorie T v konjunktivním normálním tvaru, tedy množina klauzulí. Výstupem je informace o tom, zda T je či není splnitelná.

```
/* 1. propagace literálů */
dokud T obsahuje literály {
    pro každý literál L v T {
        odstraň z T všechny klauzule obsahující L
        odstraň opačný literál ze všech klauzulí, které jej obsahují
    }
    pokud je nyní T prázdná, vrať true
    pokud T obsahuje prázdnou klauzuli, vrať false
}
/* 2. splitting (rekurzí) */
zvol nějaký literál L z nějaké zbylé klauzule
je-li T,L splnitelná, vrať true
je-li T,not(L) splnitelná, vrať true
jinak vrať false
```

1.4.14 Cvičení. Pro formuli φ a její prvotní podformuli A buď $\varphi_A^\bot$ formule, která vznikne z formule φ nahrazením každého výskytu A symbolem $\bot$, a podobně buď $\varphi_A^\top$ formule, kde je A nahrazena symbolem $\top$. Přitom symboly $\bot, \top$ považujeme za nulární spojky s hodnotami 0 resp 1. Označme jako φ_A^* formuli $\varphi_A^\bot \vee \varphi_A^\top$. Potom (i) $\varphi \models \varphi_A^*$; (ii) je-li ψ formule, která neobsahuje A, a je-li $\varphi \models \psi$, je také $\varphi_A^* \models \psi$; (iii) φ je splnitelná právě když φ_A^* je splnitelná. Formule φ_A^* je nejsilnější možný důsledek formule φ, který nepoužívá A.

1.4.15 Cvičení. Buď $\varphi \models \psi$. Potom existuje formule ϑ, pro kterou $\varphi \models \vartheta \models \psi$, a která používá jen ty proměnné, jež se vyskytují ve formuli φ i ve formuli ψ. Říkáme pak, že formule ϑ *interpoluje* vztah $\varphi \models \psi$.

Kompaktnost Pro konečnou výrokovu teorii $T = \{\varphi_1, \ldots, \varphi_n\}$ se otázka po splnitelnosti redukuje na splnitelnost formule $\varphi_1 \wedge \cdots \wedge \varphi_n$, kterou dokážeme zodpovědět dokonce algoritmicky. Pro nekonečnou teorii T jsou však takové metody nepoužitelné. V tomto paragrafu dokážeme větu o kompaktnosti výrokové logiky, která popisuje splnitelnost nekonečných teorií.

1.4.16 Cvičení. (a) V jazyce prvotních formulí $\{A_n; n \in \mathbb{N}\}$ uvažte nekonečné teorie $S = \{\neg A_n \leftrightarrow A_{n+2}; n \in \mathbb{N}\}$ a $T = \{\neg A_n \leftrightarrow (A_{n+1} \vee A_{n+2}); n \in \mathbb{N}\}$. Pro každou z nich rozhodněte, zda je splnitelná, a pokud ano, popište všechna splňující ohodnocení. (b) Ukažte, že ani jedna z teorií S a T nevyplývá z druhé. (c) Pro nekonečnou teorii T je přirozené se ptát, zda existuje nějaká konečná část $T_0 \subset T$ tak, že $T \models T_0$. Otázku splnitelnosti teorie T lze potom redukovat na otázku splnitelnosti konečné teorie T_0. Ukažte, že teorie S a T výše nemají žádnou konečnou ekvivalentní část.

1.4.17 Cvičení. Popište nějakou výrokovou teorii T, která není splnitelná, ale (i) každá formule z T je splnitelná; (ii) každá dvouprvková $S \subseteq T$ je splnitelná; (ii) každá tříprvková $S \subseteq T$ je splnitelná.

1.4.18 Věta (o kompaktnosti výrokové logiky)**.** *Výroková teorie je splnitelná právě tehdy, když je splnitelná každá její konečná část.*

Věta je zajímavá jen pro nekonečné teorie. Zároveň jeden směr tvrzení je triviální: ohodnocení splňující danou teorii splňuje i každou její část.

Uvedeme dva důkazy věty o kompaktnosti: nejprve pro speciální případ spočetného jazyka, ve kterém vystačíme s matematickou indukcí. V rámci důkazu budeme používat pojem *konečně splnitelná teorie*, to je taková, jejíž každá konečná část je splnitelná. Máme ukázat, že taková teorie je ve skutečnosti splnitelná. K důkazu budeme potřebovat následující pozorování.

1.4.19 Lemma. *Buď T konečně splnitelná výroková teorie, buď φ výroková formule. Pak alespoň jedna z teorií $T \cup \{\varphi\}$ nebo $T \cup \{\neg\varphi\}$ je konečně splnitelná.*

Důkaz. Pokud ne, pak existují nějaké konečné části $T_0 \subseteq T$ a $T_1 \subseteq T$ tak, že ani teorie $T_0 \cup \{\varphi\} \subseteq T \cup \{\varphi\}$, ani teorie $T_1 \cup \{\neg\varphi\} \subseteq T \cup \{\neg\varphi\}$ není splnitelná. Pak ale $T_0 \cup T_1 \subseteq T$ je nesplnitelná konečná část teorie T. □

Důkaz věty o kompaktnosti. Buď T nějaká konečně splnitelná výroková teorie. Hledáme ohodnocení, které splní celou teorii T. Předpokládáme, že jazyk je spočetný, takže všechny výrokové formule lze očíslovat přirozenými čísly a v nějakém pořadí $\{\varphi_n; n \in \mathbb{N}\}$ je projít indukcí.[14]

Budujeme postupně výrokovou teorii U, která rozšiřuje teorii T. Položme $U_0 = T$. Známe-li konečně splnitelnou teorii U_n, buď $U_{n+1} = U_n \cup \{\varphi_n\}$, pokud je konečně splnitelná; jinak buď $U_{n+1} = U_n \cup \{\neg\varphi_n\}$. Každopádně U_{n+1} je opět konečně splnitelná, podle předchozího lemmatu. Nakonec položme $U = \bigcup U_n$.

Teorie U je konečně splnitelná: konečná část U je již částí některé U_n, a ta je konečně splnitelná. Zároveň pro každé dvě výrokové formule φ, ψ platí:

(i) $\neg\varphi \in U$ právě když $\varphi \notin U$. Oba případy nastat nemohou, jelikož U je konečně splnitelná. Naopak každá formule φ má svůj index v očíslování φ_n, takže nejpozději v U_{n+1} se ocitne buďto φ nebo $\neg\varphi$.

(ii) $\varphi \wedge \psi \in U$ právě když $\varphi, \psi \in U$. Kdyby totiž $\varphi \wedge \psi \in U$, ale $\varphi \notin U$ nebo $\psi \notin U$, pak by podle (i) bylo $\neg\varphi \in U$ nebo $\neg\psi \in U$, a tedy $\{\neg\varphi, \varphi \wedge \psi\}$ nebo $\{\neg\psi, \varphi \wedge \psi\}$ by byl nesplnitelný konečný fragment teorie U. Je-li naopak $\varphi, \psi \in U$, ale $\varphi \wedge \psi \notin U$, pak podle (i) je $\neg(\varphi \wedge \psi) \in U$, načež $\{\varphi, \psi, \neg(\varphi \wedge \psi)\}$ je nesplnitlený konečný fragment.

(iii) $\varphi \vee \psi \in U$ právě když $\varphi \in U$ nebo $\psi \in U$. Je-li totiž $\varphi \vee \psi \in U$, ale $\varphi, \psi \notin U$, pak je $\neg\varphi, \neg\psi \in U$ podle (i), a tedy $\{\varphi \vee \psi, \neg\varphi, \neg\psi\}$ je konečný nesplnitelný fragment. Podobně naopak.

(iv) $\varphi \to \psi \in U$ právě když $\neg\varphi \in U$ nebo $\psi \in U$. Je-li totiž $(\varphi \to \psi) \in U$, ale $\neg\varphi, \psi \notin U$, pak je $\varphi, \neg\psi \in U$ podle (i), načež $\{\varphi, \varphi \to \psi, \neg\psi\}$ je konečný nesplnitelný fragment. Podobně naopak.

(v) $\varphi \leftrightarrow \psi \in U$ právě když $\varphi, \psi \in U$ nebo $\neg\varphi, \neg\psi \in U$. Je-li totiž $\varphi \leftrightarrow \psi \in U$, ale například $\varphi \in U$ a $\psi \notin U$, pak $\neg\psi \in U$ podle (i), načež $\{\varphi \leftrightarrow \psi, \varphi, \neg\psi\}$ je konečný nesplnitelný fragment. Podobně naopak.

[14] Zdůrazňujeme, že jsme očíslovali vůbec *všechny* formule daného jazyka, ne jen teorii T.

Položme konečně $v(\varphi) = 1$ právě když $\varphi \in U$. Podle výše uvedených vlastností se jedná o pravdivostní ohodnocení podle 1.2.1: respektuje negace, konjunkce, atd. Při ohodnocení v jsou splněny právě všechny formule z U, tedy specielně všechny formule z $T \subseteq U$. Tedy T je splnitelná. □

Zbývá větu o kompaktnosti dokázat pro jazyk $\mathcal{A}$ libovolné mohutnosti. Předvedeme obecný důkaz, který nepředpokládá spočetnost jazyka — neobejdeme se ale bez jistých topologických pojmů.

Důkaz věty o kompaktnosti. Buď T nějaká konečně splnitelná výroková teorie. Pro každou konečnou část $S \subseteq T$ označme jako $sat(S)$ množinu těch ohodnocení $v : \mathcal{A} \to 2$, která splňují konečnou teorii S. Podle předpokladu je množina $sat(S)$ pro každou konečnou $S \subseteq T$ neprázdná. Snadno se nahlédne, že je uzavřená v topologickém součinu $2^{\mathcal{A}}$. Zároveň systém $\mathcal{S} = \{sat(S); S \subseteq T \text{ konečná}\}$ je centrovaný, neboť průnik $sat(S_1) \cap \cdots \cap sat(S_n)$ obsahuje neprázdnou množinu $sat(S_1 \cup \cdots \cup S_n)$. Máme tedy centrovaný systém $\mathcal{S}$ neprázdných uzavřených množin v topologickém prostoru $2^{\mathcal{A}}$, který je podle Tichonovovy věty kompaktní. Tedy průnik $\bigcap \mathcal{S}$ je neprázdný. Každé ohodnocení $v \in \bigcap \mathcal{S} \neq \emptyset$ splňuje všechny konečné fragmenty $S \subseteq T$ zároveň, specielně splňuje každou formuli z T. □

Poznamenejme, že předvedený důkaz je čistě *existenční*: ukázali jsme, že nějaké splňující ohodnocení existuje, ale žádné konkrétní takové jsme nepředvedli.

1.4.20 Důsledek. *Pro výrokovu teorii T a výrokovou formuli φ je $T \models \varphi$ právě tehdy, když $T_0 \models \varphi$ pro nějaký konečný fragment $T_0 \subseteq T$.*

Důkaz. $T \models \varphi$ právě když $T, \neg\varphi$ je nesplnitelná. Podle věty o kompaktnosti tedy již pro nějakou konečnou $T_0 \subseteq T$ je $T_0, \neg\varphi$ nesplnitelná; tedy $T_0 \models \varphi$. □

1.5 Dokazatelnost

Dosud jsme se v různých podobách zabývali *sémantikou* výrokové logiky, tj. zkoumali jsme otázky pravdivosti, splnitelnosti a důsledku. Nyní popíšeme druhou tvář výrokové logiky, její *formální odvozovací systém*. Zavedeme pojem *formálního důkazu* a budeme se ptát, které formule jsou dokazatelné, ať už v samotné logice nebo z jiných formulí. Dokážeme *větu o dedukci*, která podstatně zkracuje a zjednodušuje argumenty o dokazatelnosti, a předvedeme *úplnost* výrokové logiky, podle které jsou pojmy *pravdivosti* a *dokazatelnosti* výrokových formulí v nejlepším možném souladu.

Formální odvozovací systém Při výstavbě formálního odvozovacího systému výrokové logiky potřebujeme nejprve popsat *jazyk*, který budeme používat. Některé formule tohoto jazyka pak zvolíme za *axiomy*, ze kterých budeme odvozovat vše ostatní, a stanovíme *odvozovací pravidla*, která popisují jediné povolené způsoby takového odvozování. Je téměř filozofickou otázkou, které axiomy a jaká pravidla zvolit, a různé formální systémy nabízejí různé odpovědi. Systém navržený D. Hilbertem je všeobecně považován za standardní.

Hilbertův systém Jazykem Hilbertova systému je jazyk výrokové logiky redukovaný na spojky $\neg$ a $\to$. Smyslem této redukce je úspornost vyjádření: víme z 1.3.1, že $\{\neg, \to\}$ tvoří minimální univerzální množinu spojek. Axiomem je každá instance každé z následujících formulí.

H1: $A \to (B \to A)$

H2: $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

H3: $(\neg B \to \neg A) \to (A \to B)$

Jediným odvozovacím pravidlem je *modus ponens* neboli *pravidlo odloučení*:

MP: Z formulí φ a $\varphi \to \psi$ odvoď formuli ψ.

Je H1–H3 vhodným základem pro stavbu výrokové dokazatelnosti? Zachycuje právě modus ponens způsob, kterým rozum spěje od známého k novému? Těmito otázkami se zabývat nebudeme, ponecháme je filozofii matematiky.

1.5.1 Cvičení. Hilbertův systém nemá tři axiomy, nýbrž nekonečně mnoho axiomů tří typů. (a) Které z následujících formulí jsou axiomy, a kterého typu? (b) S pomocí 1.1.7 implementujte proceduru, která rozpoznává, zda vstupní formule je hilbertovským axiomem, a kterého typu.

$(A \to B) \to ((\neg C \leftrightarrow (D \wedge E)) \to (A \to B))$
$(A \to B) \to ((\neg C \leftrightarrow (D \wedge E)) \to (A \to (A \vee B)))$
$(A \to ((B \wedge \neg C) \to D)) \to ((A \to (B \wedge \neg C)) \to (A \to D))$
$(A \to ((B \wedge \neg C) \to D)) \to ((A \to (B \wedge \neg C)) \to D)$
$(\neg(A \wedge B) \to (C \vee D)) \to (\neg(C \vee D) \to (A \wedge B))$
$(\neg(A \wedge B) \to \neg\neg(C \vee D)) \to (\neg(C \vee D) \to (A \wedge B))$

1.5.2 Definice. Konečná posloupnost výrokových formulí $\varphi_1, \ldots, \varphi_n$ je *důkaz* formule φ ve výrokové logice, pokud každá formule φ_i je buďto axiom, nebo je z některých předcházejících formulí odvozena pravidlem modus ponens, a φ_n je formule φ. Pokud existuje nějaký důkaz formule φ, řekneme, že φ je *dokazatelná* ve výrokové logice, a píšeme $\vdash \varphi$.

Pojem formálního důkazu zachycuje to, co od něj v matematice očekáváme: vychází z předem výslovně daných předpokladů, používá konečně mnoho předem daných pravidel, a je kdykoli znovu ověřitelný v každém ze svých konečně mnoha kroků. Toto ověření může být dokonce mechanické, viz 1.5.6.

1.5.3 Příklad. Následující posloupnost formulí je důkazem formule $A \to A$ ve výrokové logice. U každého kroku uvádíme výslovně, kterého axiomu či odvozovacího pravidla je právě použito.

H1: $(A \to ((A \to A) \to A))$

H2: $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$

MP: $(A \to (A \to A)) \to (A \to A)$

H1: $(A \to (A \to A))$

MP: $(A \to A)$

Všimněme si, že pojem důkazu je čistě syntaktický: formální důkaz je jistá posloupnost formulí, tedy výrazů jistého tvaru, která sama je jistého tvaru.

Jedná se výhradně o manipulaci se symboly — otázka pravdivosti nehraje ve formálním důkaze žádnou roli.[15]

Je snadné ověřit, že uvedená posloupnost je skutečně důkazem. To ale nedává žádný návod, jak jej nalézt. Později uvidíme, že pro dokazatelnou formuli je i nalezení důkazu algoritmicky řešitelný problém, i když výpočetně dosti náročný.

1.5.4 Cvičení. Ověřte podrobně, že následující posloupnost formulí je formálním důkazem.[16] U každého kroku uveďte, kterého axiomu či odvozujícího pravidla bylo právě použito. Existuje nějaký kratší důkaz?

$\neg A \to (\neg B \to \neg A)$
$(\neg B \to \neg A) \to (A \to B)$
$((\neg B \to \neg A) \to (A \to B)) \to (\neg A \to ((\neg B \to \neg A) \to (A \to B)))$
$\neg A \to ((\neg B \to \neg A) \to (A \to B))$
$(\neg A \to ((\neg B \to \neg A) \to (A \to B))) \to ((\neg A \to (\neg B \to \neg A)) \to (\neg A \to (A \to B)))$
$(\neg A \to (\neg B \to \neg A)) \to (\neg A \to (A \to B))$
$\neg A \to (A \to B)$

1.5.5 Cvičení. Je-li $\varphi_1, \ldots, \varphi_n$ formální důkaz, buďte $A_1, \ldots, A_k$ všechny prvotní formule, které se v něm vyskytují. Buďte dále $\psi_1, \ldots, \psi_k$ libovolné výrokové formule. Potom posloupnost $\varphi_1^*, \ldots, \varphi_n^*$, kde formule φ_i^* je instance formule φ_i vzniklá substitucí formulí ψ_j za proměnné A_j, je opět výrokovým důkazem. Stručně řečeno, každá „instance důkazu“ je opět důkazem, takže důkaz libovolné formule lze snadno přepsat na důkaz libovolné její instance.

1.5.6 Cvičení. Implementujte *proof checker*, tj. program, který na vstupu čte konečnou posloupnost výrokových formulí (jedna formule na jednom řádku), a přitom ověřuje, zda je tato posloupnost důkazem v Hilbertově systému.

1.5.7 Definice. Buď T výroková teorie, buď φ výroková formule. O posloupnosti výrokových formulí $\varphi_1, \ldots, \varphi_n$ řekneme, že je ***důkazem formule φ z předpokladů T*** (nebo že je ***důkazem φ v teorii T***), pokud φ_n je formule φ, a každá formule φ_i je buďto axiom výrokové logiky, nebo formule z T, nebo je z nějakých předchozích φ_j, φ_k odvozena pravidlem modus ponens. Pokud nějaký takový důkaz φ z T existuje, řekneme, že φ je ***dokazatelná v T***, a píšeme $T \vdash \varphi$.

Zobecnění je tedy v tom, že jako kroky důkazu připouštíme i formule z množiny T. Výše zavedená notace $\vdash \varphi$ pak odpovídá tomu, že φ je dokazatelná z prázdné množiny předpokladů, tedy v samotné výrokové logice.

Je zvykem psát stručněji například $B, \neg A \to \neg B \vdash A$ místo formálnějšího $\{B, \neg A \to \neg B\} \vdash A$. Podobně rozšiřujeme-li teorii T o nějaké další předpoklady φ a ψ, je zvykem psát stručně T, φ, ψ místo $T \cup \{\varphi, \psi\}$ apod.

1.5.8 Cvičení. (a) Ověřte podrobně kroky formálního důkazu formule $A \to B$ z $\neg A$: $\neg A$, $\neg A \to (\neg B \to \neg A)$, $\neg B \to \neg A$, $(\neg B \to \neg A) \to (A \to B)$, $A \to B$.
(b) Podejte důkaz formule $A \to B$ z B a důkaz formule A z $B, \neg A \to \neg B$.

[15] Pozorný čtenář se zřejmě pozastaví nad tím, že i v tomto textu předkládáme „důkazy,“ a nejsou to posloupnosti formulí (zatím kromě 1.5.3). Abychom tyto dvě úrovně jazyka důsledně oddělili, mohli bychom důkazy v tomto textu nazývat *metadůkazy* nebo *demonstrace*. Zůstaneme však u dosavadní terminologie, a spoléháme na čtenáře, který je schopen rozlišit formální důkaz formule *ve výrokové logice* od demonstrací vět *o výrokové logice*, vedených v češtině; ta hraje roli *metajazyka*, kterým zde mluvíme o formulích, teoriích — a důkazech.

[16] Dokazovaná formule je jednou z pouček starověké logiky, a jako každá taková má své latinské znění: *ex impossibili sequitur quodlibet*, neboli *z nemožného plyne cokoli*.

1.5.9 Cvičení ([T]). Pro výrokovou teorii T označme jako $Thm(T)$ množinu těch formulí, které mají důkaz v T. Rozhodněte, zda platí následující tvrzení:

(a) $T \subseteq Thm(T)$

(b) $Thm(Thm(T)) = Thm(T)$

(c) $S \subseteq T$ právě tehdy, když $Thm(S) \subseteq Thm(T)$

(d) $S \subseteq Thm(T)$ právě tehdy, když $Thm(S) \subseteq Thm(T)$

(e) $Thm(S \cup T) = Thm(S) \cup Thm(T)$

(f) $Thm(S \cup T) = Thm(S \cup Thm(T)) = Thm(Thm(S) \cup Thm(T))$

(g) Je-li $T_n \subseteq T_{n+1}$ pro každé $n \in \mathbb{N}$, pak $Thm(\bigcup T_n) = \bigcup Thm(T_n)$

Korektnost a bezespornost Po zavedení pojmu důkazu a dokazatelnosti je přirozené se ptát, které formule jsou dokazatelné, resp. dokazatelné z daných předpokladů. Ukážeme nejprve, že Hilbertův systém je *korektní* a výroková logika je tudíž *bezesporná*.

1.5.10 Věta (o korektnosti). *Buď T výroková teorie a φ výroková formule. Potom pokud je φ dokazatelná v T, je logickým důsledkem T. Speciálně výroková logika dokazuje výhradně tautologie.*

Důkaz. Tvrzení dokážeme indukcí podle délky důkazu. Buď totiž $\varphi_1, \varphi_2, \ldots, \varphi_n$ důkaz formule φ v teorii T. Je-li φ_i axiom výrokové logiky, je tautologií, jak jsme ověřili v 1.2.11 a 1.2.21, a tedy $T \models \varphi_i$. Je-li φ_i prvkem T, je $T \models \varphi_i$ z definice. Konečně je-li φ_i odvozena z nějakých předchozích formulí pomocí modus ponens, stačí si uvědomit, že pravdivostní ohodnocení, které splňuje formule ϱ a $\varrho \to \vartheta$, splňuje i formuli ϑ; tedy modus ponens odvozuje z tautologií jen tautologie, a z důsledků teorie T opět jen důsledky teorie T. □

1.5.11 Definice. Výroková teorie je *sporná*, pokud z ní lze dokázat libovolnou formuli; v opačném případě je *bezesporná*. Řekneme, že sám formální systém je *sporný*, pokud prázdná teorie je sporná.

Podle věty o korektnosti tedy *výroková logika je bezesporná*: formule, které nejsou tautologické, nelze v logice dokázat. Z věty o korektnosti také získáváme:

1.5.12 Věta. *Každá splnitelná teorie je bezesporná.*

Důkaz. Buď T daná teorie, buď v nějaké splňující ohodnocení. Je-li T sporná, dokazuje z definice každou formuli, tedy specielně nějakou φ a zároveň $\neg\varphi$. Podle věty o korektnosti je potom zároveň $T \models \varphi$ a $T \models \neg\varphi$. Ohodnocení v tedy splňuje jak φ tak $\neg\varphi$, což není možné. □

Právě použitá vlastnost sporné teorie, totiž dokazatelnost nějaké formule φ a $\neg\varphi$ současně, ve skutečnosti sporné teorie charakterisuje, a někdy je přijímána jako definice. Podle 1.5.4 je totiž $\vdash \neg\varphi \to (\varphi \to \psi)$ pro každou formuli ψ, a pokud je $T \vdash \varphi$ a $T \vdash \neg\varphi$, máme dvojím použitím modus ponens také $T \vdash \psi$.

Věta o dedukci Předvedeme důležitý technický obrat, který mnohokrát využijeme při prokazování dokazatelnosti: větu o dedukci, která důkazy zpřehledňuje a zkracuje. S její pomocí pak dokážeme několik jednoduchých formulí, které budeme v dalším běžně používat.

1.5.13 Věta (o dedukci). *Buď T výroková teorie, buďte φ, ψ výrokové formule. Potom je $T \vdash \varphi \to \psi$ právě když $T, \varphi \vdash \psi$.*

Věta o dedukci je ospravedlněním běžného obratu, který používáme při důkazu nějaké implikace $\varphi \to \psi$: předpoklad φ přidáme k ostatním předpokladům, a v takto vzniklé teorii dokazujeme závěr ψ. Takový důkaz je typicky kratší a průhlednější, neboť dokazujeme jednodušší formuli ze silnějších předpokladů.

Demonstrace věty je konstruktivní: popisuje algoritmus, který přepisuje důkaz formule ψ v teorii T, φ na důkaz formule $\varphi \to \psi$ v teorii T a naopak. Je vedena *indukcí podle délky důkazu*.

Důkaz. (i) Je-li $T \vdash \varphi \to \psi$, buď $\vartheta_1, \vartheta_2, \ldots, \vartheta_n \rightleftharpoons (\varphi \to \psi)$ důkaz formule $\varphi \to \psi$ v teorii T. Přidáme-li k tomuto důkazu ještě formule φ, ψ, získáme posloupnost $\vartheta_1, \vartheta_2, \ldots, (\varphi \to \psi), \varphi, \psi$, která je důkazem formule ψ v teorii T, φ.

(ii) Je-li $T, \varphi \vdash \psi$, buď $\vartheta_1, \vartheta_2, \ldots, \vartheta_n \rightleftharpoons \psi$ důkaz formule ψ v teorii T, φ. Ukážeme postupně indukcí pro $i \leq n$, že pro každou formuli ϑ_i z důkazu je $T \vdash \varphi \to \vartheta_i$. Tím bude specielně pro $i = n$ ukázáno, že $T \vdash \varphi \to \psi$.

(a) Je-li ϑ_i axiom logiky, pak $\vartheta_i, \vartheta_i \to (\varphi \to \vartheta_i), \varphi \to \vartheta_i$ je důkazem $\varphi \to \vartheta_i$.

(b) Je-li ϑ_i předpoklad z T, pak posloupnost z (a) je důkazem $\varphi \to \vartheta_i$ z T.

(c) Je-li $\vartheta_i \rightleftharpoons \varphi$, máme ukázat $T \vdash \varphi \to \varphi$. Podle 1.5.3 je dokonce $\vdash \varphi \to \varphi$.

(d) Je-li ϑ_i odvozena z nějakých předchozích $\vartheta_j, \vartheta_j \to \vartheta_i$, pak podle indukčního předpokladu je již $T \vdash \varphi \to \vartheta_j$ a $T \vdash \varphi \to (\vartheta_j \to \vartheta_i)$. Důkaz $\varphi \to \vartheta_i$ z T potom vznikne tak, že za důkazy formulí $\varphi \to \vartheta_j$ a $\varphi \to (\vartheta_j \to \vartheta_i)$ zařadíme

H2: $(\varphi \to (\vartheta_j \to \vartheta_i)) \to ((\varphi \to \vartheta_j) \to (\varphi \to \vartheta_i))$

MP: $(\varphi \to \vartheta_j) \to (\varphi \to \vartheta_i)$

MP: $\varphi \to \vartheta_i$

Jiné případy než (a)–(d) nastat nemohou. Tím je věta dokázána. □

1.5.14 Příklad. Přepíšeme důkaz 1.5.8 formule $A \to B$ z formule $\neg A$ na důkaz implikace $\neg A \to (A \to B)$ ve výrokové logice. Následujeme konstruktivní demonstraci věty o dedukci: pro každou formuli $\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5$ původního důkazu, tj. $\neg A$, $\neg A \to (\neg B \to \neg A)$, $\neg B \to \neg A$, $(\neg B \to \neg A) \to (A \to B)$, $A \to B$, sestrojíme důkaz formule $\neg A \to \vartheta_i$.

(1) Formule ϑ_1 je samotná $\neg A$, případ 1.5.13 (c), pomocí 1.5.3.

$(\neg A \to ((\neg A \to \neg A) \to \neg A))$
$(\neg A \to ((\neg A \to \neg A) \to \neg A)) \to ((\neg A \to (\neg A \to \neg A)) \to (\neg A \to \neg A))$
$(\neg A \to (\neg A \to \neg A)) \to (\neg A \to \neg A)$
$(\neg A \to (\neg A \to \neg A))$
$(\neg A \to \neg A)$

(2) Formule ϑ_2 je axiomem výrokové logiky, případ 1.5.13 (a):

$\neg A \to (\neg B \to \neg A)$
$(\neg A \to (\neg B \to \neg A)) \to (\neg A \to (\neg A \to (\neg B \to \neg A)))$
$\neg A \to (\neg A \to (\neg B \to \neg A))$

(3) Formule ϑ_3 vznikla užitím modus ponens na ϑ_1 a ϑ_2, případ 1.5.13 (d):

$(\neg A \to (\neg A \to (\neg B \to \neg A))) \to ((\neg A \to \neg A) \to (\neg A \to (\neg B \to \neg A)))$
$(\neg A \to \neg A) \to (\neg A \to (\neg B \to \neg A))$
$\neg A \to (\neg B \to \neg A)$

(4) Formule ϑ_4 je opět axiomem logiky, případ 1.5.13 (a):

$(\neg B \to \neg A) \to (A \to B)$
$((\neg B \to \neg A) \to (A \to B)) \to (\neg A \to ((\neg B \to \neg A) \to (A \to B)))$
$\neg A \to ((\neg B \to \neg A) \to (A \to B))$

(5) Formule ϑ_5 vznikla užitím modus ponens na ϑ_3 a ϑ_4, případ 1.5.13 (d):

$(\neg A \to ((\neg B \to \neg A) \to (A \to B))) \to ((\neg A \to (\neg B \to \neg A)) \to (\neg A \to (A \to B)))$
$(\neg A \to (\neg B \to \neg A)) \to (\neg A \to (A \to B))$
$\neg A \to (A \to B)$

Formální důkazy zřejmě rychle rostou do délky i pro jednoduché formule.[17] Věta o dedukci umožňuje *předvést dokazatelnost* bez nutnosti *podat konkrétní důkaz*, a udržet argumenty o dokazatelnosti na únosné délce. Přitom každé použití věty o dedukci lze eliminovat jako výše, zcela mechanicky.

1.5.15 Příklad. Formule $(A \to (B \to C)) \to (B \to (A \to C))$ je dokazatelná. Podle věty o dedukci jsou totiž následující tvrzení ekvivalentní

$$\begin{aligned} &\vdash (A \to (B \to C)) \to (B \to (A \to C)) \\ A \to (B \to C) &\vdash B \to (A \to C) \\ A \to (B \to C), B &\vdash A \to C \\ A \to (B \to C), B, A &\vdash C \end{aligned}$$

a poslední důkaz se snadno sestaví. Trojím použitím věty o dedukci jej potom lze přepsat na důkaz původní formule. Podobně se předvede dokazatelnost formule $(A \to B) \to ((B \to C) \to (A \to C))$.

1.5.16 Cvičení. Rozšiřte svůj proof checker o *důkazový preprocessor*, který bude přijímat i argumenty používající větu o dedukci, a všechna její použití rozvine do skutečného formálního důkazu, tak jako v 1.5.14.

Budeme větu o dedukci v dalším volně používat při demonstracích dokazatelnosti některých jednoduchých vět výrokové logiky, které budeme později potřebovat. Laskavý čtenář může ve vybraných případech vyzkoušet, oč pracnější by bylo předložit konkrétní důkaz ve výrokové logice, případně může takové důkazy pomocí 1.5.16 konstruovat.

1.5.17 Lemma. (i) $\vdash \neg\neg A \to A$, (ii) $\vdash A \to \neg\neg A$.

Důkaz. S využitím 1.5.4 a věty o dedukci máme

1.5.4: $\vdash \neg\neg A \to (\neg A \to \neg\neg\neg A)$
VD: $\neg\neg A \vdash (\neg A \to \neg\neg\neg A)$
H3: $\vdash (\neg A \to \neg\neg\neg A) \to (\neg\neg A \to A)$

[17] Právě zkonstruovaný důkaz je dosti neefektivní a lze jej podstatně zjednodušit: prvních jedenáct kroků dokazuje instanci axiomu, který je navíc přítomen už na šestém kroku. Odstraněním takových redundancí vznikne právě důkaz 1.5.4.

MP: $\neg\neg A \vdash \neg\neg A \to A$

VD: $\neg\neg A \vdash A$

VD: $\vdash \neg\neg A \to A$

a

(i) $\vdash \neg\neg\neg A \to \neg A$

H3: $\vdash (\neg\neg\neg A \to \neg A) \to (A \to \neg\neg A)$

MP: $\vdash A \to \neg\neg A$

□

1.5.18 Lemma. (i) $\vdash (A \to B) \to (\neg B \to \neg A)$, (ii) $\vdash A \to (\neg B \to \neg(A \to B))$

Důkaz. S využitím 1.5.17 a věty o dedukci máme (zapisujeme již stručněji)

1.5.17, VD: $\neg\neg A \vdash A$

MP: $\neg\neg A, A \to B \vdash B$

1.5.17, MP: $\neg\neg A, A \to B \vdash \neg\neg B$

VD: $A \to B \vdash \neg\neg A \to \neg\neg B$

H3, MP: $A \to B \vdash \neg B \to \neg A$

VD: $\vdash (A \to B) \to (\neg B \to \neg A)$

a

MP: $A, A \to B \vdash B$

VD: $A \vdash (A \to B) \to B$

(i), MP: $A \vdash \neg B \to \neg(A \to B)$

VD: $\vdash A \to (\neg B \to \neg(A \to B))$

□

1.5.19 Lemma. $\vdash (\neg A \to A) \to A$.

Důkaz. S využitím 1.5.18 a věty o dedukci máme

MP: $\neg A, \neg A \to A \vdash A$

VD: $\neg A \vdash (\neg A \to A) \to A$

1.5.18, MP: $\neg A \vdash \neg A \to \neg(\neg A \to A)$

VD, VD: $\vdash \neg A \to \neg(\neg A \to A)$

H3, MP: $\vdash (\neg A \to A) \to A$

□

1.5.20 Cvičení. (i) $\vdash (A \to \neg B) \to (B \to \neg A)$, (ii) $\vdash (\neg A \to B) \to (\neg B \to A)$.

S pomocí věty o dedukci a předchozích lemmat získáváme následující charakterizaci dokazatelnosti. Můžeme ji považovat za formalizaci důkazu sporem.

1.5.21 Lemma. *$T \vdash \varphi$ právě když $T, \neg\varphi$ je sporná teorie.*

Důkaz. (i) Podle 1.5.4 je $\vdash \neg\varphi \to (\varphi \to \psi)$ a podle 1.5.15 pak $\vdash \varphi \to (\neg\varphi \to \psi)$. Je-li tedy $T \vdash \varphi$, pak je $T \vdash \neg\varphi \to \psi$, a podle věty o dedukci je $T, \neg\varphi \vdash \psi$.

(ii) Je-li $T, \neg\varphi$ sporná, pak dokazuje všechny formule, specielně $T, \neg\varphi \vdash \varphi$, a podle věty o dedukci je $T \vdash \neg\varphi \to \varphi$. Přitom $\vdash (\neg\varphi \to \varphi) \to \varphi$ podle 1.5.19, takže pomocí modus ponens dostáváme $T \vdash \varphi$. □

Uvedené důkazy se týkají jen spojek $\neg$ a $\to$, ve kterých je formulována Hilbertova axiomatika. Ukážeme ještě některé jednoduché důkazy týkající se ostatních spojek, které chápeme jako zkratky za příslušné ekvivalenty v základním jazyce.

1.5.22 Lemma. (i) $A \wedge B \vdash A, B$, (ii) $A, B \vdash A \wedge B$.

Důkaz. (i) $A \wedge B$ je zkratka za $\neg(A \to \neg B)$. Podle 1.5.4 je $\neg A \to (A \to B)$, takže z 1.5.17 a 1.5.18 pomocí modus ponens máme $\vdash \neg(A \to \neg B) \to A$. Podle věty o dedukci tedy $\neg(A \to \neg B) \vdash A$. Podobně máme $\neg B \to (A \to \neg B)$ podle H1, takže z 1.5.17 a 1.5.18 pomocí modus ponens opět $\vdash \neg(A \to \neg B) \to B$. (ii) Podle 1.5.17 je $A, B \vdash \neg\neg B$, a pomocí 1.5.18 máme $A, \neg\neg B \vdash \neg(A \to \neg B)$. Pomocí modus ponens tedy $A, B \vdash A \wedge B$. □

1.5.23 Důsledek. *Formule $A \leftrightarrow B$ je zkratka za $(A \to B) \wedge (B \to A)$, takže*

(i) $A \leftrightarrow B \vdash A \to B$; $A \leftrightarrow B \vdash B \to A$

(ii) $A \to B, B \to A \vdash A \leftrightarrow B$

(iii) *Je-li $\vdash A \leftrightarrow B$, pak $T \vdash A$ právě když $T \vdash B$.*

(iv) $\vdash (A_1 \to (A_2 \to \dots (A_n \to B) \dots) \leftrightarrow ((A_1 \wedge A_2 \cdots \wedge A_n) \to B)$

1.5.24 Lemma (o ekvivalenci). *Buď φ formule, a buď φ' formule, která vznikne z φ nahrazením všech výskytů podformulí $\varphi_1, \dots, \varphi_n$ podformulemi $\varphi'_1, \dots, \varphi'_n$. Potom je-li $\vdash \varphi_i \leftrightarrow \varphi'_i$ pro každé $i \leq n$, je také $\vdash \varphi \leftrightarrow \varphi'$.*

Důkaz. Indukcí podle složitosti formule φ. Je-li φ atomická nebo některá z φ_i, není co dokazovat. Je-li φ tvaru $\neg\psi$ a pro formuli ψ již máme $\vdash \psi \leftrightarrow \psi'$, je podle 1.5.23 také $\vdash \psi \to \psi'$, a podle 1.5.18 také $\vdash \varphi' \to \varphi$; podobně pro opačnou implikaci. Je-li φ tvaru $\psi \to \vartheta$ a máme již $\vdash \psi \leftrightarrow \psi'$ a $\vartheta \leftrightarrow \vartheta'$, máme také $\vdash \psi' \to \psi$ a $\vartheta \leftrightarrow \vartheta'$. Přitom $(\psi' \to \psi) \to ((\psi \to \vartheta) \to ((\vartheta \to \vartheta') \to (\psi' \to \vartheta')))$ je dokazatelná formule (skládání implikací jako v 1.5.15), takže dvojím použitím modus ponens máme $(\psi \to \vartheta) \to (\psi' \to \vartheta')$; podobně naopak. □

Úplnost výrokové logiky Dokážeme větu o úplnosti Hilbertova systému pro výrokovou logiku, která popisuje soulad mezi výrokovou pravdivostí a dokazatelností: výroková logika dokazuje právě a jen všechny tautologie. Hilbertovské axiomy a odvozovací pravidla tedy plně charakterisují pravdivost pomocí čistě formálních, syntaktických prostředků.

1.5.25 Lemma (o neutrální formuli). *Buď T výroková teorie a buďte φ a ψ výrokové formule. Je-li $T, \varphi \vdash \psi$ a zároveň $T, \neg\varphi \vdash \psi$, pak je také $T \vdash \psi$.*

Důkaz. Z předpokladu $T, \neg\varphi \vdash \psi$ máme podle 1.5.18 s použitím modus ponens také $T \vdash \neg\psi \to \neg\neg\varphi$, a podle věty o dedukci tedy $T, \neg\psi \vdash \neg\neg\varphi$. Podle 1.5.17 tedy s použitím modus ponens máme $T, \neg\psi \vdash \varphi$. Zároveň z předpokladu $T, \varphi \vdash \psi$ máme podle věty o dedukci $T \vdash \varphi \to \psi$. Dalším použitím modus ponens tedy $T, \neg\psi \vdash \psi$ a podle věty o dedukci je $T \vdash \neg\psi \to \psi$. Přitom podle 1.5.19 je $\vdash (\neg\psi \to \psi) \to \psi$, takže opět pomocí modus ponens konečně máme $T \vdash \psi$. □

V dalším využijeme následující značení. Je-li φ výroková formule a je-li v nějaké pravdivostní ohodnocení, pak φ^v značí buďto formuli φ, pokud $v(\varphi) = 1$, nebo formuli $\neg\varphi$, pokud $v(\varphi) = 0$. V každém případě tedy $v(\varphi^v) = 1$.

1.5.26 Lemma. *Buď φ výroková formule a buďte $A_1, \dots, A_n$ právě všechny její prvotní podformule. Potom pro každé ohodnocení je $A_1^v, \dots, A_n^v \vdash \varphi^v$.*

Důkaz. Je-li sama φ prvotní formule, je tvrzení samozřejmé. Je-li φ tvaru $\neg\psi$ a pro ψ je již tvrzení dokázáno, uvažme dvě možnosti. Pokud $v(\psi) = 0$, je ψ^v formule $\neg\psi$, a podle indukčního předpokladu je $A_1^v, \dots, A_n^v \vdash \neg\psi$; přitom $\neg\psi$ je formule φ^v. Pokud $v(\psi) = 1$, je ψ^v formule ψ a podle indukčního předpokladu již máme $A_1^v, \dots, A_n^v \vdash \psi$. Podle 1.5.17 je $\vdash \psi \to \neg\neg\psi$ a tedy pomocí modus ponens máme $A_1^v, \dots, A_n^v \vdash \neg\neg\psi$. Přitom $\neg\neg\psi$ je formule φ^v.

Konečně je-li φ tvaru $\psi \to \vartheta$ a pro formule ψ a ϑ je již tvrzení dokázáno, rozlišíme opět případy podle hodnot $v(\psi)$ a $v(\vartheta)$. V případě $v(\psi) = 0$ je ψ^v formule $\neg\psi$ a φ^v je formule $\psi \to \vartheta$. Přitom podle 1.5.4 je pomocí věty o dedukci $\neg\psi \vdash \psi \to \vartheta$, takže tvrzení plyne z indukčního předpokladu. V případě $v(\psi) = 1 = v(\vartheta)$ je $v(\varphi) = 1$, takže φ^v je formule $\psi \to \vartheta$. Přitom podle prvního axiomu a věty o dedukci je $\vartheta \vdash \psi \to \vartheta$ a vlevo stojí ϑ^v. V případě $v(\psi) = 1, v(\vartheta) = 0$ je $v(\varphi) = 0$, takže φ^v je formule $\neg\varphi$, tj. formule $\neg(\psi \to \vartheta)$. Podle 1.5.18 a věty o dedukci je $\psi, \neg\vartheta \vdash \neg(\psi \to \vartheta)$; přitom vlevo stojí ψ^v a ϑ^v a vpravo je φ^v. □

1.5.27 Věta (Post)**.** *Každá výroková tautologie je dokazatelná.*

Důkaz. Buď φ tautologie, buďte $A_1, \dots, A_n$ všechny její prvotní podformule. Pro libovolné ohodnocení v je podle předchozího lemmatu $A_1^v, \dots, A_n^v \vdash \varphi$. Buď w pravdivostní ohodnocení, které se shoduje s ohodnocením v všude kromě proměnné A_n, kde dává opačnou hodnotu. I nyní máme $A_1^w, \dots, A_n^w \vdash \varphi$, tj. $A_1^v, A_2^v, \dots, A_{n-1}^v, A_n^w \vdash \varphi$. Je tedy zároveň

$$A_1^v, A_2^v, \dots, A_{n-1}^v, A_n \vdash \varphi$$
$$A_1^v, A_2^v, \dots, A_{n-1}^v, \neg A_n \vdash \varphi$$

a podle 1.5.25 máme $A_1^v, \dots, A_{n-1}^v \vdash \varphi$. Nyní stačí tento postup n-krát zopakovat a získáme $\vdash \varphi$. □

1.5.28 Věta (o úplnosti výrokové logiky)**.** *Pro výrokovou formuli φ a výrokovou teorii T je $T \vdash \varphi$ právě tehdy, když $T \models \varphi$. Specielně výroková logika dokazuje právě a jen tautologie.*

Důkaz. Jeden směr je zněním věty o korektnosti. Buď naopak $T \models \varphi$. Podle věty o kompaktnosti je $T_0 \models \varphi$ už pro nějakou konečnou $T_0 = \{\varphi_1, \dots, \varphi_n\} \subseteq T$. To znamená, že formule $\varphi_1 \to (\varphi_2 \to \dots (\varphi_n \to \varphi) \dots)$ je tautologie, a podle Postovy věty je dokazatelná v logice. Použijeme-li nyní n-krát větu o dedukci, dostáváme $\varphi_1, \dots, \varphi_n \vdash \varphi$, neboli $T_0 \vdash \varphi$, a tedy také $T \vdash \varphi$. □

Dokázali jsme větu, která popisuje souvislost mezi *pravdivostí* a *dokazatelností* ve výrokové logice. Tato věta má několik důsledků, resp. ekvivalentů.

1.5.29 Věta. *Výroková teorie je bezesporná právě tehdy, když je splnitelná.*

Důkaz. Jeden směr je 1.5.12. Je-li T nesplnitelná, pak podle věty o kompaktnosti je nesplnitelná už nějaká konečná $\{\varphi_1, \dots, \varphi_n\} \subseteq T$. Formule $\neg\varphi_1 \vee \dots \vee \neg\varphi_n$ je tedy tautologie, označme ji φ. Podle věty o úplnosti je $\vdash \varphi$, tedy i $T \vdash \varphi$. Zároveň $T \vdash \varphi_i$ pro každou φ_i, tedy podle 1.5.22 je i $T \vdash \varphi_1 \wedge \dots \wedge \varphi_n$. Přitom tato formule je ekvivalentní s $\neg\varphi$, a podle 1.5.23 máme také $T \vdash \neg\varphi$. □

V předchozím důkaze jsme použili větu o kompaktnosti, kterou naopak snadno získáme jako důsledek. Buď totiž T nějaká nesplnitelná teorie. Podle předchozí věty je T je sporná. Důkaz sporu v T ale používá jen nějakou konečnou část $T_0 \subseteq T$, neboť důkaz je konečná posloupnost. Tedy už konečná $T_0 \subseteq T$ je sporná, a tedy nesplnitelná.

Zároveň předchozí věta dokazuje větu o úplnosti: je-li $T \models \varphi$, je $T, \neg\varphi$ nesplnitelná, a tedy sporná. Tedy již nějaká konečná $T_0, \neg\varphi$ je sporná, což podle 1.5.21 znamená právě tolik, že $T_0 \vdash \varphi$.

Rozhodnutelnost výrokové logiky Otázka po dokazatelnosti výrokové formule je podle Postovy věty totéž, jako otázka po její pravdivosti. Přitom o pravdivosti výrokové formule dokážeme efektivně rozhodnout. To znamená, že existuje procedura, která *efektivně rozhoduje o dokazatelnosti* každé výrokové formule. Říkáme, že výroková logika je *rozhodnutelná*.

V další kapitole se budeme zabývat predikátovou logikou, která také má svou větu o úplnosti, ale rozhodnutelná není. V predikátové logice není k dispozici žádná analogie pravdivostních tabulek, která byla umožnila ověřit pravdivost formule při všech ohodnoceních.

Alternativní systémy Popsali jsme Hilbertův odvozovací systém pro výrokovou logiku, který je zavedeným standardem, a budeme jej v dalším používat. Jako dodatek popíšeme několik dalších formálních systémů, ze kterých se Hilbertův systém postupně vytříbil.[18]

1.5.30 Cvičení. Ukažte, že formální systém výrokové logiky[19] s následujícími axiomy a odvozovacím pravidlem modus ponens je stejně silný jako Hilbertův.

F1: $\varphi \to (\psi \to \varphi)$

F2: $(\varphi \to (\psi \to \vartheta)) \to ((\varphi \to \psi) \to (\varphi \to \vartheta))$

F3: $(\varphi \to (\psi \to \vartheta)) \to (\psi \to (\varphi \to \vartheta))$

F4: $(\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$

F5: $\varphi \to \neg\neg\varphi$

F6: $\neg\neg\varphi \to \varphi$

1.5.31 Cvičení. Ukažte, že formální systém výrokové logiky[20] s následujícími axiomy a pravidlem modus ponens je stejně silný jako klasická výroková logika.

G1: $\varphi \to (\psi \to \varphi)$

G2: $(\varphi \to (\varphi \to \psi)) \to (\varphi \to \psi)$

G3: $(\varphi \to (\psi \to \vartheta)) \to (\psi \to (\varphi \to \vartheta))$

G4: $(\psi \to \vartheta) \to ((\varphi \to \psi) \to (\varphi \to \vartheta))$

[18] Zájemce o historický vývoj odkazujeme na [T].

[19] G. Frege, *Begriffsschrift*, Halle, 1879

[20] D. Hilbert, *Die logischen Grundlagen der Mathematik*, Math. Annalen 88 (1923), 151–165

G5: $\varphi \to (\neg\varphi \to \psi)$

G6: $(\varphi \to \psi) \to ((\neg\varphi \to \psi) \to \psi)$

1.5.32 Cvičení. (a) Výroková teorie T je *nezávislá*, pokud pro každou vlastní podmnožinu $S \subset T$ existuje nějaká formule dokazatelná v T, která není dokazatelná v S. Ukažte, že teorie T je nezávislá právě tehdy, když pro libovolnou formuli φ z T je $(T \setminus \{\varphi\}) \cup \{\neg\varphi\}$ bezesporná teorie.

(b) Ukažte, že systémy F1–F6 a G1–G6 jsou závislé.

(c) Ukažte, že axiomy H1–H3 jsou nezávislé. Z úplnosti a nezávislosti plyne, že kdybychom ke klasické výrokové logice přidali další axiom nebo odvozovací pravidlo, vzniklý systém by byl redundantní. Z větší množiny axiomů a pomocí více pravidel bychom snadněji dokazovali věty *ve výrokové logice*; s úplným nezávislým systémem naopak stručněji dokážeme různá tvrzení *o výrokové logice*.

(d) Ukažte, že formální systém výrokové logiky[21] s následujícími axiomy a odvozovacím pravidlem modus ponens je nezávislý, a stejně silný jako Hilbertův.

L1: $(\varphi \to \psi) \to ((\psi \to \vartheta) \to (\varphi \to \vartheta))$

L2: $(\neg\varphi \to \varphi) \to \varphi$

L3: $\varphi \to (\neg\varphi \to \psi)$

1.5.33 Cvičení. Uvažte formální odvozovací systém pro výrokovou logiku, jehož axiomy jsou hilbertovské formule H1–H3, přeložené ovšem do jazyka $\{\neg, \vee\}$, a jediným odvozovacím pravidlem je *pravidlo rezoluce*: z formulí $(\varphi \vee \psi_1 \vee \cdots \vee \psi_n)$ a $(\neg\varphi \vee \vartheta_1 \vee \cdots \vee \vartheta_m)$ odvoď formuli $(\psi_1 \vee \cdots \vee \psi_n \vee \vartheta_1 \vee \cdots \vee \vartheta_m)$. Korektnost tohoto systému plyne z 1.4.6. Ukažte, že tento systém je zároveň úplný.

1.5.34 Cvičení. Přirozeným požadavkem při budování formálního odvozovacího systému je dosáhnout co nejúspornějšího tvaru. Proto výstavba Hilbertova systému začíná redukcí jazyka na minimální univerzální množinu spojek $\{\neg, \to\}$. V tomto redukovaném jazyce pak formulujeme úplnou axiomatiku. Podobně rezoluční kalkul používá minimální jazyk spojek $\neg$ a $\vee$. Podle 1.3.2 víme, že existují univerzální množiny sestávající z jediné spojky; tedy systém formulovaný v jazyce o *dvou* spojkách lze formálně stále považovat za zbytečně složitý.

Uvažte formální systém[22] v jazyce $\uparrow$, jehož axiomy jsou instance formule

$$(P \uparrow (Q \uparrow R)) \uparrow ((S \uparrow (S \uparrow S)) \uparrow ((T \uparrow Q) \uparrow ((P \uparrow T) \uparrow (P \uparrow T))))$$

a jehož jediným odvozovacím pravidlem je *pravidlo odmítnutí*: z formulí P a $P \uparrow (Q \uparrow R)$ odvoď formuli R. Ukažte, že tento systém je stejně silný jako Hilbertův, tedy že dokazuje právě všechny tautologie.

[21] J. Łukasiewicz, *Elementy logiki matematycznej*, Warszawa, 1929

[22] J. G. P. Nicod, *A reduction in the number of primitive propositions of logic*, Proc. Cambdridge Phil. Soc. 19 (1917), 32–41

Kapitola 2

Predikátová logika

Jazyk matematiky je jemnější než jazyk výrokové logiky, a v této kapitole jej prozkoumáme podrobněji. Zavedeme relační a funkční symboly, potřebné pro popis matematických struktur, a popíšeme jejich syntax a sémantiku. Hilbertovské axiomy a odvozovací pravidla rozšíříme na tyto nové symboly, a popíšeme odvozovací systém predikátové logiky, který je formálním rámcem matematiky. O tomto systému ukážeme, že je korektní a úplný. Dokážeme větu o kompaktnosti predikátové logiky, a předvedeme některé její důsledky.

2.1 Jazyk predikátové logiky

Zkoumáme-li formální jazyk, který má sloužit k popisu nějakých struktur, musíme se nejdříve ptát, co všechno chceme tímto jazykem zachytit. Jistě například chceme mít možnost *pojmenovat* některé konkrétní objekty. K tomu slouží v jazyce predikátové logiky *konstantní symboly* neboli *konstanty*. Například symboly **0** a **1** v aritmetice nebo π v analýze jsou jména jistých význačných čísel, konstanty *sin* či *exp* pojmenovávají jisté konkrétní funkce, atd.

Kromě jmen konkrétních objektů potřebujeme mít v jazyce i obecná jména pro objekty, když budeme chtít mluvit o *nějakém* čísle, prostoru, permutaci, atd. K tomu slouží v jazyce predikátové logiky *proměnné*. Budeme se držet tradice a používat jako proměnné písmena latinské abecedy ($x, y, z, \dots$), případně s indexy ($x_1, x_2, x_3, \dots$) apod.

Potřebujeme se také vyjadřovat o *vlastnostech* jednotlivých objektů a o *vztazích* mezi nimi. Chceme například mluvit o prvočíselnosti, o dělitelnosti jednoho čísla druhým, o různých uspořádáních, o kolmosti přímek, o symetrii grafů, o ekvivalenci gramatik, apod. K tomu slouží v jazyce predikátové logiky *relační symboly* neboli *predikáty*. Například $\leq$ je obvyklým predikátovým symbolem v teorii uspořádání, symbol $\parallel$ může značit rovnoběžnost v geometrii, symbol $\in$ obvykle značí náležení jedné množiny do druhé, atd. Jednotlivé predikáty se liší svou *četností*: *unární* predikáty mluví o vlastnosti jednoho objektu (prvočíselnost, spojitost), *binární* o vztahu dvou (kolmost přímek, náležení jedné množiny do druhé, dělitelnost jednoho čísla druhým), další o třech, atd.

Dále chceme mít možnost popsat různé *operace*, které s objekty provádíme: chceme mluvit o násobení čísel, skládání permutací, řetězení slov, sjednocení

množin, převracení zlomků, atd. K tomu slouží v predikátové logice *funkční symboly*, které se opět liší svou četností, podobně jako u relací.

Své promluvy potřebujeme *kvantifikovat*: někdy se chceme vyjádřit o *všech* zkoumaných objektech („každý vektor z lineární báze …“), jindy jen tvrdíme, že objekt s nějakou vlastností *existuje*. K tomu slouží *kvantifikátory*, v klasické logice $\forall$ (*obecný* či *velký*) a $\exists$ (*existenční* či *malý*).[1]

Své promluvy chceme také skládat do logických celků pomocí spojek, jak jsme podrobně popsali v části o výrokové logice.

2.1.1 Definice. *Jazyk predikátové logiky* sestává z

(a) množiny *konstantních symbolů*

(b) množiny *funkčních symbolů*, u každého je dána jeho četnost

(c) množiny *relačních symbolů*, u každého je dána jeho četnost

(d) neomezené množiny symbolů pro *proměnné*

(e) *výrokových spojek* $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

(f) *kvantifikátorů* $\forall$ a $\exists$

(g) závorek {[()]} apod. pro lepší čitelnost

Předpokládáme zároveň, že výše uvedené množiny symbolů jsou navzájem disjunktní, a tedy žádná proměnná není zároveň konstantou, závorka není zároveň predikátem, spojka není jménem funkce atd.[2]

Symboly (a), (b), (c) jsou pro ten který jazyk specifické, a odrážejí oblast, o které chceme daným jazykem mluvit (viz příklady níže). Nazývají se *mimologické* nebo *speciální symboly*. Ostatní symboly jsou společné všem predikátovým jazykům, nazývají se *logické symboly*: proměnné, spojky, atd potřebujeme v každém jazyce, ať už jím budeme mluvit o čemkoli.

Zvláštní postavení má binární relační symbol $=$ pro rovnost. Obvykle se řadí též mezi logické symboly, a jeho chování popisují speciální axiomy. Pokud v dalším neřekneme jinak, je predikát $=$ přítomen v každém uvažovaném jazyce; jedná se pak o *jazyk s rovností*.

2.1.2 Příklad. (a) *Jazyk teorie množin* obsahuje jediný binární predikát $\in$.

(b) *Jazyk orientovaných grafů* obsahuje jediný binární relační symbol $\rightarrow$.

(c) *Jazyk teorie uspořádání* obsahuje jediný binární relační symbol $<$.

(d) *Jazyk teorie grup* obsahuje binární funkční symbol $*$, unární funkční symbol $^{-1}$ a konstantu $\mathbf{1}$. Žádné relační symboly nemá.

(e) *Jazyk aritmetiky* obsahuje konstanty $\mathbf{0}$ a $\mathbf{1}$, binární relační symbol $<$, unární funkční symbol S, a binární funkční symboly $+$ a $*$.

Jednotlivé jazyky jsou vytvořeny se záměrem popsat nějakou oblast matematiky. Například jazyk teorie uspořádání s jediným predikátem $<$ je vhodný pro popis uspořádaných množin, zatímco při popisu aritmetiky bychom s ním těžko vystačili. Podobně jazykem teorie grup lze popsat vlastnosti grupové operace, inverzních prvků a neutrálního prvku, ale k popisu grafu vhodný není.

[1] Převrácená písmena A a E jsou symboly pro slova *alle* a *existiert*, resp. *all* a *exists*.

[2] Podobně jako programovací jazyk typicky nedovoluje proměnnou jménem `while` apod.

V příkladech jsme uvedli jazyky v jejich základní podobě. Je běžnou praxí jazyk postupně rozšiřovat o nové symboly spolu s tím, jak objevujeme nové vlastnosti zkoumaných objektů. Náležitosti takového rozšiřování popíšeme ve 2.5.

2.1.3 Definice. Buď $\mathcal{L}$ jazyk predikátové logiky. Potom *term* jazyka $\mathcal{L}$ je každý konečný výraz, který vznikne konečnou aplikací následujících pravidel:

(a) Každá konstanta jazyka $\mathcal{L}$ je term.

(b) Každá proměnná jazyka $\mathcal{L}$ je term.

(c) Jsou-li $t_1, \ldots, t_n$ termy jazyka $\mathcal{L}$, a je-li f nějaký n-ární funkční symbol jazyka $\mathcal{L}$, pak také $f(t_1, \ldots, t_n)$ je term.

Termy popisují provádění operací, resp. pojmenovávají objekty, které operacemi vznikají. U binárních operací je zvykem psát funkční symbol mezi operandy — píšeme například $x + y$ místo $+(x, y)$ apod.

2.1.4 Příklad. Základní jazyky teorie množin, teorie grafů a teorie uspořádání žádné termy kromě proměnných nemají. Výrazy $\mathbf{1}$, $x * \mathbf{1}$, $x * y$, x^{-1}, $(x * y)^{-1}$, $y * x * y^{-1}$ jsou termy jazyka grup. Výrazy $\mathbf{1}$, $x + y$, $x + \mathbf{1}$, $y * \mathbf{0}$, $x * (y + z)$, $S(x * (\mathbf{1} + \mathbf{1}))$ jsou termy jazyka aritmetiky.

2.1.5 Definice. Buď $\mathcal{L}$ jazyk predikátové logiky. Potom *formule* jazyka $\mathcal{L}$ je každý výraz, který vznikne konečnou aplikací následujících pravidel:

(a) Jsou-li t_1 a t_2 termy jazyka $\mathcal{L}$, potom výraz $t_1 = t_2$ je formule.

(b) Jsou-li $t_1, \ldots, t_n$ termy jazyka $\mathcal{L}$ a je-li R nějaký n-ární relační symbol jazyka $\mathcal{L}$, potom výraz $R(t_1, \ldots, t_n)$ je formule.

(c) Jsou-li φ, ψ formule, pak také následující výrazy jsou formule: $(\neg\varphi), (\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \to \psi), (\varphi \leftrightarrow \psi)$.

(d) Je-li x proměnná a φ formule, pak také výrazy $(\forall x)\varphi$ a $(\exists x)\varphi$ jsou formule.

Podformulí dané formule je pak každý podřetězec, který je sám formulí.

Formule tvaru (a) a (b) se nazývají *atomické* – jsou to nejjednoduší promluvy, které lze v daném jazyce učinit. Zřejmě atomické jsou právě ty formule, které nemají žádné vlastní podformule. Formule z bodu (c) jsou vytvořené z jednodušších formulí pomocí výrokových spojek, které jsme podrobně zkoumali v předchozí kapitole. Jazyk predikátové logiky je bohatší a jemnější než jazyk výrokové logiky: spojkami nyní spojujeme výrazy, které mají vlastní vnitřní strukturu, nejsou to již dále nedělitelné symboly.

Formule $(\forall x)\varphi$ a $(\exists x)\varphi$ z bodu (d) čteme „pro všechna x (platí) φ“ a „existuje x tak, že (platí) φ.“ Důležitou vlastností jazyka klasické logiky je to, že umožňuje kvantifikovat pouze proměnné, tedy jednotlivé objekty, nikoli však množiny objektů, vlastnosti, množiny vlastností, atd. Jedná se o *jazyk prvního řádu*. Jazyky vyšších řádů zkoumat nebudeme.[3]

[3] Jazyky vyšších řádů mají kromě proměnných pro jednotlivé objekty také proměnné pro množiny objektů či několik druhů proměnných pro různé druhy objektů, například proměnné pro přirozená čísla. Takové logiky se od klasické predikátové logiky podstatně liší; například logika druhého řádu nesplňuje větu o kompaktnosti. V úvodu jsme popsali, jak lze omezení dané jazykem prvního řádu v matematice obejít prostřednictvím teorie množin.

Tak jako u binárních funkčních symbolů, je u binárních relačních symbolů zvykem používat infixní notaci a psát xRy místo $R(x,y)$, jak požaduje definice. Píšeme například $x < y$ místo $<(x,y)$ či $x \in y$ místo $\in(x,y)$, apod. Negace atomických formulí zapisujeme obvykle jako $x \neq y$, $x \notin y$, $x \not< y$ apod., místo formálně správných $\neg(x = y)$, $\neg(x \in y)$, $\neg(x < y)$. Běžně také píšeme $(\forall x, y)\varphi$ místo $(\forall x)(\forall y)\varphi$, $(\exists x, y)\varphi$ místo $(\exists x)(\exists y)\varphi$ a podobně.

Ve výrokové logice jsme přijali úmluvu o vazební síle jednotlivých spojek, abychom zjednodušili psaní závorek. Podle definice formule jazyka predikátové logiky se kvantifikátor váže k následující podformuli silněji než jakákoli výroková spojka. Tedy například $(\forall x)\varphi \to \psi$ znamená $((\forall x)\varphi) \to \psi$, nikoli $(\forall x)(\varphi \to \psi)$, podobně $(\forall x)(\exists y)\varphi \wedge \psi$ znamená $((\forall x)(\exists y)\varphi) \wedge \psi$, nikoli $(\forall x)(\exists y)(\varphi \wedge \psi)$, atd.

2.1.6 Příklad. (a) Následující výrazy jsou formulemi teorie množin: $x \in y$, $x \notin y$, $(\forall x)(x \notin x)$, $(\forall x)(x \notin y)$, $(\exists y)(\forall x)(x \notin y)$, $(\forall x)((x \in y) \to (x \in z))$, $(\forall x)(\forall y)((\forall z)(z \in x \leftrightarrow z \in y) \to (x = y))$, $(\forall t)((t \in z) \leftrightarrow ((t \in x) \vee (t \in y)))$, $(\forall t)((t \in z) \leftrightarrow ((t \in x) \wedge (t \in y)))$, $(\forall t)((t \in z) \leftrightarrow (\forall u)(u \in t \to u \in x))$.

(b) Následující výrazy jsou formulemi jazyka[4] orientovaných grafů: $x \to y$, $(\forall x)(x \not\to x)$, $(\exists x)(\forall y)(x \to y)$, $(\forall x)(\forall y)(x \to y)$, $(\forall x)(\forall y)(x \to y \leftrightarrow y \to x)$, $(\exists x)(\exists y)(\exists z)(x \to y \wedge y \to z \wedge z \to x)$, $(\exists x)(\exists y)(\forall z)(x \to z \vee y \to z)$.

(c) Následující výrazy jsou formulemi jazyka teorie uspořádání: $x < y$, $(\forall x)(x \not< x)$, $(\forall x)(\forall y)(\forall z)((x < y \wedge y < z) \to (x < z))$, $\neg(x < y \wedge y < x)$, $(\exists x)(\forall y)(x < y)$, $(\exists z)(x < z \wedge z < y)$, $(\forall x)(\forall y)[(x < y) \to (\exists z)(x < z \wedge z < y)]$, $(\exists y)[(x < y) \wedge (\forall z)((x < z) \to (y < z \vee y = z))]$, $(\forall x)(\forall y)(\exists z)(x < z \wedge y < z)$.

(d) Následující výrazy jsou formule jazyka teorie grup: $\mathbf{1} * x = x$, $x * x^{-1} = \mathbf{1}$, $x * x = \mathbf{1}$, $(\forall x)(\mathbf{1} * x = x \wedge x = x * \mathbf{1})$, $(\forall x)(\forall y)(\forall z)(x * (y * z) = (x * y) * z)$, $(x * y)^{-1} = y^{-1} * x^{-1}$, $(\exists y)(y * x * y^{-1} = x)$, $(\forall x)(\forall y)(x * y = y * x)$.

Jazyk teorie grup nemá žádné relační symboly kromě rovnosti, takže nemá jiné atomické formule než rovnosti termů. V jazyce grup je běžné vynechávat symbol pro grupovou operaci a psát prostě xy místo $x * y$ a podobně.

(e) Následující výrazy jsou formulemi jazyka aritmetiky: $x < y$, $S(x) \neq \mathbf{0}$, $x + \mathbf{0} = \mathbf{1} * x$, $x + y = y + x$, $(\exists u)(\exists v)((x * u = y) \wedge (x * v = z))$, $(\exists x)(y = S(x))$, $(\exists u)(x * u = y)$, $(x * y = \mathbf{0}) \to (x = \mathbf{0} \vee y = \mathbf{0})$, $(\exists u)((x * u = y) \wedge (x * u = z))$, $(\forall y)[(\exists z)(x = y * z) \to (y = \mathbf{1} \vee z = \mathbf{1})]$, $(\nexists u)(x = u + u)$, $(\exists u)(x = u * u)$, $(\forall x)(\forall y)(\forall z)(x*(y+z) = (x*y)+(x*z))$, $(\forall x)(\exists y)(x < y \wedge (\exists u)(x = (u+u)+\mathbf{1}))$.

Ve tvrzeních o přirozených číslech se kromě základních kvantifikací „pro všechna čísla“ a „pro nějaké číslo“ často užívají i obraty „pro skoro všechna čísla“ a „pro nekonečně mnoho čísel“. Například obvyklá definice limity v analýze požaduje, aby v každém okolí ležely skoro všechny členy posloupnosti, v aritmetice se dokazuje, že existuje nekonečně mnoho prvočísel, atd. V jazyce aritmetiky se pro takové obraty někdy používají výrazy $(\forall^\infty x)\varphi$ a $(\exists^\infty x)\varphi$, které jsou zkratkami za formule $(\exists y)(\forall x)((x > y) \to \varphi)$ a $(\forall y)(\exists x)((x > y) \wedge \varphi)$.

2.1.7 Cvičení. Napište formule základního jazyka teorie množin, které vyjadřují následující vlastnosti a vztahy mezi množinami: množina x je prázdná; existuje prázdná množina; žádná množina není sama svým prvkem; množina x je podmnožinou množiny y; množiny se stejnými prvky se rovnají; množina

[4] Symbol $\to$ pro orientovanou hranu (šipku) mezi dvěma vrcholy v grafu nemá nic společného s výrokovou spojkou $\to$ (implikace). Tím porušujeme úmluvu o tom, že jednotlivé množiny symbolů jazyka predikátové logiky jsou navzájem disjunktní.

x má právě tři prvky; množina x je sjednocením množin y a z; množina x je průnikem množin y a z; sjednocení množin y a z je nejmenší nadmnožina obou; průnik množin y a z je největší podmnožina obou; ke každým dvěma množinám x a y existuje množina, jejímiž jedinými dvěma prvky jsou právě množiny x a y; ke každé množině existuje množina jejích podmnožin; neexistuje množina, jejímiž prvky by byly všechny množiny.

2.1.8 Cvičení. Napište formule jazyka orientovaných grafů, které vyjadřují následující vlastnosti: žádný vrchol nemá smyčku; každé dva vrcholy jsou spojeny cestou délky nejvýše pět; neexistuje žádná cesta délky tři; z vrcholu x vede hrana do všech ostatních vrcholů; vrchol x je isolovaný; žádný vrchol není isolovaný; existují alespoň dva vrcholy stupně dva; každý vrchol leží na trojúhelníku.

2.1.9 Cvičení. Napište formule jazyka uspořádání, které vyjadřují následující vlastnosti uspořádaných množin: každé dva prvky jsou porovnatelné; mezi každými dvěma porovnatelnými prvky leží další prvek; mezi jistými dvěma porovnatelnými prvky neleží žádný další; každé dva prvky mají společnou majorantu; prvek x je nejmenší (největší); existuje nejmenší (největší) prvek; existuje nejvýše jeden největší prvek; neexistuje nejmenší ani největší prvek; existují nejméně dva maximální prvky; nad každým prvkem existuje maximální prvek; žádné dva maximální prvky nejsou porovnatelné.

2.1.10 Cvičení. Napište formule jazyka aritmetiky, které vyjadřují následující vlastnosti přirozených čísel: x je sudé; x je liché; x je čtverec; x dělí y; x má jen liché dělitele; x je společný dělitel y a z; čísla x a y jsou soudělná; x je největší společný dělitel y a z; x je nejmenší společný násobek y a z; x je prvočíslo; x je mocnina dvou; x je nejmenší prvočíslo z rozkladu čísla y; v rozkladu čísla x se nevyskytují vyšší než první mocniny; každé prvočíslo kromě 2 je liché; prvočísel je nekonečně mnoho; prvočíselných dvojčat[5] je nekonečně mnoho; prvočíselných dvojčat je konečně mnoho; každé číslo má jen konečně mnoho dělitelů; každé sudé číslo kromě 2 je součtem dvou prvočísel; 0 je nejmenší číslo; každé číslo kromě 0 má bezprostředního předchůdce; žádná dvě různá čísla nemají stejného následníka; neexistuje žádné největší číslo.

2.2 Sémantika predikátové logiky

Popsali jsme syntax jazyka predikátové logiky, totiž termy a formule. Nyní popíšeme způsob, jakým je syntaktickým útvarům přiřazen význam, totiž jak jsou speciální symboly jazyka *realizovány* ve strukturách; termy jsou potom jména objektů a formule jsou tvrzení o těchto objektech.

2.2.1 Definice. Buď $\mathcal{L}$ jazyk predikátové logiky. Potom *realizací* jazyka $\mathcal{L}$, nebo též *strukturou* či *modelem* pro $\mathcal{L}$, je neprázdná množina M, opatřená

(i) vyznačeným prvkem $c^{\mathfrak{M}} \in M$ za každou konstantu c;

(ii) n-ární funkcí $f^{\mathfrak{M}} : M^n \to M$ za každý n-ární funkční symbol f;

(iii) n-ární relací $R^{\mathfrak{M}} \subseteq M^n$ za každý n-ární relační symbol R.

[5] Prvočíselná dvojčata jsou sousedící prvočísla, například 17 a 19.

Množina M je *nosnou množinou* neboli *univerzem* modelu, a její prvky jsou *individua*. Říkáme, že struktura $\mathfrak{M} = (M, R^{\mathfrak{M}}, \dots, f^{\mathfrak{M}}, \dots, c^{\mathfrak{M}}, \dots)$ *realizuje* symboly jazyka $\mathcal{L}$ v množině M, a píšeme $\mathfrak{M} \models \mathcal{L}$.

Abychom zdůraznili rozdíl mezi syntaxí a sémantikou, totiž rozdíl mezi symboly jazyka na jedné straně, a jejich realizacemi na straně druhé, budeme například rozlišovat mezi *symbolem* **0** jazyka aritmetiky a přirozeným *číslem* 0, které tento symbol realizuje ve standardním modelu $\mathbb{N}$. Podobně $+$ je *symbol* jazyka aritmetiky, kdežto $+^{\mathbb{N}}$ je *binární funkce* na množině přirozených čísel, což není totéž. Je důležité oddělit symbol a jeho význam, i když u každodenních symbolů jako $+$ může být pohodlné identifikovat symbol s jeho „samozřejmým“ významem. Tentýž symbol však může být v jiné struktuře realizován jinak.

Například v množině $\mathbb{R}^+$ kladných reálných čísel i v množině $\mathbb{Z}$ celých čísel lze přirozeným způsobem realizovat jazyk teorie grup. V prvním případě je symbol $*$ realizován operací násobení, inverzy jsou převrácená čísla, a neutrálním prvkem je reálné číslo 1. V druhém případě je symbol $*$ realizován operací sčítání, inverzy jsou opačná čísla, a neutrálním prvkem je číslo 0.

Realizací jazyka je dán význam jeho konstantním, funkčním a relačním symbolům. Chceme-li přisoudit význam i dalším syntaktickým útvarům, totiž termům a formulím, musíme začít s proměnnými.

2.2.2 Definice. Buď $\mathcal{L}$ jazyk a $\mathfrak{M} = (M, \dots)$ jeho realizace. Potom každé zobrazení e z množiny proměnných jazyka $\mathcal{L}$ do množiny M nazveme *ohodnocením proměnných*. Pro dané ohodnocení e a daný term t jazyka $\mathcal{L}$ definujeme *hodnotu* $t[e] \in M$ termu t při ohodnocení e indukcí podle složitosti následovně:

(a) je-li t konstanta c, buď $t[e]$ prvek $c^{\mathfrak{M}} \in M$

(b) je-li t proměnná x, buď $t[e]$ prvek $e(x) \in M$

(c) je-li t tvaru $f(t_1, \dots, t_n)$, kde f je n-ární funkční symbol realizovaný funkcí $f^{\mathfrak{M}}$, a t_i jsou termy s hodnotami $t_i[e]$, buď $t[e]$ prvek $f^{\mathfrak{M}}(t_1[e], \dots, t_n[e])$. ∎

2.2.3 Lemma. *Buď $\mathfrak{M} \models \mathcal{L}$, buďte e_1 a e_2 ohodnocení, která se shodují na proměnných $x_1, \dots, x_k$. Je-li t nějaký term jazyka $\mathcal{L}$, ve kterém se vyskytují jen proměnné $x_1, \dots, x_k$, potom hodnoty $t[e_1]$ a $t[e_2]$ jsou stejné.*

Je-li dána realizace jazyka a ohodnocení proměnných, můžeme konečně definovat *splňování* formulí. K tomu použijeme následující značení. Je-li e nějaké ohodnocení proměnných v množině M, pak pro proměnnou x a prvek $m \in M$ označíme jako $e(x/m)$ takové ohodnocení, které proměnnou x ohodnocuje prvkem $m \in M$, a všude jinde se shoduje s ohodnocením e.

2.2.4 Definice. Buď $\mathcal{L}$ jazyk prvního řádu, buď $\mathfrak{M} \models \mathcal{L}$, a buď e ohodnocení proměnných v $\mathfrak{M}$. O formuli φ jazyka $\mathcal{L}$ řekneme, že je *splněna v $\mathfrak{M}$ při ohodnocení* e, a píšeme $\mathfrak{M} \models \varphi[e]$, pokud nastává některý z následujících případů.

(a) $\mathfrak{M} \models (t_1 = t_2)[e]$ pokud prvky $t_1[e] \in M$ a $t_2[e] \in M$ jsou identické.

(b) $\mathfrak{M} \models R(t_1, \dots, t_n)[e]$ pokud $(t_1[e], \dots, t_n[e]) \in R^{\mathfrak{M}}$,
kde R je nějaký n-ární relační symbol a $R^{\mathfrak{M}}$ jeho realizace v $\mathfrak{M}$.

(c) $\mathfrak{M} \models (\neg\psi)[e]$ pokud není $\mathfrak{M} \models \psi[e]$; píšeme $\mathfrak{M} \not\models \psi[e]$.

(d) $\mathfrak{M} \models (\psi \wedge \vartheta)[e]$ pokud $\mathfrak{M} \models \psi[e]$ a $\mathfrak{M} \models \vartheta[e]$.

(e) $\mathfrak{M} \models (\psi \vee \vartheta)[e]$ pokud $\mathfrak{M} \models \psi[e]$ nebo $\mathfrak{M} \models \vartheta[e]$.

(f) $\mathfrak{M} \models (\psi \to \vartheta)[e]$ pokud $\mathfrak{M} \not\models \psi[e]$ nebo $\mathfrak{M} \models \vartheta[e]$.

(g) $\mathfrak{M} \models (\psi \leftrightarrow \vartheta)[e]$ pokud je $\mathfrak{M} \models \psi[e]$ právě když $\mathfrak{M} \models \vartheta[e]$.

(h) $\mathfrak{M} \models ((\forall x)\psi)[e]$ pokud pro každé $m \in M$ je $\mathfrak{M} \models \psi[e(x/m)]$.

(i) $\mathfrak{M} \models ((\exists x)\psi)[e]$ pokud pro nějaké $m \in M$ je $\mathfrak{M} \models \psi[e(x/m)]$.

Pokud $\mathfrak{M} \models \varphi[e]$ při všech ohodnoceních e, řekneme, že φ je *splněna v* $\mathfrak{M}$, nebo *pravdivá v* $\mathfrak{M}$, nebo že *platí v* $\mathfrak{M}$, a píšeme $\mathfrak{M} \models \varphi$. Je-li $\mathfrak{M} \models \varphi$ pro každý model $\mathfrak{M}$ jazyka $\mathcal{L}$, řekneme, že formule φ je *logicky platná*, a píšeme $\models \varphi$.

V definici požadujeme, aby predikát $=$ byl vždy realizován tak, jak od rovnosti očekáváme, totiž relací identity. Splňování ostatních atomických formulí je dáno tím, jak struktura $\mathfrak{M}$ realizuje speciální symboly jazyka $\mathcal{L}$. Indukční kroky pro logické spojky a kvantifikátory jsou potom definovány tak, jak chápeme obraty „nebo“, „pro všechny“, atd.

Zřejmě pro každé $\mathfrak{M}, \varphi, e$ je buďto $\mathfrak{M} \models \varphi[e]$ nebo $\mathfrak{M} \models \neg\varphi[e]$, a pravdivost formule při daném ohodnocení závisí jen na hodnotách těch proměnných, které se v ní vyskytují. Ve skutečnosti platí více.

Volné a vázané proměnné Aritmetická formule $(\exists y)(x = y + y)$ je tvrzení o číslu x (totiž *číslo x je sudé*), nikoli tvrzení o číslu y. Proměnná y je ve formuli kvantifikována, proměnná x nikoli; jejich role nejsou rovnocenné.

2.2.5 Definice. Buď φ nějaká formule jazyka $\mathcal{L}$ predikátové logiky. Řekneme, že výskyt proměnné x ve formuli φ je *vázaný*, pokud je částí nějaké podformule tvaru $(\forall x)\psi$ nebo $(\exists x)\psi$. Výskyt proměnné, který není vázaný, je *volný*. Formule bez volných proměnných je *uzavřená formule* neboli *sentence* jazyka $\mathcal{L}$. Formule bez vázaných proměnných je *otevřená*.

Například v aritmetické formuli $(\forall y)(\forall z)(x = y * z \to (x = y \vee x = z))$ je proměnná x volná, kdežto proměnné y, z jsou vázané. Podle následujícího lemmatu záleží splnění této formule pouze na ohodnocení volné proměnné x.

2.2.6 Lemma. *Buď $\mathfrak{M} \models \mathcal{L}$, buďte e_1 a e_2 nějaká dvě ohodnocení, která se shodují na proměnných $x_1, \ldots, x_n$. Je-li φ formule jazyka $\mathcal{L}$, jejíž všechny* volné *proměnné jsou mezi $x_1, \ldots, x_n$, pak je $\mathfrak{M} \models \varphi[e_1]$ právě tehdy, když $\mathfrak{M} \models \varphi[e_2]$.*

Důkaz. Tvrzení dokážeme indukcí podle složitosti formule. Je-li φ atomická, je tvaru $t_1 = t_2$ nebo $R(t_1, \ldots, t_k)$, kde $t_1, \ldots, t_k$ jsou termy a R nějaký predikát jazyka $\mathcal{L}$. V tom případě jsou všechny proměnné ve φ volné, takže ohodnocení e_1, e_2 se shodují na všech proměnných ve φ. Všechny zúčastněné termy t_i jsou tedy při e_1, e_2 realizovány stejnými individui, takže podle definice splňování atomických formulí znamená $\mathfrak{M} \models \varphi[e_1]$ právě tolik co $\mathfrak{M} \models \varphi[e_2]$. Indukční krok pro výrokové spojky je zřejmý. Je-li φ tvaru $(\forall x)\psi$, potom $\mathfrak{M} \models \varphi[e_1]$ znamená podle definice právě tolik, že $\mathfrak{M} \models \psi[e_1(x/m)]$ pro každé $m \in M$.

Přitom pro každé $m \in M$ se ohodnocení $e_1(x/m)$ a $e_2(x/m)$ shodují na volných proměnných formule ψ: je-li x_i volná ve φ, pak tuto shodu předpokládáme, je-li x_i proměnná x, plyne shoda z definice $e(x/m)$. Podle indukčního předpokladu je tedy $\mathfrak{M} \models \psi[e_1(x/m)]$ právě když $\mathfrak{M} \models \psi[e_2(x/m)]$. Tedy $\mathfrak{M} \models \psi[e_2(x/m)]$ pro každé $m \in M$, neboli $\mathfrak{M} \models \varphi[e_2]$. Indukční krok pro $\exists$ je analogický. □

Například aritmetická formule $(\forall y)(\forall z)(x = y * z \to (x = y \vee x = z))$ je ve standardním modelu $\mathbb{N}$ s obvyklými operacemi splněna při takových ohodnoceních e, při kterých je volná proměnná x ohodnocena nějakým prvočíslem $e(x)$, zatímco na ohodnocení vázaných proměnných y, z nezáleží. V dalším budeme běžně ohodnocovat jen volné proměnné. Krajním případem je uzavřená formule, která žádné volné proměnné nemá, a v dané struktuře tedy buďto platí při každém ohodnocení, nebo při žádném.

Proměnná může mít ve formuli volné i vázané výskyty, například ve formuli $(x * y = \mathbf{1}) \wedge ((\forall x)(\forall y)(x * y = y * x))$. Taková situace je jistým způsobem nežádoucí,[6] lze se jí však vždy vyhnout: podle předchozího pozorování lze vázané výskyty proměnných nahradit nějakými jinými, dosud nepoužitými proměnnými. Například formuli $(x * y = \mathbf{1}) \wedge ((\forall p)(\forall q)(p * q = q * p))$ splňují právě ta ohodnocení, která splňují předchozí formuli.

2.2.7 Lemma. *Ke každé formuli φ jazyka $\mathcal{L}$ existuje formule ψ, ve které žádná proměnná není zároveň volná i vázaná, a pro každý model $\mathfrak{M} \models \mathcal{L}$ a každé ohodnocení e platí $\mathfrak{M} \models \varphi[e]$ právě když $\mathfrak{M} \models \psi[e]$.*

Napíšeme-li v dalším $\varphi(x_1, \ldots, x_n)$ místo φ, chceme tím říci, že všechny volné proměnné formule φ jsou mezi proměnnými $x_1, \ldots, x_n$, a žádná z nich se ve φ nevyskytuje zároveň vázaně.[7] Je-li potom $\varphi(x_1, \ldots, x_n)$ nějaká formule, a jsou-li $a_1, \ldots, a_n$ prvky nějaké dané struktury $\mathfrak{M}$, budeme někdy stručně psát $\mathfrak{M} \models \varphi[a_1, \ldots, a_n]$ místo $\mathfrak{M} \models \varphi[e(x_1/a_1, \ldots, x_n/a_n)]$.

2.2.8 Cvičení. Pro každou z následujících formulí jazyka teorie orientovaných grafů popište *všechna* ohodnocení proměnných v množině $M = \{0, 1, 2, 3\}$ opatřené relací $\{(0,0), (0,1), (0,2), (0,3), (1,3), (2,3), (3,0), (3,3)\}$, při kterých daná formule je (resp. není) splněna — nebo ukažte, že takové ohodnocení neexistuje.

$(x \to x)$; $\neg(x \to x)$; $(\exists y)(x \to y)$; $(\exists y)(y \to x)$; $(\forall y)(x \to y)$; $(\forall y)(y \to x)$; $(\exists u)(x \to u \wedge u \to y)$; $(\exists u)(x \to u \wedge u \to u)$; $(\exists u)(\exists v)(x \to u \wedge u \to v \wedge v \to y)$.

Dále rozhodněte, zda jsou v této struktuře splněny následující sentence.

$(\forall x)(\exists y)((x \to y) \wedge (y \to x))$; $(\forall x)(\exists y)(\exists z)((x \to y) \wedge (y \to z) \wedge (z \to x))$; $(\forall x)(\forall y)(\exists u)(\exists v)((x \to u) \wedge (u \to v) \wedge (v \to y))$; $(\forall x)(\forall y)((x \to y) \vee (y \to x))$.

2.2.9 Cvičení. Pro každou z následujících formulí jazyka uspořádání popište nějaké ohodnocení proměnných, při kterém daná formule je (resp. není) splněna ve struktuře $(\mathbb{N}, <)$, $(\mathbb{N}, |)$, $(\mathbb{N}, \mathbb{N} \times \mathbb{N})$, $(\mathbb{Z}, <)$, $(\mathbb{Q}, <)$, $(\mathbb{R}, <)$, $(P(\mathbb{N}), \subset)$ — nebo ukažte, že takové ohodnocení neexistuje: $(\forall y)(x < y \vee x = y)$; $(\forall y)\neg(x < y)$; $(\exists z)(x < z \wedge z < y)$; $(x < y) \wedge \neg(\exists z)(x < z \wedge z < y)$; $\neg(x < y \vee y < x)$.

Rozhodněte, které z následujících sentencí v těchto strukturách platí:

[6] Podobně jako lokální proměnná a globální proměnná stejného jména v kódu programu.

[7] Netvrdíme tím nutně, že každá z proměnných $x_1, \ldots, x_n$ skutečně má volný výskyt ve formuli φ. To je podobné obvyklému zápisu polynomu $p(x_1, \ldots, x_n)$ v algebře, kterým zdůrazňujeme jména proměnných, ale netvrdíme, že u každé stojí nutně nenulový koeficient.

$(\forall x)(\forall y)(\forall z)(x < y \wedge y < z \to x < z)$; $(\forall x)\neg(x < x)$; $(\forall x)(\exists y)(x < y)$; $(\forall x)(\forall y)(x < y \to (\exists z)(x < z \wedge y < z))$; $(\forall x)(\forall y)(x < y \vee x = y \vee y < x)$.

2.2.10 Cvičení. Pro každou z následujících formulí jazyka teorie grup popište nějaké ohodnocení proměnných ve strukturách $(\mathbb{Z}, +, -, 0)$, $(\mathbb{Q}^+, *, ^{-1}, 1)$, při kterém daná formule je (resp. není) splněna — nebo ukažte, že takové ohodnocení neexistuje: $\mathbf{1}*x = x$; $(\exists y)(y*y = x)$; $(\exists y)(y*y*y = x)$; $(x*y)^{-1} = x^{-1}*y^{-1}$; $(\exists y)(y * x * y^{-1} = x)$. Rozhodněte, zda v těchto strukturách platí sentence $(\forall x)[(\forall y)(x * y = y) \to (x = \mathbf{1})]$.

2.2.11 Cvičení. Pro každou z následujících formulí jazyka aritmetiky popište nějaké ohodnocení proměnných v množině $\mathbb{N}$ s obvyklým uspořádáním a obvyklými operacemi součtu, součinu a následníka, při kterém daná formule je (resp. není) splněna — nebo ukažte, že takové ohodnocení neexistuje.

$(\exists x)(y = S(x))$; $(\exists u)(x * u = y)$; $(\exists x)(\exists y)(\exists z)((u = x * z) \wedge (v = y * z))$; $(\exists u)(\exists v)((x * u = y) \wedge (x * v = z))$; $(\forall y)(\forall z)(x = y * z) \to (y = \mathbf{1} \vee z = \mathbf{1})$; $(\exists y)(x < y \wedge (\exists u)(y = (u + u) + \mathbf{1}))$; $(\forall y)((\exists u)(y = u * u) \to (y < x))$.

2.2.12 Cvičení. Napište sentenci jazyka $\{+, *, \mathbf{0}, \mathbf{1}\}$, která (a) neplatí v $\mathbb{N}$, ale platí v $\mathbb{Z}$; (b) neplatí v $\mathbb{Z}$, ale platí v $\mathbb{Q}$; (c) neplatí v $\mathbb{Q}$, ale platí v $\mathbb{R}$; (d) neplatí v $\mathbb{R}$, ale platí v $\mathbb{C}$. Všechny číselné obory uvažujeme s obvyklými operacemi.

2.2.13 Cvičení. Uvažte jazyk s jediným unárním predikátem p a rozhodněte, které modely tohoto jazyka splňují sentenci $(\forall x)(\forall y)[x = y \vee (p(x) \wedge \neg p(y))]$.

2.2.14 Cvičení ([Šv]). Uvažte následující sentence v jazyce s binárním relačním symbolem $\preceq$ a dvěma binárními funkčními symboly $\otimes$ a $\oplus$:

(i) $(\forall x)(\forall y)(x \otimes y \preceq x \wedge x \otimes y \preceq y)$

(ii) $(\forall x)(\forall y)(x \preceq y \leftrightarrow (\exists z)(x \oplus z = y))$

(iii) $(\forall x)(\forall y)(\forall z)(x \oplus z \preceq y \oplus z \to x \preceq y)$

(iv) $(\forall x)(\forall y)(\forall z)((x \oplus y) \oplus z = x \oplus (y \oplus z))$

(v) $(\forall x)(\forall y)(\forall z)(z \preceq x \wedge z \preceq y \to z \preceq x \otimes y)$

Rozhodněte, které z těchto sentencí jsou splněny v následujících strukturách:

(a) množina přirozených čísel s obvyklým uspořádáním a operacemi;

(b) interval $[0, 1]$ s obvyklým uspořádáním a operacemi součinu a průměru;

(c) interval $(-1, 1)$ s obvyklým uspořádáním a operacemi součinu a průměru;

(d) množina $\mathbb{N}$, kde $\preceq$ je dělitelnost, $\oplus$ součin a $\otimes$ je největší společný dělitel;

(e) množina $P(\mathbb{N})$, kde $\preceq$ je inkluze, $\oplus$ je sjednocení a $\otimes$ je průnik.

2.2.15 Cvičení. Množina $A \subseteq \mathbb{N}$ tvoří *spektrum*, pokud existuje sentence φ nějakého jazyka $\mathcal{L}$, jejíž konečné modely mají právě mohutnosti $n \in A$. Napište sentence, které ukazují, že (a) každá konečná množina, jakož i množina všech (b) čtverců (c) prvočísel (d) mocnin dvojky tvoří spektrum.

2.2.16 Definice. Formule jazyka $\mathcal{L}$, která je pravdivá v každém modelu $\mathfrak{M} \models \mathcal{L}$, je *logicky platná*. Formule, která je splněna alespoň v jednom modelu při alespoň jednom ohodnocení, je *splnitelná*. Formule, která není splněna v žádném modelu jazyka $\mathcal{L}$ při žádném ohodnocení, je *kontradikce*.

Kontradikce jsou právě nesplnitelné formule; negace logicky platné formule je kontradikce a naopak. Snadným zdrojem logicky platných formulí v libovolném jazyce $\mathcal{L}$ jsou výrokové tautologie: stačí v libovolné výrokové tautologii nahradit výrokové proměnné formulemi jazyka $\mathcal{L}$; například $(x < y) \to (x < y)$ je logicky platná formule jazyka uspořádání. Podobně jako u výrokových tautologií, ani zde nelze očekávat, že by logicky platné formule říkaly cokoli specifického: formule $(x < y) \to (x < y)$ je díky svému syntaktickému tvaru splněna v jakékoli realizaci jazyka, tedy bez ohledu na to, jaká relace na jaké množině realizuje symbol $<$ a jak jsou ohodnoceny proměnné x a y.

To může vzbudit dojem, že všechny logicky platné formule jsou podobně snadno rozpoznatelné. Opak je pravdou: dá se ukázat, že neexistuje žádný algoritmus, který by o každé předložené formuli jazyka predikátové logiky dokázal rozhodnout, zda je logicky platná (s výjimkou jazyka bez predikátů apod).[8]

2.2.17 Cvičení. Pro každou formuli $\varphi(x)$ s jednou volnou proměnnou jsou formule $\neg(\forall x)\varphi(x) \leftrightarrow (\exists x)\neg\varphi(x)$ a $\neg(\exists x)\varphi(x) \leftrightarrow (\forall x)\neg\varphi(x)$ logicky platné. Pro formuli $\psi(x, y)$ se dvěma volnými proměnnými jsou formule $\neg(\forall x)(\exists y)\psi(x, y) \leftrightarrow$ $(\exists x)(\forall y)\neg\psi(x, y)$ a $\neg(\exists x)(\forall y)\psi(x, y) \leftrightarrow (\forall x)(\exists y)\neg\psi(x, y)$ logicky platné. ∎

2.2.18 Cvičení. Uvažte následující formule v jazyce s jedním binárním predikátem R. Je alespoň jedna z nich logicky platná? Dokažte nebo dejte protipříklad. $(\exists y)(\forall x)R(x, y) \to (\forall x)(\exists y)R(x, y)$; $(\forall x)(\exists y)R(x, y) \to (\exists y)(\forall x)R(x, y)$.

2.2.19 Cvičení. Uvažte následující formule v jazyce se dvěma unárními predikáty P, Q. Rozhodněte, které z nich jsou logicky platné, splnitelné, a kontradiktorické. Pokud daná formule není kontradikcí, popište nějakou strukturu, ve které platí; pokud není logicky platná, popište strukturu, ve které neplatí.

$$(\forall x)(P(x) \wedge Q(x)) \leftrightarrow ((\forall x)P(x) \wedge (\forall x)Q(x))$$
$$(\exists x)(P(x) \wedge Q(x)) \leftrightarrow ((\exists x)P(x) \wedge (\exists x)Q(x))$$
$$(\forall x)(P(x) \vee Q(x)) \leftrightarrow ((\forall x)P(x) \vee (\forall x)Q(x))$$
$$(\exists x)(P(x) \vee Q(x)) \leftrightarrow ((\exists x)P(x) \vee (\exists x)Q(x))$$
$$(\forall x)(P(x) \to Q(x)) \leftrightarrow ((\forall x)P(x) \to (\forall x)Q(x))$$
$$(\exists x)(P(x) \to Q(x)) \leftrightarrow ((\exists x)P(x) \to (\exists x)Q(x))$$
$$(\forall x)(P(x) \leftrightarrow Q(x)) \leftrightarrow ((\forall x)P(x) \leftrightarrow (\forall x)Q(x))$$
$$(\exists x)(P(x) \leftrightarrow Q(x)) \leftrightarrow ((\exists x)P(x) \leftrightarrow (\exists x)Q(x))$$

[8] Ve výrokové logice, jak víme, dopadnou podobné úvahy triviálně: formule výrokové logiky neříkají nic, a o jejich logické pravdivosti lze algoritmicky rozhodovat.

Substituce termů za proměnné V matematice je běžné dosazovat za proměnné do termů a formulí, tak jako se dosazuje do rovnic v elementární algebře. Jsou-li $x_1, \ldots, x_n$ navzájem různé proměnné a $t, t_1, \ldots, t_n$ jsou libovolné termy jazyka $\mathcal{L}$, buď $t_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$ term, ve kterém je každý výskyt proměnné x_i nahrazen termem t_i.[9] Indukcí podle složitosti termu t se snadno dokáže, že $t_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$ je opět term jazyka $\mathcal{L}$. Podobně pro formuli φ jazyka $\mathcal{L}$, proměnné $x_1, \ldots, x_n$ a termy $[t_1, \ldots, t_n]$ buď $\varphi_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$ formule, která vznikne z formule φ současným nahrazením všech *volných* výskytů proměnné x_i termem t_i. Každá taková formule je pak *instancí* formule φ.

Smysl této substituce je následující: formule $\varphi_x[t]$ „říká“ o termu t totéž, co φ „říká“ o své volné proměnné x. Buď například φ formule $(\exists y)(x = y + y)$ jazyka aritmetiky s jednou volnou proměnnou x, která říká x ***je sudé číslo***. Je-li t term $p + q$, je $\varphi_x[t]$ formule $(\exists y)(p + q = y + y)$ s volnými proměnnými p, q, která říká $p + q$ ***je sudé číslo***. Je-li ovšem t term $y + 1$, je $\varphi_x[t]$ formule $(\exists y)(y + 1 = y + y)$, ve které je proměnná y vázaná. To nás vede k následující definici.

2.2.20 Definice. Buď x proměnná, t term a φ formule jazyka $\mathcal{L}$. Řekneme, že term t je ***substituovatelný*** za proměnnou x do formule φ, pokud žádná proměnná y termu t není vázaná v žádné podformuli, ve které je x volná.

Napíšeme-li v dalším $\varphi_x[t]$, předpokládáme, že term t je substituovatelný. Podle lemmatu 2.2.7 lze formuli φ vždy nahradit ekvivalentní formulí, ve které jsou případně přejmenovány „kolidující“ vázané proměnné.

2.2.21 Cvičení. Ukažte indukcí, že jsou-li $t, t_1, \ldots, t_n$ termy, pak výraz, který vznikne z termu t nahrazením výskytů navzájem různých proměnných $x_1, \ldots, x_n$ v termu t po řadě termy $t_1, \ldots, t_n$, je opět term (téhož jazyka). Podobně, je-li φ formule, pak výraz, který vznikne z formule φ nahrazením volných výskytů proměnných $x_1, \ldots, x_n$ ve formuli φ po řadě termy $t_1, \ldots, t_n$, je opět formule.

2.2.22 Cvičení. Buď $\mathfrak{M} \models \mathcal{L}$, buď φ formule jazyka $\mathcal{L}$, buďte $x_1, \ldots, x_n$ proměnné, a buď e ohodnocení proměnných, při kterém $t_i[e]$ je $m_i \in M$. Pak $t_{x_1,\ldots,x_n}[t_1, \ldots, t_n][e]$ je $t[e(x_1/m_1, x_n/m_n)]$, a $\mathfrak{M} \models \varphi_{x_1,\ldots,x_n}[t_1, \ldots, t_n][e]$ platí právě když $\mathfrak{M} \models \varphi[e(x_1/m_1, x_n/m_n)]$.

2.3 Dokazatelnost

Hilbertův systém Výstavbu formálního odovozovacího systému zahájíme jako v případě výrokové logiky redukcí jazyka. Z výrokových spojek popíšeme jen chování $\neg$ a $\rightarrow$, ostatní spojky chápeme jako příslušné zkratky. Z obou kvantifikátorů používáme jen univerzální $\forall$, a na výraz $(\exists x)\varphi$ hledíme jako na zkratku za $\neg(\forall x)\neg\varphi$. Smyslem této redukce je zjednodušit základní jazyk a snížit počet potřebných axiomů. Každou formuli daného jazyka $\mathcal{L}$ predikátové logiky lze ekvivalentně vyjádřit i v redukovaném jazyce.

Jako axiomy nyní přijmeme jisté formule popisující chování výrokových spojek a obecného kvantifikátoru. Předně, jsou-li A, B, C formule jazyka $\mathcal{L}$, pak každá z následujících formulí je axiomem predikátové logiky:

[9] Tato nahrazení se dějí „současně,“ aby nezáleželo na jejich pořadí a abychom předešli nejasnostem při kolizi ve jménech proměnných. Je-li například t aritmetický term $x * (y + z)$ a s, t, u jsou po řadě termy $(1 + 1)$, $(0 * z)$ a $(y + 1)$, je $t_{x,y,z}[s, t, u]$ term $(1 + 1) * ((0 * z) + (y + 1))$.

$$A \to (B \to A)$$

$$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$$

$$(\neg B \to \neg A) \to (A \to B)$$

Přijetím těchto axiomů se výroková logika stává částí predikátové logiky. Pokud za množinu $\mathcal{A}$ prvotních (výrokových) formulí vezmeme množinu všech atomických formulí jazyka $\mathcal{L}$ a všech formulí tvaru $(\forall x)\varphi$ a $(\exists x)\varphi$, kde x je proměnná a φ je formule jazyka $\mathcal{L}$, pak už každá formule jazyka $\mathcal{L}$ vzniká z $\mathcal{A}$ jen pomocí výrokových spojek. Přijmeme-li do predikátové logiky odvozovací pravidlo modus ponens (což za chvíli učiníme), bude každá výroková tautologie nad $\mathcal{A}$ dokazatelná v predikátové logice.

Syntaktické vlastnosti obecného kvantifikátoru popisují dvě schemata axiomů. Prvním z nich je *schema axiomů specifikace*: pro každou proměnnou x, každou formuli φ jazyka $\mathcal{L}$, a každý term t substituovatelný do φ za x je formule

$$(\forall x)\varphi \to \varphi_x[t]$$

axiomem predikátové logiky. Smysl těchto axiomů je dosti přirozený: pokud formule φ platí „pro každé“ x, pak platí i v každém speciálním případě.

Druhé schema je spíše technické; využijeme ho při důkazu věty o dedukci a při hledání prenexního tvaru formulí. Pro každé dvě formule φ, ψ jazyka $\mathcal{L}$ a každou proměnnou x, která nemá volný výskyt ve formuli φ, je následující formule axiomem:

$$(\forall x)(\varphi \to \psi) \to (\varphi \to (\forall x)\psi)$$

Odvozovacími pravidly predikátové logiky jsou *modus ponens*, které známe již z výrokové logiky, a se kterým do predikátové logiky spolu s výrokovými axiomy přechází výroková dokazatelnost, a *pravidlo generalizace*:

Pro libovolnou proměnnou x, z formule φ odvoď formuli $(\forall x)\varphi$.

Je-li tedy dokazatelná nějaká formule φ, která má případně volnou proměnnou x, pak je dokazatelná i $(\forall x)\varphi$. Takové je postavení volných proměnných.

Později přijmeme ještě axiomy popisující binární predikát $=$ pro rovnost. Tím rozšíříme predikátovou logiku v jazyce $\mathcal{L}$ do *logiky s rovností*.

Zavedením axiomů a odvozovacích pravidel predikátové logiky je dán nový význam symbolu $\vdash$ pro dokazatelnost. Sám pojem důkazu se zavede analogicky jako ve výrokové logice, ale $\vdash$ nyní znamená dokazatelnost z právě zavedených axiomů pomocí obou odvozovacích pravidel.

2.3.1 Definice. Buď $\mathcal{L}$ jazyk predikátové logiky. Řekneme, že konečná posloupnost $\varphi_1, \ldots, \varphi_n$ formulí jazyka $\mathcal{L}$ je *důkazem* formule φ v predikátové logice, pokud φ_n je formule φ a každá φ_i je buďto axiomem predikátové logiky, nebo je z nějakých předchozích formulí odvozena pomocí některého z odvozovacích pravidel. Pokud nějaký takový důkaz existuje, řekneme, že formule φ je *dokazatelná* v predikátové logice, a píšeme $\vdash \varphi$.

Ukážeme nejprve, že s každou dokazatelnou formulí je dokazatelná i každá její instance. Narozdíl od výrokové logiky musíme ošetřit výskyty proměnných.

2.3.2 Lemma (o instancích). *Buď φ dokazatelná formule a buď ψ její instance, tj. formule tvaru $\varphi_{x_1,\dots,x_n}[t_1,\dots,t_n]$, pro nějaké termy $t_1,\dots,t_n$. Potom i ψ je dokazatelná. Jinými slovy, každá instance dokazatelné formule je dokazatelná.*

Důkaz. Dokazujeme indukcí podle počtu dosazených termů. Je-li ψ tvaru $\varphi_x[t]$, pak z $\vdash \varphi$ máme $\vdash (\forall x)\varphi$ pomocí pravidla generalizace, a $\vdash (\forall x)\varphi \to \varphi_x[t]$ je axiom specifikace, takže pomocí modus ponens dostáváme $\vdash \varphi_x[t]$. Pokud je dosazených termů více, musíme ošetřit případ, kdy některý z termů obsahuje proměnnou, za kterou dosazujeme jiný term.[10] Buďte $y_1,\dots,y_n$ proměnné, které se nevyskytují ve formuli φ ani v termech $t_1,\dots,t_n$. Potom z předpokladu $\vdash \varphi$ dostáváme $\vdash \varphi_{x_1}[y_1]$ jako výše, a stejně můžeme postupovat dále: substitucí y_2 za x_2 dostaneme opět dokazatelnou formuli $\varphi_{x_1,x_2}[y_1,y_2]$, až nakonec máme $\vdash \varphi_{x_1,\dots,x_n}[y_1,\dots,y_n]$. V této formuli nemají proměnné $x_1,\dots,x_n$ volný výskyt, na místech původního volného výskytu x_i se nyní volně vyskytuje y_i. Přitom term t_i je substituovatelný za y_i. Označíme-li poslední formuli jako ϑ, pak z dokazatelnosti ϑ máme také $\vdash \vartheta_{y_1}[t_1]$ jako na začátku. Přitom v termu t_1 se nevyskytuje žádná z proměnných y_i, a substitucí t_2 za y_2 do formule $\vartheta_{y_1}[t_1]$ vznikne dokazatelná formule $\vartheta_{y_1,y_2}[t_1,t_2]$. Takto postupně získáme dokazatelnou formuli $\vartheta_{y_1,\dots,y_n}[t_1,\dots,t_n]$, což je právě instance ψ formule φ. □

Zavedli jsme pojem dokazatelnosti v predikátové logice. Dokážeme nejprve některé jednoduché věty o kvantifikátorech, které budeme v dalším používat jako technické obraty. Využíváme při tom volně výsledky z výrokové logiky.

2.3.3 Lemma. *Pro formuli φ, proměnnou x a term t je $\vdash \varphi_x[t] \to (\exists x)\varphi$.*

Důkaz. Jedná se o duální podobu axiomu specifikace. Totiž $(\forall x)\neg\varphi \to \neg\varphi_x[t]$ je axiom, a podle 1.5.17 je $\vdash \neg\neg(\forall x)\neg\varphi \to (\forall x)\neg\varphi$. Přitom $(\exists x)\varphi$ je zkratka za $\neg(\forall x)\neg\varphi$, takže složením obou implikací máme $\vdash \neg(\exists x)\varphi \to \neg\varphi_x[t]$, a pomocí H3 a modus ponens nakonec také $\vdash \varphi_x[t] \to (\exists x)\varphi$. □

2.3.4 Lemma. *Buď $\vdash \varphi \to \psi$. Potom pro každou proměnnou x je také*

(i) $\vdash \varphi \to (\forall x)\psi$, *pokud x není volná ve formuli φ.*

(ii) $\vdash (\exists x)\varphi \to \psi$, *pokud x není volná ve formuli ψ.*

Důkaz. (i) Je-li $\vdash \varphi \to \psi$, je pomocí pravidla generalizace také $\vdash (\forall x)(\varphi \to \psi)$. Pokud x není volná ve formuli φ, je $(\forall x)(\varphi \to \psi) \to (\varphi \to (\forall x)\psi)$ axiom, takže aplikací modus ponens máme $\vdash \varphi \to (\forall x)\psi$. (ii) Je-li $\vdash \varphi \to \psi$, je také $\vdash \neg\psi \to \neg\varphi$, a pokud x není volná v ψ, máme pomocí (i) také $\vdash \neg\psi \to (\forall x)\neg\varphi$. Potom je ale $\vdash \neg(\forall x)\neg\varphi \to \neg\neg\psi$, tedy také $\vdash (\exists x)\varphi \to \psi$. □

2.3.5 Lemma (o distribuci). *Je-li $\vdash \varphi \to \psi$, je pro každou proměnnou x také*

(i) $\vdash (\forall x)\varphi \to (\forall x)\psi$

(ii) $\vdash (\exists x)\varphi \to (\exists x)\psi$

[10] Pokud například do formule φ tvaru $x_1 = x_2$ dosazujeme za x_1 term t_1 tvaru x_2 a za x_2 term t_2 tvaru x_1, je $\varphi_{x_1,x_2}[t_1,t_2]$ podle definice formule $x_2 = x_1$. Dosazujeme-li ovšem postupně, je $\varphi_{x_1}[t_1]$ formule $x_2 = x_2$ a $(\varphi_{x_1}[t_1])_{x_2}[t_2]$ je formule $x_1 = x_1$, zatímco v opačném pořadí bude $\varphi_{x_2}[t_2]$ formule $x_1 = x_1$ a $(\varphi_{x_2}[t_2])_{x_1}[t_1]$ formule $x_2 = x_2$. Proto nemůžeme přímočaře využít výsledek pro n dosazených termů při indukčním kroku pro $n+1$ termů.

Důkaz. (i) Formule $(\forall x)\varphi \to \varphi$ je instancí axiomu specifikace, takže při $\vdash \varphi \to \psi$ máme složením implikací také $\vdash (\forall x)\varphi \to \psi$. Přitom x není volná ve formuli $(\forall x)\varphi$, takže podle 2.3.4 je také $\vdash (\forall x)\varphi \to (\forall x)\psi$. (ii) Formule $\psi \to (\exists x)\psi$ je instancí 2.3.3, takže při $\vdash \varphi \to \psi$ máme složením implikací $\vdash \varphi \to (\exists x)\psi$. Přitom x není volná ve formuli $(\exists x)\psi$, takže podle 2.3.4 je také $\vdash (\exists x)\varphi \to (\exists x)\psi$. □

2.3.6 Lemma (o ekvivalenci). *Buď φ formule, a buď φ' formule, která vznikne z φ nahrazením všech výskytů podformulí $\varphi_1, \dots, \varphi_n$ podformulemi $\varphi'_1, \dots, \varphi'_n$. Potom je-li $\vdash \varphi_i \leftrightarrow \varphi'_i$ pro každé $i \leq n$, je také $\vdash \varphi \leftrightarrow \varphi'$.*

Důkaz. Dokazujeme indukcí podle složitosti formule φ. Pro φ atomickou není co dokazovat. Indukční kroky pro výrokové spojky $\neg$ a $\to$ známe z výrokové logiky. Zbývá případ, kdy φ je tvaru $(\forall x)\psi$, přičemž pro ψ je $\vdash \psi \leftrightarrow \psi'$ již dokázáno. Je tedy $\vdash \psi \to \psi'$ a $\vdash \psi' \to \psi$, a použitím 2.3.5 dostáváme $\vdash (\forall x)\psi \to (\forall x)\psi'$ a $\vdash (\forall x)\psi' \to (\forall x)\psi$, neboli $\vdash \varphi \to \varphi'$ a $\vdash \varphi' \to \varphi$. □

2.3.7 Definice. Je-li φ formule, potom její *varianta* vznikne postupným nahrazením podformulí tvaru $(Qx)\vartheta$ podformulemi tvaru $(Qy)\vartheta_x[y]$, kde Q je kvantifikátor a y je nějaká proměnná, která se nevyskytuje nikde jinde ve φ.

Jinými slovy, varianta formule vznikne přejmenováním vázaných proměnných. Nepřekvapí nás, že takové formule jsou dokazatelně ekvivalentní:

2.3.8 Lemma (o variantách). *Je-li ψ varianta formule φ, je $\vdash \varphi \leftrightarrow \psi$.*

Důkaz. Stačí ukázat, že nahrazované podformule jsou ekvivalentní. Ukážeme případ pro kvantifikátor $\forall$, případ pro $\exists$ je analogický. Je-li $(\forall x)\vartheta$ nahrazovaná podformule, máme $\vdash (\forall x)\vartheta \to \vartheta_x[y]$ podle axiomu specifikace. Přitom y se nevyskytuje ve formuli $(\forall x)\vartheta$, takže $\vdash (\forall x)\vartheta \to (\forall y)\vartheta_x[y]$ podle 2.3.4. Opačná implikace $\vdash (\forall y)\vartheta_x[y] \to (\forall x)\vartheta$ se dokáže stejně. Tedy $\vdash (\forall x)\vartheta \leftrightarrow (\forall y)\vartheta_x[y]$. □

2.3.9 Lemma. *Pro formuli φ, proměnné $x_1, \dots, x_n$ a termy $t_1, \dots, t_n$ je*

(i) $\vdash (\forall x_1) \dots (\forall x_n)\varphi \to \varphi_{x_1,\dots,x_n}[t_1, \dots, t_n]$

(ii) $\vdash \varphi_{x_1,\dots,x_n}[t_1, \dots, t_n] \to (\exists x_1) \dots (\exists x_n)\varphi$

Důkaz. Pro formuli φ je $\vdash (\forall x_n)\varphi \to \varphi$ pomocí axiomu specifikace, jelikož $\varphi_x[x]$ je formule φ. Tak postupně získáme $\vdash (\forall x_{n-1})(\forall x_n)\varphi \to (\forall x_n)\varphi$ až po formuli $\vdash (\forall x_1) \dots (\forall x_n)\varphi \to (\forall x_2) \dots (\forall x_n)\varphi$. Postupnou aplikací modus ponens pak dostaneme $\vdash (\forall x_1) \dots (\forall x_n)\varphi \to \varphi$. Přitom (i) je instancí této formule. Tvrzení (ii) se dokáže podobně. Je $\vdash (\forall x_n)\neg\varphi \to \neg\varphi$, tedy také $\vdash \neg\neg(\forall x_n)\neg\varphi \to \neg\varphi$, neboli $\vdash \neg(\exists x_n)\varphi \to \neg\varphi$, a standardním výrokovým obratem pak $\vdash \varphi \to (\exists x_n)\varphi$. Podobně jako výše získáme $\vdash \varphi \to (\exists x_1) \dots (\exists x_n)\varphi$, a (ii) je její instancí. □

S pomocí předchozích lemmat lze snadno ukázat, že pro každou permutaci π na množině indexů $1, \dots, n$ je $\vdash (\forall x_1) \dots (\forall x_n)\varphi \leftrightarrow (\forall x_{\pi(1)}) \dots (\forall x_{\pi(n)})\varphi$ a $\vdash (\exists x_1) \dots (\exists x_n)\varphi \leftrightarrow (\exists x_{\pi(1)}) \dots (\exists x_{\pi(n)})\varphi$.

2.3.10 Definice. Buď φ formule, jejíž volné proměnné jsou právě $x_1, \dots, x_n$. Potom $(\forall x_1) \dots (\forall x_n)\varphi$ je *uzávěr* formule φ, který budeme dále značit $\overline{\varphi}$.

Ve značení $\overline{\varphi}$ sice zaniká pořadí, ve kterém jsou volné proměnné formule φ kvantifikovány, ale podle předchozího pozorování na tomto pořadí nezáleží.

2.3.11 Věta (o uzávěru). *Pro každou formuli φ je $\vdash \varphi$ právě když $\vdash \overline{\varphi}$.*

Důkaz. Je-li $\vdash \varphi$, získáme $\vdash \overline{\varphi}$ opakovanou aplikací pravidla generalizace. Je-li naopak $\vdash \overline{\varphi}$, získáme $\vdash \varphi$ aplikací 2.3.9 a modus ponens. □

Věta o uzávěru popisuje syntaktickou roli volných proměnných: dokážeme-li formuli s volnými proměnnými, je tím dokázán i její uzávěr, ve kterém jsou všechny volné proměnné vázány obecným kvantifikátorem. Dokážeme-li například $x > 0$, je tím dokázáno i $(\forall x)(x > 0)$. Podobně například teorie grup může požadavek na komutativitu zachytit stejně dobře axiomem $x * y = y * x$ jako axiomem $(\forall x)(\forall y)(x * y = y * x)$.

2.3.12 Věta (o konstantách). *Buď φ formule s volnými proměnnými $x_1, \ldots, x_k$, buďte $c_1, \ldots, c_k$ nové konstanty. Potom je $\vdash \varphi$ právě když $\vdash \varphi_{x_1,\ldots,x_k}[c_1, \ldots, c_k]$.*

Důkaz. Jeden směr plyne ihned z věty o instancích. V opačném směru označme formuli $\vdash \varphi_{x_1,\ldots,x_k}[c_1, \ldots, c_k]$ jako φ' a předpokládejme, že $\vartheta_1, \ldots, \vartheta_n$ je důkaz formule φ'. Buďte $y_1, \ldots, y_k$ proměnné, které se nevyskytují ve formuli φ ani v uvažovaném důkaze. Označme jako φ_i formuli, která vznikne z formule ϑ_i nahrazením všech výskytů všech konstant c_j po řadě proměnnými y_j. Potom $\varphi_1, \ldots, \varphi_n$ je důkaz formule $\varphi_{x_1,\ldots,x_k}[y_1, \ldots, y_k]$: je-li ϑ_i axiom predikátové logiky, je φ_i axiom stejného typu, a je-li ϑ_i odvozena z předchozích formulí nějakým odvozovacím pravidlem, je také φ_i odvozena z odpovídajících formulí stejným pravidlem. Je tedy $\vdash \varphi_{x_1,\ldots,x_k}[y_1, \ldots, y_k]$, a formule φ je její instance. □

2.3.13 Věta (o dedukci). *Buď T množina formulí jazyka $\mathcal{L}$, buďte φ, ψ formule jazyka $\mathcal{L}$, přičemž φ je uzavřená. Potom $T \vdash \varphi \to \psi$ právě když $T, \varphi \vdash \psi$.*

Důkaz. Věta se dokáže indukcí podle složitosti důkazu jako ve výrokové logice. Ve směru zprava doleva je navíc potřeba rozlišit jen případ, kdy nějaká formule ϑ_i z důkazu $\vartheta_1, \ldots, \vartheta_n$ formule ψ v teorii T, φ je odvozena pomocí pravidla generalizace z nějaké předchozí formule ϑ_j. V takovém případě je ϑ_i formule tvaru $(\forall x)\vartheta_j$. Přitom proměnná x není volná ve formuli φ, jelikož φ je uzavřená, takže $(\forall x)(\varphi \to \vartheta_j) \to (\varphi \to (\forall x)\vartheta_j)$ je axiom. Z indukčního předpokladu již máme $T \vdash \varphi \to \vartheta_j$, pomocí pravidla generalizace tedy také $T \vdash (\forall x)(\varphi \to \vartheta_j)$, a pomocí modus ponens dostáváme $T \vdash \varphi \to (\forall x)\vartheta_j$, neboli $T \vdash \varphi \to \vartheta_i$. □

Ve skutečnosti potřebujeme ve větě o dedukci zaručit jen to, že žádná proměnná x, na kterou se v důkazu formule ψ z množiny T, φ použilo pravidlo generalizace, není volná ve formuli φ, abychom mohli použít patřičný axiom. Není nutné, aby φ byla uzavřená, nicméně předpoklad o volných proměnných je ve větě o dedukci podstatný: například z formule $x = 0$ je dokazatelná formule $(\forall x)(x = 0)$, ale implikace $(x = 0) \to (\forall x)(x = 0)$ dokazatelná není.

2.3.14 Důsledek. *$T \vdash \varphi$ právě tehdy, když $T \cup \{\neg\overline{\varphi}\}$ je sporná.*

Důkaz. Pokud T dokazuje formuli φ, pak podle věty o uzávěru dokazuje i $\overline{\varphi}$, takže $T, \neg\overline{\varphi}$ je sporná. Je-li naopak $T \cup \{\neg\overline{\varphi}\}$ sporná, pak dokazuje jakoukoli formuli, tedy i formuli $\overline{\varphi}$. Podle věty o dedukci je potom $T \vdash \neg\overline{\varphi} \to \overline{\varphi}$, a pomocí 1.5.19 získáme $T \vdash \overline{\varphi}$ jako v případě výrokové logiky, tedy také $T \vdash \varphi$. □

Logika s rovností Binární predikát $=$ pro rovnost má v jazyce výsadní postavení. Většinou jej považujeme za součást každého jazyka, a v sémantice požadujeme, aby byl vždy realizován tak jak očekáváme, totiž relací identity. Jeho syntaktické vlastnosti nyní popíšeme ve třech schematech axiomů, která zachycují přirozené představy o rovnosti: každé individuum je rovno samo sobě, navzájem si rovná individua splňují tytéž relace a chovají se stejně ve funkcích.

(E1) Pro každou proměnnou x je formule $x = x$ axiomem.

(E2) Pro proměnné $x_1, \ldots, x_n, y_1, \ldots, y_n$ a n-ární predikátový symbol R:
$x_1 = y_1 \to (x_2 = y_2 \to \ldots \to (x_n = y_n \to R(x_1, \ldots, x_n) \to R(y_1, \ldots, y_n)) \ldots)$

(E3) Pro proměnné $x_1, \ldots, x_n, y_1, \ldots, y_n$ a n-ární funkční symbol f:
$x_1 = y_1 \to (x_2 = y_2 \to \ldots \to (x_n = y_n \to f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)) \ldots)$

2.3.15 Lemma. *Pro každé proměnné x, y, z je*

(i) $\vdash x = y \to y = x$

(ii) $\vdash x = y \to (y = z \to x = z)$

Důkaz. (i) Formule[11] $x = y \to x = x \to x = x \to y = x$ je případem axiomu (E2). Obvyklou záměnou předpokladů v implikaci je tedy dokazatelná i formule $x = x \to x = x \to x = y \to y = x$. Podle (E1) tedy máme $\vdash x = y \to y = x$ dvojím užitím modus ponens.

(ii) Formule $y = x \to z = z \to y = z \to x = z$ je případem axiomu (E2); odtud opět $\vdash z = z \to y = x \to y = z \to x = z$. Užitím modus ponens a (E1) je $\vdash y = x \to y = z \to x = z$, a podle (i) tedy také $\vdash x = y \to y = z \to x = z$. □

Věty o rovnosti se indukcí snadno rozšíří z rovnosti proměnných na rovnosti termů, a ze substituce jednoho termu na konečně mnoho.

2.3.16 Lemma. *Buďte $s_1, \ldots, s_n, t_1, \ldots, t_n$ termy, pro které $\vdash s_i = t_i$. Pak*

(i) *Je-li s term, a term t vznikne z termu s nahrazením výskytů termů s_i odpovídajícími termy t_i, pak $\vdash s = t$.*

(ii) *Je-li φ formule, a formule ψ vznikne z φ nahrazením výskytů termů s_i v atomických podformulích odpovídajícími termy t_i, pak $\vdash \varphi \leftrightarrow \psi$.*

2.3.17 Lemma. *Jsou-li $s_1, \ldots, s_n, t_1, \ldots, t_n, t$ termy, x je proměnná, která se nevyskytuje v termu t, a φ je libovolná formule, pak*

(i) $\vdash s_1 = t_1 \to s_2 = t_2 \to \ldots \to s_n = t_n \to t[s_1, \ldots, s_n] = t[t_1, \ldots, t_n]$

(ii) $\vdash s_1 = t_1 \to s_2 = t_2 \to \ldots \to s_n = t_n \to \varphi[s_1, \ldots, s_n] \leftrightarrow \varphi[t_1, \ldots, t_n]$

Prenexní tvar Ve výrokové logice jsme dokázali větu o normálním tvaru: každou výrokovou formuli lze z prvotních formulí sestavit jen s pomocí jistých spojek, navíc ve zvoleném pořadí. Prenexní tvar je analogií normálního tvaru pro formule jazyka predikátové logiky: požadujeme, aby se kvantifikátory použily při výstavbě formule až nakonec. Později ukážeme, jak lze takový tvar využít v teorii modelů a při strojovém dokazování vět.

[11] Vynechané závorky se kumulují doprava.

2.3.18 Definice. Formule $(Q_1x_1)\dots(Q_nx_n)\varphi$ je v *prenexním normálním tvaru*, pokud Q_i jsou kvantifikátory, x_i jsou proměnné, a φ je otevřená podformule. Kvantifikace $(Q_1x_1)\dots(Q_nx_n)$ pak tvoří její *prefix* a otevřená podformule φ její *otevřené jádro*. Formule v prenexním tvaru je *univerzální* (*existenční*), pokud všechny kvantifikátory v jejím prefixu jsou univerzální (existenční).

2.3.19 Věta. *Ke každé formuli φ existuje prenexní formule φ' tak, že $\vdash \varphi \leftrightarrow \varphi'$.*

Prenexní tvar formule se získá pomocí *prenexních operací*, které postupně přesouvají kvantifikátory z podformulí do prefixu. Normální tvar otevřeného jádra pak získáme stejně jako ve výrokové logice. Při popisu prenexních operací použijeme následující značení: pokud symbol Q zastupuje jeden z kvantifikátorů, pak $\overline{Q}$ zastupuje opačný kvantifikátor.

(i) Nahraď podformuli nějakou její variantou.

(ii) Nahraď podformuli $(Qx)\neg\psi$ formulí $\neg(\overline{Q}x)\psi$.

(iii) Nahraď podformuli $\psi \to (Qx)\vartheta$ formulí $(Qx)(\psi \to \vartheta)$, pokud x není volná v ψ.

(iv) Nahraď podformuli $(Qx)\psi \to \vartheta$ formulí $(\overline{Q}x)(\psi \to \vartheta)$, pokud x není volná v ϑ.

(v) Nahraď podformuli $(Qx)\psi \wedge \vartheta$ formulí $(Qx)(\psi \wedge \vartheta)$, pokud x není volná v ϑ.

(vi) Nahraď podformuli $(Qx)\psi \vee \vartheta$ formulí $(Qx)(\psi \vee \vartheta)$, pokud x není volná v ϑ.

Například formule $(\forall x)(\forall y)(x < y \to (\exists z)(x < z \wedge z < y))$ jazyka uspořádání není v prenexním tvaru, ale s použitím (iii) ji lze přepsat na formuli $(\forall x)(\forall y)(\exists z)(x < y \to (x < z \wedge z < y))$. Podobně pro aritmetickou formuli[12]

(i) $(\forall x)(\forall y)[((\forall k)(x \neq k+k) \wedge \underline{(\forall k)}(y \neq k+k)) \to (\forall k)(x * y \neq k+k)]$

(i) $(\forall x)(\forall y)[((\forall k)(x \neq k+k) \wedge (\forall l)(y \neq l+l)) \to \underline{(\forall k)}(x * y \neq k+k)]$

(v) $(\forall x)(\forall y)[(\underline{(\forall k)}(x \neq k+k) \wedge (\forall l)(y \neq l+l)) \to (\forall m)(x * y \neq m+m)]$

(v) $(\forall x)(\forall y)[(\forall k)((x \neq k+k) \wedge \underline{(\forall l)}(y \neq l+l)) \to (\forall m)(x * y \neq m+m)]$

(iii) $(\forall x)(\forall y)[(\forall k)(\forall l)((x \neq k+k) \wedge (y \neq l+l)) \to \underline{(\forall m)}(x * y \neq m+m)]$

(iv) $(\forall x)(\forall y)(\forall m)[\underline{(\forall k)}(\forall l)((x \neq k+k) \wedge (y \neq l+l)) \to (x * y \neq m+m)]$

(iv) $(\forall x)(\forall y)(\forall m)(\exists k)[\underline{(\forall l)}((x \neq k+k) \wedge (y \neq l+l)) \to (x * y \neq m+m)]$

$(\forall x)(\forall y)(\forall m)(\exists k)(\exists l)[((x \neq k+k) \wedge (y \neq l+l)) \to (x * y \neq m+m)]$

Rekurzivním prováděním prenexních operací lze od každé formule dospět k prenexnímu tvaru. Zbývá ukázat, že nahrazované podformule jsou ekvivalentní.

2.3.20 Lemma. *Pro každou proměnnou x a každé formule ψ, ϑ jsou podformule nahrazované v prenexních operacích dokazatelně ekvivalentní.*

Důkaz. V případě (i) se jedná o větu o variantách.
Případ (ii) pro $\forall$: Máme $\vdash \neg(\forall x)\psi \leftrightarrow \neg(\forall x)\neg\neg\psi$, neboli $\vdash \neg(\forall x)\psi \leftrightarrow (\exists x)\neg\psi$.
Případ (ii) pro $\exists$: Máme $\vdash \neg(\exists x)\psi \leftrightarrow \neg(\exists x)\neg\neg\psi$, neboli $\vdash \neg(\exists x)\psi \leftrightarrow (\forall x)\neg\psi$.
Případ (iii) pro $\forall$: Pokud x není volná v ψ, je $(\forall x)(\psi \to \vartheta) \to (\psi \to (\forall x)\vartheta)$ axiom. V opačném směru máme $\vdash ((\forall x)\vartheta \to \vartheta) \to ((\psi \to (\forall x)\vartheta) \to (\psi \to \vartheta))$, což je výroková tautologie, složení implikací $\psi \to (\forall x)\vartheta$ a $(\forall x)\vartheta \to \vartheta$. Přitom $(\forall x)\vartheta \to \vartheta$ je axiom specifikace, takže $\vdash (\psi \to (\forall x)\vartheta) \to (\psi \to \vartheta)$. Podle 2.3.4 je tedy také $\vdash (\psi \to (\forall x)\vartheta) \to (\forall x)(\psi \to \vartheta)$, neboť x není volná v $\psi \to (\forall x)\vartheta$.

[12] Podtržením vyznačujeme, kterou kvantifikovanou podformulí se momentálně zabýváme.

Případ (iii) pro $\exists$: Formule $(\vartheta \to (\exists x)\vartheta) \to ((\psi \to \vartheta) \to (\psi \to (\exists x)\vartheta))$ je výroková tautologie (složení implikací), přitom podle 2.3.3 je $\vdash \vartheta \to (\exists x)\vartheta$. Máme tedy $\vdash (\psi \to \vartheta) \to (\psi \to (\exists x)\vartheta)$, takže i $\vdash (\exists x)(\psi \to \vartheta) \to (\psi \to (\exists x)\vartheta)$ podle 2.3.4, neboť x není volná v $\psi \to (\exists x)\vartheta$. V opačném směru využijeme výrokovou tautologii $(\neg A \to C) \to ((B \to C) \to ((A \to B) \to C))$. Formule $\neg\psi \to (\psi \to \vartheta)$ je tautologie, a podle 2.3.3 je $\vdash (\psi \to \vartheta) \to (\exists x)(\psi \to \vartheta)$, takže $\vdash \neg\psi \to (\exists x)(\psi \to \vartheta)$. Formule $\vartheta \to (\psi \to \vartheta)$ je axiom, a podle 2.3.5 je také $\vdash (\exists x)\vartheta \to (\exists x)(\psi \to \vartheta)$. Označíme-li formuli ψ jako A, formuli $(\exists x)\vartheta$ jako B, a formuli $(\exists x)(\psi \to \vartheta)$ jako C, dokázali jsme $\neg A \to C$ a $B \to C$, takže podle tautologie výše máme $\vdash (A \to B) \to C$, neboli $\vdash (\psi \to (\exists x)\vartheta) \to (\exists x)(\psi \to \vartheta)$. Případ (iv) pro $\forall$: Formule $((\forall x)\psi \to \vartheta) \leftrightarrow (\neg\vartheta \to \neg(\forall x)\psi)$ je tautologie, takže $\vdash ((\forall x)\psi \to \vartheta) \leftrightarrow (\neg\vartheta \to \neg(\forall x)\neg\neg\psi)$, tj. $\vdash ((\forall x)\psi \to \vartheta) \leftrightarrow (\neg\vartheta \to (\exists x)\neg\psi)$. Přitom podle (iii) pro $\exists$ je $\vdash (\neg\vartheta \to (\exists x)\neg\psi) \leftrightarrow (\exists x)(\neg\vartheta \to \neg\psi)$, neboť x není volná v $\neg\vartheta$. Je tedy $\vdash ((\forall x)\psi \to \vartheta) \leftrightarrow (\exists x)(\neg\vartheta \to \neg\psi)$, takže nakonec máme $\vdash ((\forall x)\psi \to \vartheta) \leftrightarrow (\exists x)(\psi \to \vartheta)$. Případ (iv) pro $\exists$ se dokáže podobně pomocí (iii) pro $\forall$. V případě (v) a (vi) můžeme formuli nejprve přeložit do jazyka výrokových spojek $\neg, \to$ a odkázat se na předchozí případy. □

Důkaz věty. Je-li daná formule φ atomická, není co dokazovat. Je-li φ tvaru $\neg\psi$ a známe-li prenexní tvar ψ' formule ψ, získáme prenexní tvar φ' formule φ pomocí operace (ii) pro negaci. Je-li φ tvaru $\psi \to \vartheta$ a známe-li prenexní tvary ψ', ϑ', zvolme nejprve varianty ψ'', ϑ'' tak, aby žádná volná proměnná v ψ'' nebyla vázaná v ϑ'' a naopak. Prenexní tvar formule $\psi'' \to \vartheta''$ pak získáme operacemi (iii) a (iv) pro implikaci. Je-li φ tvaru $(\forall x)\psi$ a známe-li prenexní tvar ψ' formule ψ, je formule $(\forall x)\psi'$ prenexním tvarem formule φ, pokud x není vázaná v ψ; jinak je sama ψ' prenexním tvarem formule φ. Je-li φ tvaru $\psi \wedge \vartheta$ nebo $\psi \vee \vartheta$, můžeme tyto spojky nejprve přeložit do jazyka $\neg, \to$, nebo použít operace (v) a (vi) pro konjunkci a disjunkci. Formuli φ tvaru $\psi \leftrightarrow \vartheta$ můžeme ekvivalentně nahradit konjunkcí obou implikací a použít (iii)–(v). □

Sadu prenexních operací zřejmě můžeme rozšířit o některé další technické obraty, například: podformuli tvaru $(Qx)\psi$ nahraď podformulí ψ, pokud proměnná x není volná v ψ. V definici prenexu pak můžeme navíc požadovat, aby všechny kvantifikované proměnné v prefixu byly navzájem různé.

2.3.21 Cvičení. Ukažte na příkladech, že zákazy volných výskytů proměnných v prenexním axiomu a v prenexních operacích jsou nutné, tj. že: (i) pokud se proměnná x volně vyskytuje v φ, pak formule $(\forall x)(\varphi \to \psi) \to (\varphi \to (\forall x)\psi)$ není logicky platná; (ii) pokud se proměnná x volně vyskytuje v ψ, pak ekvivalence $((Qx)\varphi \to \psi) \leftrightarrow (\overline{Q}x)(\varphi \to \psi)$ nejsou logicky platné. (iii) pokud se proměnná x volně vyskytuje v φ, potom ekvivalence $(\varphi \to (Qx)\psi) \leftrightarrow (Qx)(\varphi \to \psi)$, $(\varphi \wedge (Qx)\psi) \leftrightarrow (Qx)(\varphi \wedge \psi)$ a $(\varphi \vee (Qx)\psi) \leftrightarrow (Qx)(\varphi \vee \psi)$ nejsou logicky platné.

2.4 Úplnost

V předchozích oddílech jsme popsali sémantiku a syntax predikátové logiky, totiž splňování ve strukturách a formální odvozovací systém. Nyní ukážeme, že zavedená syntax a sémantika si vzájemně odpovídají: formule dokazatelné v Hilbertově systému jsou právě logicky platné formule.

Místo dokazatelnosti v logice a platnosti v realizacích jazyka budeme rovnou obecněji zkoumat dokazatelnost v nějaké teorii a platnost v jejích modelech. Nejprve zavedeme potřebné pojmy.

2.4.1 Definice. Buď $\mathcal{L}$ jazyk predikátové logiky. Potom *teorie* v jazyce $\mathcal{L}$ je jakákoli množina formulí jazyka $\mathcal{L}$; tyto formule jsou pak její *axiomy*. Řekneme, že realizace $\mathfrak{M}$ jazyka $\mathcal{L}$ je *modelem* teorie T, a píšeme $\mathfrak{M} \models T$, pokud každý axiom z T je splněn v $\mathfrak{M}$. Pokud formule φ jazyka $\mathcal{L}$ platí v každém modelu $\mathfrak{M} \models T$, řekneme, že φ je *logickým důsledkem* teorie T a píšeme $T \models \varphi$.

Speciálně modelem prázdné teorie je každá realizace jazyka $\mathcal{L}$; pokud formule φ platí v každé realizaci jazyka $\mathcal{L}$, řekneme, že je *logicky platná* a píšeme $\models \varphi$.

Zformulovat nějakou teorii, tedy množinu formulí, nejlépe konečnou, je standardní způsob, jak vymezit struktury, kterými se chceme zabývat. Ve vhodném, k tomu účelu vytvořeném jazyce zformulujeme axiomy, které považujeme z nějakého důvodu za přirozené nebo zajímavé, a zkoumáme struktury, ve kterých tyto axiomy platí, tj. modely takové teorie.

2.4.2 Příklad. Teorie (ostrého) uspořádání v jazyce s jedním binárním predikátem $<$ obsahuje axiomy $(\forall x)(\forall y)(\forall z)(x < y \wedge y < z \to x < z)$ a $(\forall x)\neg(x < x)$. Přidáním axiomů $(\forall x)(\forall y)(x < y \vee x = y \vee y < x)$, $(\forall x)(\exists y)(\exists z)(x < y \wedge z < x)$ a $(\forall x)(\forall y)(\exists z)(x < y \to x < z \wedge z < y)$ pak vznikne teorie lineárních, neomezených, hustých uspořádání.

2.4.3 Příklad. Teorie grup v jazyce s binárním funkčním symbolem $+$, unárním funkčním symbolem $-$ a konstantou 0 obsahuje jako axiomy následující sentence: $(\forall x)(\forall y)(\forall z)((x + y) + z = x + (y + z))$, $(\forall x)(x + 0 = x)$, $(\forall x)(x + (-x) = 0)$. Rozšířením o axiom $(\forall x)(\forall y)(x + y = y + x)$ vznikne teorie komutativních grup.

2.4.4 Příklad. Teorie (komutativních) okruhů v jazyce s binárními funkčními symboly $+$ a $*$, unárním funkčním symbolem $-$ a konstantami 0 a 1 rozšiřuje teorii (komutativních) grup o axiomy $(\forall x)(\forall y)(\forall z)((x * y) * z = x * (y * z))$, $(\forall x)(x * 1 = x)$ a $(\forall x)(\forall y)(\forall z)(x * (y + z) = (x * y) + (x * z))$. Teorie oborů integrity rozšiřuje teorii okruhů o axiom $(\forall x)(\forall y)(x * y = 0 \to x = 0 \vee y = 0)$. Teorie těles rozšiřuje teorii okruhů o axiom $(\forall x)(x \neq 0 \to (\exists y)(x * y = 1))$.

2.4.5 Příklad. Robinsonova aritmetika je teorie v jazyce $\{+, *, S, \mathbf{0}\}$, kde binární funkční symboly $+$ a $*$ značí součet a součin, unární funkční symbol S značí následníka, a symbol $\mathbf{0}$ je konstanta. Axiomy jsou následující:

(i) $(\forall x)(S(x) \neq \mathbf{0})$

(ii) $(\forall x)(\forall y)(S(x) = S(y) \to x = y)$

(iii) $(\forall x)(x + \mathbf{0} = x)$

(iv) $(\forall x)(\forall y)(x + S(y) = S(x + y))$

(v) $(\forall x)(x * \mathbf{0} = \mathbf{0})$

(vi) $(\forall x)(\forall y)(x * S(y) = x * y + x)$

Peanova aritmetika obsahuje kromě těchto šesti axiomů navíc dalších nekonečně mnoho axiomů tvořících *schema axiomů indukce*: pro každou aritmetickou formuli $\varphi(x)$ s jednou volnou proměnnou je následující formule axiomem.

$$[\varphi(\mathbf{0}) \wedge (\forall x)(\varphi(x) \to \varphi(S(x)))] \to (\forall x)\varphi(x)$$

Standardním modelem aritmetiky je struktura $\mathbb{N}$ přirozených čísel s obvyklými operacemi (v další kapitole ji podrobně definujeme), ale má i jiné modely.

Při práci v konkrétní teorii nás zajímají především její specifické důsledky, tj. taková tvrzení, ve kterých se projeví její axiomy. Logicky platné formule, které platí v každé realizaci jejího jazyka, nejsou z pohledu konkrétní teorie příliš zajímavé: platí i v modelech jakékoli jiné teorie se stejným jazykem. Například algebraik pracující v teorii grup se zajímá o vlastnosti, které plynou z axiomů teorie grup, zatímco logicky platná formule $(\forall x)(x * x = x * x)$ jej nezaujme, přestože platí v každé grupě. Predikátová logika stojí v pozadí jako vyjadřovací a důkazový aparát. Zde se ovšem zabýváme právě tímto obecným aparátem.

Směřujeme k větě o úplnosti predikátové logiky, podle které je daná formule dokazatelná v dané teorii právě tehdy, když je jejím logickým důsledkem. Jeden směr této ekvivalence je obsažen v následující větě.

2.4.6 Věta (o korektnosti)**.** *Buď T teorie v jazyce $\mathcal{L}$ a buď φ formule jazyka $\mathcal{L}$. Je-li φ dokazatelná v T, pak platí v každém jejím modelu.*

Důkaz. Buď $\varphi_1, \ldots, \varphi_n$ důkaz formule φ v teorii T, buď $\mathfrak{M} \models T$ libovolný model. Indukcí ukážeme, že každá φ_i platí v $\mathfrak{M}$ při libovolném ohodnocení.

(i) Je-li φ_i formule z T, je $\mathfrak{M} \models \varphi_i$ z definice.

(ii) Je-li φ_i axiom výrokové logiky, je tautologií, a snadno se ověří, že platí v $\mathfrak{M}$ (jakož i v každém jiném modelu jazyka $\mathcal{L}$) při libovolném ohodnocení.

(iii) Je-li φ_i axiom specifikace tvaru $(\forall x)\psi \to \psi_x[t]$, buď e libovolné ohodnocení proměnných v $\mathfrak{M}$. Pokud při tomto ohodnocení není v $\mathfrak{M}$ splněna formule $(\forall x)\psi$, pak celá implikace platí. V opačném případě je $\mathfrak{M} \models \psi[e(x/m)]$ pro každé $m \in M$, speciálně pro $t[e] \in M$, a tedy $\mathfrak{M} \models \psi_x[t][e]$.

(iv) Je-li φ_i axiom tvaru $(\forall x)(\psi \to \vartheta) \to (\psi \to (\forall x)\vartheta)$, kde x není volná v ψ, buď e libovolné ohodnocení proměnných, a uvažme jediný zajímavý případ, kdy $\mathfrak{M} \models (\forall x)(\psi \to \vartheta)[e]$. To znamená, že pro každé $m \in M$ je $\mathfrak{M} \models (\psi \to \vartheta)[e(x/m)]$, neboli máme buďto $\mathfrak{M} \not\models \psi[e(x/m)]$ nebo $\mathfrak{M} \models \vartheta[e(x/m)]$. V prvním případě je pak i $\mathfrak{M} \not\models \psi[e]$, neboť proměnná x nemá volný výskyt v ψ, ve druhém případě je z definice $\mathfrak{M} \models (\forall x)\vartheta[e]$. V obou případech je tedy $\mathfrak{M} \models (\psi \to (\forall x)\vartheta)[e]$.

(v) Je-li φ_i některý z axiomů rovnosti, snadno se ověří přímo z definice splňování, že platí v $\mathfrak{M}$ (jakož i v každém jiném modelu jazyka $\mathcal{L}$).

(vi) Je-li φ_i odvozena z nějakých předchozích φ_j a $\varphi_j \to \varphi_i$ pomocí modus ponens, pak pro libovolné ohodnocení e máme z indukčního předpokladu $\mathfrak{M} \models \varphi_j[e]$ a $\mathfrak{M} \models (\varphi_j \to \varphi_i)[e]$. Již z výrokové logiky víme, že modus ponens je korektní, tj. že za těchto předpokladů je také $\mathfrak{M} \models \varphi_i[e]$.

(vii) Je-li φ_i tvaru $(\forall x)\varphi_j$ odvozena z nějaké předchozí φ_j pomocí pravidla generalizace, pak pro libovolné ohodnocení e máme z indukčního předpokladu $\mathfrak{M} \models \varphi_j[e]$. Speciálně tedy $\mathfrak{M} \models \varphi_j[e(x/m)]$ pro každé $m \in M$, takže z definice splňování je $\mathfrak{M} \models (\forall x)\varphi_j[e]$, neboli $\mathfrak{M} \models \varphi_i[e]$.

Ukázali jsme, že každá formule φ_i z důkazu $\varphi_1, \ldots, \varphi_n$ platí v každém modelu $\mathfrak{M} \models T$ při jakémkoli ohodnocení. Tím je speciálně pro φ_n věta dokázána. □

Z důkazu věty o korektnosti vidíme, že axiomy rovnosti, axiomy predikátové logiky, jakož i všechny formule z nich odvozené pomocí odvozovacích pravidel, platí nejen v modelu dané teorie, ale v každé realizaci jejího jazyka. Jinými slovy, *všechny formule dokazatelné v predikátové logice jsou logicky platné*.

Věta o korektnosti dává také návod, jak o nějaké formuli φ předvést, že *není* v dané teorii T dokazatelná: stačí najít model $\mathfrak{M} \models T$ a nějaké ohodnocení, při kterém φ není splněna. Například formule $(\forall x)(\forall y)(x * y = y * x)$ nemůže být dokazatelná v teorii grup, protože neplatí v nekomutativních grupách.

2.4.7 Důsledek. *Teorie, která má model, je bezesporná.*

Důkaz. Buď $\mathfrak{M} \models T$ a buď φ libovolná uzavřená formule jazyka teorie T. Podle definice splňování platí buďto $\mathfrak{M} \models \varphi$ nebo $\mathfrak{M} \models \neg\varphi$. Podle věty o korektnosti tedy buďto formule $\neg\varphi$ nebo formule φ není v T dokazatelná. □

Modelem predikátové logiky, jakožto prázdné teorie v jazyce $\mathcal{L}$, je každá realizace $\mathfrak{M} \models \mathcal{L}$. Podle předchozí věty tedy *predikátová logika je bezesporná*.

Věta o úplnosti Každá formule dokazatelná v predikátové logice je podle věty o korektnosti logicky platná. Dokážeme nyní i opačné tvrzení: každá logicky platná formule má formální důkaz v predikátové logice. Tím bude ukázáno, že syntax a sémantika Hilbertova systému si plně odpovídají. Tak jako ve větě o korektnosti, budeme zkoumat obecněji dokazatelnost v nějaké dané teorii.

2.4.8 Věta (Gödel). *Buď $\mathcal{L}$ jazyk predikátové logiky, buď T teorie v jazyce $\mathcal{L}$. Pak pro každou formuli φ jazyka $\mathcal{L}$ je $T \vdash \varphi$ právě když $T \models \varphi$.*

2.4.9 Věta (Gödel). *Teorie je bezesporná, právě když má model.*

Implikace zleva doprava v první větě je právě věta o korektnosti, a implikace zprava doleva ve druhé větě je věta 2.4.7. Všimněme si nejprve, že první věta plyne z druhé. Je-li $T \models \varphi$, máme ukázat, že $T \vdash \varphi$. Pokud ne, pak $T, \neg\overline{\varphi}$ je bezesporná, tedy podle 2.4.9 má model, což ale při $T \models \varphi$ není možné.

Zbývá tedy pro danou bezespornou teorii T v jazyce $\mathcal{L}$ najít model. Máme popsat nějakou nosnou množinu (univerzum), a na ní realizovat speciální symboly jazyka $\mathcal{L}$ tak, abychom splnili všechny axiomy teorie T. *A priori* se však nenabízí žádná množina, jejíž prvky bychom mohli použít jako individua, ani přirozený způsob, jak na takové množině zavést relační strukturu. Výchozí myšlenka, pocházející od L. Henkina, je následující.

Předložená teorie T je čistě syntaktický útvar — máme k dispozici jen výrazy jejího jazyka: proměnné, speciální symboly, termy, formule, axiomy.

Z těchto výrazů pak jen termy popisují nějaké „objekty“, a mají-li hrát roli individuí, nesmí jejich identita záviset na proměnných. Hledaný model tedy postavíme z termů bez proměnných, které budou samy svou realizací. Funkce budeme realizovat přirozeným způsobem: aplikace funkčního symbolu na termy je opět term. Relační symboly budeme realizovat s úmyslem, se kterým tento model budujeme: za splněné relace prohlásíme ty, které jsou v teorii T dokazatelné.

Jakmile přijmeme tuto ideu za svou, je konstrukce modelu dosti přirozená. Na cestě však musíme překonat několik technických překážek.

(i) Předně, pokud jazyk $\mathcal{L}$ neobsahuje žádnou konstantu, pak neexistují ani žádné termy bez proměnných — model však musí být *neprázdnou* množinou.

(ii) Je-li $\mathcal{L}$ jazyk s rovností a je-li v teorii T pro nějaké dva různé termy s, t dokazatelná formule $s = t$, nemůžeme termy s, t použít jako dvě různá individua.

(iii) Pro každou uzavřenou formuli φ jazyka $\mathcal{L}$ musí být v modelu splněna buďto formule φ nebo $\neg\varphi$. V teorii T však nemusí být ani φ ani $\neg\varphi$ dokazatelná.

(iv) Je-li v teorii T dokazatelná nějaká formule tvaru $(\exists x)\varphi$, potřebujeme v modelu pomocí nějakého termu t bez proměnných splnit formuli $\varphi_x[t]$; přitom v teorii T nemusí být žádná taková formule dokazatelná.

Problém (ii) lze vyřešit snadno: množinu termů nejprve faktorizujeme podle rovnosti dokazatelné v T. Ostatní překážky prozatím obejdeme tím, že sestrojíme model jen pro takovou teorii, které se tyto problémy netýkají.

2.4.10 Definice. Bezesporná teorie T v jazyce $\mathcal{L}$ je *úplná*, pokud pro každou uzavřenou formuli φ jazyka $\mathcal{L}$ je buďto $T \vdash \varphi$ nebo $T \vdash \neg\varphi$. Říkáme též, že T *rozhoduje* každou sentenci jazyka $\mathcal{L}$.

Například teorie grup nerozhoduje sentenci $(\forall x)(\forall y)(x * y = x * y)$, a není tedy úplnou teorií. Podobně teorie okruhů není úplná (například proto, že některé okruhy jsou tělesem a jiné ne), zato teorie algebraicky uzavřených těles dané charakteristiky je úplná. Ani teorie lineárních uspořádání není úplná, neboť nerozhoduje sentenci $(\forall x)(\exists y)(x < y)$, zatímco teorie neomezených hustých lineárních uspořádání je úplná, což ukážeme později.

Pro každý model $\mathfrak{M} \models \mathcal{L}$ je $Th(\mathfrak{M}) = \{\varphi; \mathfrak{M} \models \varphi\}$ úplná teorie v jazyce $\mathcal{L}$. Teorie T je úplná, pokud je $T = Th(\mathfrak{M})$ pro nějaký (každý) model $\mathfrak{M} \models T$. Například $Th(\mathbb{N})$ je úplné bezesporné rozšíření Peanovy aritmetiky.

Předpoklad uzavřenosti formule v definici je podstatný: má-li φ volnou proměnnou, jako například aritmetická formule $x = \mathbf{0}$, může se snadno stát, že při nějakém ohodnocení proměnných platí φ a při jiném $\neg\varphi$. Podle věty o korektnosti pak ani jedna nemůže být dokazatelná.

2.4.11 Definice. Teorie T v jazyce $\mathcal{L}$ je *Henkinova*, pokud pro každou uzavřenou formuli $(\exists x)\varphi$ jazyka $\mathcal{L}$ existuje konstanta c taková, že $T \vdash (\exists x)\varphi \to \varphi_x[c]$.

Henkinovské konstanty jsou „svědci existence.“ Jde o to, že pokud teorie T dokazuje existenci individua s nějakou vlastností, má již pro takové individum jméno v jazyce. Žádná z výše uvedených teorií není henkinovská. Například teorie uspořádání nemá svědčící konstantu pro sentenci $(\exists x)(\forall y)(x \leq y)$, ostatně nemá vůbec žádné konstanty. Můžeme takovou konstantu (řekněme 0) do jazyka přidat, a teorii rozšířit o axiom $(\exists x)(\forall y)(x \leq y) \to (\forall y)(0 \leq y)$. Ani taková

teorie nebude henkinovská: to co jsme provedli pro jednu existenční sentenci má henkinovská teorie obsahovat pro všechny existenční sentence.

Později ukážeme, že každou bezespornou teorii lze bezesporně rozšířit do úplné henkinovské teorie. Úplná teorie se pak vyhne překážce (iii), a Henkinovská teorie má dostatek konstant a termů bez proměnných na to, aby se vyhnula problémům (i) a (iv). Pro takovou teorii již nic nebrání konstrukci modelu.

2.4.12 Věta (Henkin). *Každá úplná henkinovská teorie má model.*

Důkaz. Je-li T daná teorie a je-li $\mathcal{L}$ její jazyk, označme jako τ množinu všech termů jazyka $\mathcal{L}$, ve kterých se nevyskytují žádné proměnné. Pro term $t \in \tau$ pak označme jako $[t] = \{s \in \tau; T \vdash s = t\}$ jeho *ekvivalenční třídu*; snadno se ověří, že dokazatelná rovnost je ekvivalence na τ. Buď konečně M množina těchto ekvivalenčních tříd; ta bude univerzem hledaného modelu.

Funkční a relační symboly jazyka $\mathcal{L}$ realizujeme v M následovně. Je-li f nějaký n-ární funkční symbol a jsou-li $t_1, \ldots, t_n \in \tau$, buď $f^{\mathfrak{M}}([t_1], \ldots, [t_n])$ individuum $[f(t_1, \ldots, t_n)]$. Je-li R nějaký n-ární relační symbol různý od rovnosti, buď $([t_1], \ldots, [t_n]) \in R^{\mathfrak{M}}$ právě když $T \vdash R(t_1, \ldots, t_n)$. Z axiomů rovnosti plyne, že tyto definice nezáleží na volbě reprezentantů ekvivalenčních tříd: pro $s_i \in [t_i]$ je $f^{\mathfrak{M}}([s_1], \ldots, [s_n])$ totéž individuum jako $f^{\mathfrak{M}}([t_1], \ldots, [t_n])$, a $([s_1], \ldots, [s_n]) \in R^{\mathfrak{M}}$ platí právě tehdy, když platí $([t_1], \ldots, [t_n]) \in R^{\mathfrak{M}}$. Zároveň jsou relace $R^{\mathfrak{M}}$ díky bezespornosti T zavedeny konzistentně: T nikdy nedokazuje zároveň $R(t_1, \ldots, t_n)$ a $\neg R(t_1, \ldots, t_n)$.

Zbývá ověřit, že tato relační struktura je modelem teorie T. Ukážeme indukcí, že pro každou uzavřenou formuli φ jazyka $\mathcal{L}$ je $\mathfrak{M} \models \varphi$ právě když $T \vdash \varphi$.

(a) Je-li φ tvaru $t_1 = t_2$, pak $\mathfrak{M} \models \varphi$ platí právě tehdy, když $[t_1]$ a $[t_2]$ je totéž individuum, tedy právě když $T \vdash t_1 = t_2$.

(b) Buď φ uzavřená atomická formule tvaru $R(t_1, \ldots, t_n)$. Jelikož termy t_i nemají žádné proměnné, nezávisí jejich hodnoty $t_i[e]$ na ohodnocení e: pro term t bez proměnných je vždy $t[e] = [t]$. Z definice splňování je tedy $\mathfrak{M} \models \varphi$ právě když $([t_1], \ldots, [t_n]) \in R^{\mathfrak{M}}$, tedy právě když $T \vdash R(t_1, \ldots, t_n)$.

(c) Buď φ uzavřená formule tvaru $\neg\psi$, přičemž pro formuli ψ je již tvrzení dokázáno. Z definice je $\mathfrak{M} \models \varphi$ právě tehdy, když $\mathfrak{M} \not\models \psi$, což je podle indukčního předpokladu právě když $T \nvdash \psi$. Přitom ψ je uzavřená formule, a T je úplná teorie, takže $T \nvdash \psi$ právě když $T \vdash \neg\psi$, neboli když $T \vdash \varphi$.

(d) Buď φ uzavřená formule tvaru $\psi \to \vartheta$, přičemž pro formule ψ, ϑ je již tvrzení dokázáno. Z definice je $\mathfrak{M} \models \varphi$ právě když $\mathfrak{M} \not\models \psi$ nebo $\mathfrak{M} \models \vartheta$, tedy právě když $\mathfrak{M} \models \neg\psi$ nebo $\mathfrak{M} \models \vartheta$, neboť ψ, ϑ jsou uzavřené formule. To je podle indukčního předpokladu právě když $T \vdash \neg\psi$ nebo $T \vdash \vartheta$. Snadno ověříme, že to je právě když $T \vdash \psi \to \vartheta$: je-li $T \vdash \neg\psi$, použijeme dokazatelnou formuli $\neg\psi \to (\psi \to \vartheta)$ a modus ponens; je-li $T \vdash \vartheta$, použijeme axiom $\vartheta \to (\psi \to \vartheta)$ a modus ponens. Pokud naopak $T \vdash \psi \to \vartheta$, využijeme opět toho, že T je úplná: je buďto $T \vdash \psi$, načež $T \vdash \vartheta$ pomocí modus ponens, nebo $T \vdash \neg\psi$. Celkem tedy máme $\mathfrak{M} \models \psi \to \vartheta$ právě tehdy, když $T \vdash \psi \to \vartheta$.

(e) Buď φ uzavřená formule tvaru $(\forall x)\psi$, přičemž pro každou instanci formule ψ je již tvrzení dokázáno. Z úplnosti teorie T je buďto $T \vdash \varphi$ nebo $T \vdash \neg\varphi$. Je-li $T \vdash \varphi$, tj. $T \vdash (\forall x)\psi$, je podle axiomu specifikace také $T \vdash \psi_x[t]$ pro každé $t \in \tau$. To podle indukčního předpokladu znamená, že $\mathfrak{M} \models \psi[e(x/[t])]$ pro

každé ohodnocení e a každé individuum $[t] \in M$, takže $\mathfrak{M} \models (\forall x)\psi$. Je-li naopak $T \vdash \neg\varphi$, tj. $T \vdash \neg(\forall x)\psi$, máme $T \vdash (\exists x)\neg\psi$. Přitom $(\exists x)\neg\psi$ je uzavřená formule a T je Henkinova teorie, takže pro nějakou konstantu c jazyka $\mathcal{L}$ máme $T \vdash (\exists x)\neg\psi \to \neg\psi_x[c]$. Pomocí modus ponens je pak $T \vdash \neg\psi_x[c]$, a podle indukčního předpokladu formule ψ není splněna v $\mathfrak{M}$ při ohodnocení $e(x) = [c]$, takže $\mathfrak{M} \not\models (\forall x)\psi$. Celkem tedy platí $\mathfrak{M} \models (\forall x)\psi$ právě tehdy, když $T \vdash (\forall x)\psi$.

Závěr je již snadný: je-li formule φ axiomem teorie T a je-li $\overline{\varphi}$ její uzávěr, pak podle věty o uzávěru je $T \vdash \overline{\varphi}$. Pro uzavřenou formuli $\overline{\varphi}$ podle předchozího máme $\mathfrak{M} \models \overline{\varphi}$, a tedy také $\mathfrak{M} \models \varphi$. Relační struktura $\mathfrak{M}$ je tedy modelem T. □

Model zkonstruovaný v předchozí větě se nazývá *kanonická struktura* pro teorii T. Jeho univerzum tvoří konstantní termy jazyka $\mathcal{L}$, které jsou samy svou realizací. O takovém modelu říkáme, že je „sestrojen ze slov“ nebo „ze jmen.“

Henkinovské zúplnění Dokázali jsme větu o úplnosti zatím jen pro speciální případ úplné henkinovské teorie. Nyní ukážeme, že každou bezespornou teorii lze do takové teorie bezesporně rozšířit. Úplné henkinovské rozšíření má potom svůj kanonický model, který určuje i model původní teorie.

2.4.13 Definice. Řekneme, že jazyk $\mathcal{L}'$ *rozšiřuje* jazyk $\mathcal{L}$, pokud každý speciální symbol jazyka $\mathcal{L}$ je i symbolem jazyka $\mathcal{L}'$, stejného druhu a se stejnou četností. Teorie T' *rozšiřuje* teorii T, pokud jazyk teorie T' rozšiřuje jazyk teorie T a každá formule dokazatelná v teorii T je dokazatelná i v teorii T'.

Například jazyk teorie okruhů rozšiřuje jazyk teorie grup o dva nové symboly $*$ a $\mathbf{1}$; ostatní symboly jsou přítomny v obou jazycích, a jsou stejného druhu i četnosti. Teorie těles rozšiřuje teorii oborů integrity v témže jazyce, přestože neobsahuje všechny její axiomy. Snadno se nahlédne, že T' rozšiřuje T právě tehdy, když všechny axiomy teorie T jsou dokazatelné v T'. Pokud je T' bezesporné rozšíření teorie T, pak i T je bezesporná.

2.4.14 Definice. Rozšíření T' teorie T je *konzervativní*, pokud každá formule jazyka teorie T, která je dokazatelná v T', je dokazatelná již v teorii T.

Konzervativní rozšíření má silnější výrazové prostředky, ale nedokazuje žádné další formule původního jazyka. Uvidíme později, že například rozšíření teorie o definici nového predikátu je konzervativní. Například rozšíření aritmetiky o nový predikát pro dělitelnost a příslušný definující axiom zesílí výrazové prostředky jazyka aritmetiky, ale žádná nová tvrzení o přirozených číslech tím nezískáme. Snadno se nahlédne, že konzervativní rozšíření teorie T je bezesporné právě tehdy, když T je bezesporná.

2.4.15 Věta (Henkin). *Každá teorie má konzervativní henkinovské rozšíření.*

Důkaz. Hledané rozšíření získáme přidáním nových konstant a nových axiomů. Musíme přitom zaručit, aby ke každé uzavřené formuli tvaru $(\exists x)\varphi$ existovala „svědčící“ konstanta. Za každou formuli φ jazyka $\mathcal{L}$ teorie T s jednou volnou proměnnou x přidejme do jazyka novou konstantu c_φ a do teorie jako nový axiom formuli $(\exists x)\varphi \to \varphi_x[c_\varphi]$. Tím jsme rozšířili původní jazyk $\mathcal{L}$ o množinu C_1 nových konstant do jazyka $\mathcal{L}_1$, a původní teorii T do teorie T_1 v jazyce $\mathcal{L}_1$.

Ta stále není henkinovská, v novém jazyce existují nové formule tvaru $(\exists x)\varphi$, ke kterým musíme opět přidat nové konstanty a axiomy. Tím rozšíříme jazyk $\mathcal{L}_1$ o množinu C_2 nových konstant do jazyka $\mathcal{L}_2$, a teorii T_1 do teorie T_2 v jazyce $\mathcal{L}_2$. Takto postupujeme dále indukcí přes přirozená čísla; vytváříme množiny nových konstant C_n, jazyky $\mathcal{L}_n \subseteq \mathcal{L}_{n+1}$ a teorie $T_n \subseteq T_{n+1}$. Nakonec položíme $\mathcal{L}_H = \bigcup \mathcal{L}_n$ a $T_H = \bigcup T_n$. Teorie T_H v jazyce $\mathcal{L}_H$ je potom henkinovské rozšíření. Zbývá ukázat, že je konzervativní.

Buď ψ formule jazyka $\mathcal{L}$ dokazatelná v T_H, a buďte $\vartheta_1, \ldots, \vartheta_k$ všechny henkinovské axiomy použité v důkazu formule ψ. Je tedy $T, \vartheta_1, \ldots, \vartheta_k \vdash \psi$. Podle věty o dedukci máme $T \vdash \vartheta_1 \to \cdots \to \vartheta_k \to \psi$, neboť všechny ϑ_i jsou uzavřené formule. Za cenu případného přerovnání formulí ϑ_i v implikaci můžeme předpokládat, že ϑ_1 je henkinovský axiom obsahující konstantu z množiny C_n s maximálním indexem (mezi všemi konstantami z ϑ_i). Tedy ϑ_1 je formule tvaru $(\exists x)\varphi \to \varphi_x[c_\varphi]$ a konstanta c_φ se nevyskytuje nikde v $\vartheta_2, \ldots, \vartheta_k$ ani v ψ. Podle věty o konstantách je potom $T \vdash ((\exists x)\varphi \to \varphi_x[y]) \to (\vartheta_2 \to \cdots \to \vartheta_k \to \psi)$, kde y je nějaká proměnná, která se nevyskytuje nikde v $\vartheta_1, \ldots, \vartheta_k$ ani v ψ. Pomocí 2.3.4 pak odvodíme $T \vdash (\exists y)((\exists x)\varphi \to \varphi_x[y]) \to (\vartheta_2 \to \cdots \to \vartheta_k \to \psi)$, a pomocí prenexní operace také $T \vdash ((\exists x)\varphi \to (\exists y)\varphi_x[y]) \to (\vartheta_2 \to \cdots \to \vartheta_k \to \psi)$. Přitom podle věty o variantách je $\vdash (\exists x)\varphi \to (\exists y)\varphi_x[y]$, takže pomocí modus ponens je $T \vdash \vartheta_2 \to \cdots \to \vartheta_k \to \psi$. Tak postupně dojdeme až k $T \vdash \psi$. □

2.4.16 Věta (Lindenbaum). *Každá bezesporná teorie má bezesporné zúplnění.*

Důkaz. Buď T bezesporná teorie. Uvažme systém $\mathcal{S}$ všech bezesporných rozšíření S teorie T v témže jazyce. Potom systém $\mathcal{S}$ uspořádaný inkluzí splňuje předpoklady Zornova lemmatu.[13] Předně je neprázdný, jelikož $T \in \mathcal{S}$. Je-li $\mathcal{C} \subseteq \mathcal{S}$ řetězec bezesporných rozšíření, je také $\bigcup \mathcal{C} \in \mathcal{S}$. Nad $T \in \mathcal{S}$ tedy existuje nějaký maximální prvek $S \in \mathcal{S}$. Teorie S je maximální bezesporné rozšíření teorie T. Ukážeme, že S je úplná. Kdyby ne, pak pro nějakou sentenci φ není ani φ ani $\neg\varphi$ dokazatelná v S. To podle 2.3.14 znamená, že $S \cup \{\varphi\}$ je vlastní bezesporné rozšíření teorie S, což není možné, jelikož S je maximální. □

Nyní již máme k dispozici všechny prostředky potřebné k důkazu Gödelovy věty o úplnosti. Je-li T daná bezesporná teorie, získáme nejprve konzervativní henkinovské rozšíření T_H podle 2.4.15. Teorie T_H je bezesporná, neboť konzervativně rozšiřuje T. Zúplnění T' teorie T_H ve stejném jazyce pak získáme podle 2.4.16. Teorie T' je úplná a henkinovská, má tedy model $\mathfrak{M}' \models T'$. Stačí pak vzít realizaci $\mathfrak{M}$ jazyka $\mathcal{L}$ teorie T, která má stejnou nosnou množinu jako $\mathfrak{M}'$, ale realizuje jen[14] symboly jazyka $\mathcal{L}$, stejným způsobem jako model $\mathfrak{M}'$. Potom každá formule jazyka $\mathcal{L}$ je splněna v $\mathfrak{M}$ právě když je splněna v $\mathfrak{M}'$. Přitom všechny axiomy teorie T jsou dokazatelné v T', neboť T' rozšiřuje T, takže jsou splněny v $\mathfrak{M}'$, a tedy také v $\mathfrak{M}$. Tedy $\mathfrak{M}$ je model teorie T.

Podle věty o úplnosti je sémantika predikátové logiky, totiž platnost v modelech, plně zachycena syntaktickými, tedy finitárními prostředky: zvolené axiomy a odvozovací pravidla Hilbertova kalkulu postačují k tomu, aby bylo možné dokázat každou platnou formuli.

[13] Je známo, že důkaz Lindenbaumovy věty vyžaduje nějakou formu principu maximality, potažmo axiomu výběru. Vrátíme se k této otázce v kapitole o teorii množin.

[14] Říkáme pak, že $\mathfrak{M}$ je *redukce* struktury $\mathfrak{M}'$. Tak můžeme například na každý okruh $(R, +, *, 0, 1)$ hledět zároveň jen jako na grupu $(R, +, 0)$. V našem případě ignorujeme při redukci $\mathfrak{M}'$ do $\mathfrak{M}$ to, že individua modelu $\mathfrak{M}$ zároveň realizují henkinovské konstanty.

Kompaktnost Z úplnosti predikátové logiky dostáváme ihned tento důsledek:

2.4.17 Věta (o kompaktnosti)**.** *Buď T teorie a φ formule v témže jazyce. Pak*
(i) $T \models \varphi$ *právě když* $T_0 \models \varphi$ *pro nějaký konečný fragment* $T_0 \subseteq T$.
(ii) T *má model právě tehdy, když má model každá její konečná část.*

Důkaz. (i) Podle věty o úplnosti je $T \models \varphi$ právě když $T \vdash \varphi$. Přitom důkaz formule φ je konečná posloupnost, takže používá jen konečně mnoho axiomů z nějaké konečné části $T_0 \subseteq T$. Je tedy $T_0 \vdash \varphi$, takže i $T_0 \models \varphi$. (ii) Podle věty o úplnosti má teorie model právě když je bezesporná. Přitom důkaz případného sporu v T je důkazem již v nějaké její konečné části. □

Kompaktnost je obecný princip matematiky založené na teorii množin, jejíž základy popíšeme v příští kapitole. Zatím ukážeme na příkladech dopad kompaktnosti v aritmetice, elementární algebře a teorii grafů. Kompaktnost predikátové logiky klade jistá omezení na to, co lze jazykem prvního řádu vyjádřit: tělesa charakteristiky nula nelze axiomatizovat žádnou konečnou teorií v jazyce těles, a torzní grupy nelze popsat dokonce žádnou množinou formulí jazyka teorie grup. Taková omezení jazyka prvního řádu lze obejít právě výstavbou matematiky na množinovém základě, jak jsme stručně zmínili již v úvodu.

2.4.18 Lemma. *Buďte S, T ekvivalentní teorie, tj. mají stejný jazyk a dokazují tytéž formule. Je-li S konečná, je T ekvivalentní už s nějakou konečnou $T_0 \subseteq T$.*

Důkaz. Pro každou formuli φ z teorie S je podle předpokladu $T \vdash \varphi$. Existuje tedy nějaká konečná $T_\varphi \subseteq T$ tak, že $T_\varphi \vdash \varphi$. Teorie $T_0 = \bigcup_{\varphi \in S} T_\varphi$ je konečnou částí T, přitom dokazuje všechny formule z S, a tedy i všechny formule z T. □

Jinými slovy, pokud nějaká teorie není ekvivalentní s žádným svým konečným fragmentem, nemá ani žádnou jinou ekvivalentní konečnou axiomatiku.

2.4.19 Příklad. Pro n přirozené označme term $1 + 1 + \cdots + 1$ (n sčítanců) jazyka teorie těles jako $n \times 1$, a formuli $n \times 1 = 0$ jako χ_n. Pokud dané těleso splňuje nějakou sentenci χ_n, pak první takové $n \in \mathbb{N}$ je jeho *charakteristika*. Pokud nesplňuje žádnou χ_n, říkáme, že má *charakteristiku nula*. Například $\mathbb{R}$ má charakteristiku nula a $\mathbb{Z}_p$ má charakteristiku p. Existují tedy konečná tělesa libovolně velké konečné charakteristiky. Rozšířením teorie těles o všechny formule $\neg\chi_n$ vznikne *teorie těles charakteristiky nula*, označme ji T. Ukážeme, že nemá žádný konečný ekvivalent.

Buď totiž $T_0 \subseteq T$ nějaký konečný fragment T. Konečná teorie T_0 obsahuje jen konečně mnoho z formulí $\neg\chi_n$; buď m index poslední z nich. Každé těleso charakteristiky větší než m je pak modelem teorie T_0, přitom nemá charakteristiku nula. V logice prvního řádu tedy nelze tělesa charakteristiky nula popsat žádnou konečnou množinou formulí v jazyce těles.

2.4.20 Příklad. Tělesa nenulové charakteristiky nelze axiomatizovat žádnou teorií v jazyce těles. Buď totiž T teorie těles charakteristiky nula, a předpokládejme, že U je nějaká axiomatizace těles nenulové charakteristiky. Víme, že T je nutně nekonečná, a každý její konečný fragment je splněn v nějakém modelu teorie U. To znamená, že každý konečný fragment teorie $T \cup U$ má model. Podle věty o kompaktnosti pak $T \cup U$ má model, což není možné.

2.4.21 Příklad. Je-li $(G, +, -, 0)$ grupa, označme pro $x \in G$ a přirozené n prvek $x + x + \cdots + x$ (n sčítanců) jako $n \times x$. Řekneme, že grupa G je *torzní*, pokud pro každé $x \in G$ existuje přirozené n tak, že $n \times x = 0$. Nejmenší takové n je potom *řád* prvku $x \in G$. Jinými slovy, G je torzní, pokud každý její prvek je konečného řádu. Například každá grupa $\mathbb{Z}_n$ je torzní, zatímco $\mathbb{R}$ není torzní.

Ukážeme, že množina všech formulí jazyka teorie grup, které platí ve všech torzních grupách, je zároveň splněna v nějaké grupě, která není torzní. Tím bude ukázáno, že torzní grupy nelze popsat žádnou množinou formulí jazyka grup.

Buď G torzní grupa, která má prvky neomezeně velkých konečných řádů.[15] Buď T množina všech sentencí jazyka teorie grup, které platí v G, rozšířená o všechny sentence $n \times c \neq 0$ v jazyce obohaceném o konstantu c. Potom každá konečná $T_0 \subseteq T$ má model: je splněna v grupě G, pokud konstantu c realizujeme prvkem řádu většího než každé n z konečně mnoha $n \times c \neq 0$ obsažených v T_0.

Podle věty o kompaktnosti tedy T má nějaký model H. To je grupa, ve které platí stejné sentence jazyka teorie grup jako v G, ale H není torzní, protože prvek, který v grupě H realizuje konstantu c, nemůže být konečného řádu.

2.4.22 Příklad. Aritmetické termy tvaru $S(\dots(S(0))\dots)$ se nazývají *numerály*; bývá zvykem označovat je krátce jako $\overline{n}$, je-li symbol S použit n-krát. Tedy například $\overline{4}$ je zkratka za $S(S(S(S(0))))$.

Přidejme k jazyku aritmetiky novou konstantu c a rozšiřme Peanovu aritmetiku do teorie T přidáním všech formulí $c \neq \overline{n}$. Potom každá konečná $T_0 \subseteq T$ má model: je splněna ve standardním modelu $\mathbb{N}$, pokud konstantu c realizujeme číslem větším než každý z konečně mnoha numerálů zmíněných v T_0.

Podle věty o kompaktnosti má tedy teorie T nějaký model $\mathfrak{M}$. Individuum $c^{\mathfrak{M}} \in M$, které v modelu $\mathfrak{M}$ realizuje konstantu c, nemůže zároveň realizovat žádný numerál $\overline{n}$. Tedy $\mathfrak{M}$ nemůže být isomorfní se standardním modelem $\mathbb{N}$, ve kterém naopak každé individuum (tj. přirozené číslo) je realizací odpovídajícího numerálu. Říkáme, že $\mathfrak{M}$ je *nestandardní model aritmetiky*.[16]

2.4.23 Příklad. Uspořádané těleso F je *Archimedovské*, pokud pro každé $a \in F$ existuje $k \in \mathbb{N}$ takové, že $k \times \mathbf{1} > a$. Například $(\mathbb{R}, +, *, 0, 1, <)$ je Archimedovsky uspořádané těleso. Podobně jako v předchozím příkladě lze ukázat, že existuje uspořádané těleso, které není Archimedovské.

2.4.24 Příklad. Řekneme, že rozklad $V = A \cup B$ grafu $G = (V, E)$ je *nemilý*, pokud každý vrchol $x \in V$ má v druhé části rozkladu alespoň tolik sousedů, jako ve své vlastní. Snadno se ukáže, že pro konečný graf takový rozklad existuje (uvažte maximální řez). Ukážeme, že takový rozklad existuje i v případě, kdy graf G je *lokálně konečný*, tj. pokud každý vrchol je konečného stupně.

Buď $\mathcal{L}$ jazyk obsahující konstantu c_x za každý vrchol $x \in V$ a dva unární predikáty R_A, R_B. V tomto jazyce formulujeme následující teorii T. Prvním axiomem je formule $(\forall x)(R_A(x) \leftrightarrow \neg R_B(x))$. Pro každý vrchol $x \in V$ označme jako $N_x \subseteq V$ množinu jeho sousedů; buď potom O_x systém všech $S \subseteq N_x$, které obsahují ze všech jeho sousedů alespoň polovinu. Za každý vrchol $x \in V$ jsou

[15] Takovou grupou je například direktní suma všech grup $\mathbb{Z}_n$, tj. množina všech $x \in \prod \mathbb{Z}_n$ splňujících $x_n = \mathbf{0}$ všude až na konečně mnoho n, s operacemi po složkách.

[16] Nabízí se otázka, jaké je postavení standardního modelu $\mathbb{N}$ mezi ostatními modely Peanovy aritmetiky. Dá se ukázat, že „začátek“ každého modelu aritmetiky je isomorfní s $\mathbb{N}$.

potom axiomy teorie T také formule $R_A(x) \to \bigvee_{S \in O_x} \bigwedge_{y \in S} R_B(c_y)$ a formule $R_B(x) \to \bigvee_{S \in O_x} \bigwedge_{y \in S} R_A(c_y)$. Graf G je lokálně konečný, takže takové formule lze vůbec napsat: použité konjunkce a disjunkce jsou konečné.

Každý konečný fragment T' teorie T má model: je-li $T' \subseteq T$ konečná, buď $H \subseteq G$ konečný podgraf určený konečně mnoha vrcholy $x \in V$, pro které se konstanta c_x vyskytuje v T'; potom H má nemilý rozklad, symboly jazyka $\mathcal{L}$ realizujeme v H očividným způsobem, a získáme tak model fragmentu T'.

Podle věty o kompaktnosti tedy existuje i model $\mathfrak{M} \models T$. Položme pak $A = \{x \in V; \mathfrak{M} \models R_A(c_x)\}$ a $B = \{x \in V; \mathfrak{M} \models R_B(c_x)\}$. Snadno se nahlédne, že $A \cup B = V$ je hledaný rozklad.

2.5 Rozšiřování teorií

Při práci v nějaké matematické teorii je běžné rozšiřovat její základní jazyk o nové symboly popisující nové objekty, vlastnosti a vztahy, které postupně objevujeme. Například při studiu přirozených čísel brzy odhalíme vztah dělitelnosti, existenci nejmenších společných násobků, atd. Je potom přirozené rozšířit původní jazyk o binární predikát $x|y$ a binární funkční symbol $lcm(x, y)$, i když bychom se obešli i bez nich. Podobně při studiu teorie množin brzy narazíme na existenci prázdné množiny, a je přirozené ji pojmenovat nějakou konstantou, typicky $\emptyset$, zavést binární funkční symbol $\cap$ pro průnik, atd. V tomto oddíle popíšeme formální náležitosti takového rozšiřování.

2.5.1 Věta (o novém predikátu). *Buď T teorie v jazyce $\mathcal{L}$, a buď δ formule jazyka $\mathcal{L}$, jejíž volné proměnné jsou právě $x_1, \ldots, x_n$. Buď $\mathcal{L}'$ rozšíření jazyka $\mathcal{L}$ o nový n-ární relační symbol R, a buď T' teorie v jazyce $\mathcal{L}'$, která rozšiřuje teorii T o axiom $R(x_1, \ldots, x_n) \leftrightarrow \delta(x_1, \ldots, x_n)$. Potom T' konzervativně rozšiřuje T.*

Důkaz. Ukážeme nejprve, že ke každé formuli φ jazyka $\mathcal{L}'$ existuje formule φ^* jazyka $\mathcal{L}$ taková, že $T' \vdash \varphi \leftrightarrow \varphi^*$. Pokud φ neobsahuje symbol R, stačí za φ^* vzít formuli φ. V opačném případě buď δ' nějaká varianta formule δ, která nemá žádnou proměnnou společnou s φ. Každý výskyt atomické podformule tvaru $R(y_1, \ldots, y_n)$ ve formuli φ potom nahraď instancí $\delta'(y_1, \ldots, y_n)$ formule δ'. Tak vznikne formule φ^* jazyka $\mathcal{L}$. Podle věty o ekvivalenci je $T' \vdash \varphi \leftrightarrow \varphi^*$.

Zbývá ukázat, že T' je konzervativní rozšíření T. K tomu podle předchozího stačí ukázat, že při $T' \vdash \varphi$ je $T \vdash \varphi^*$, neboť pro formuli φ jazyka $\mathcal{L}$ je φ^* sama formule φ. Buď tedy $\varphi_1, \ldots, \varphi_n$ nějaký důkaz formule φ v teorii T'. Indukcí ukážeme, že pro každé $i \leq n$ je $T \vdash \varphi_i^*$, tím specielně pro $i = n$ budeme hotovi. Je-li φ_i axiom logiky, je φ_i^* axiom stejného typu. Je-li φ_i formule z T, je v jazyce $\mathcal{L}$, takže φ_i^* je formule φ_i. Je-li φ_i instance nového axiomu, je φ_i^* formule tvaru $\delta' \leftrightarrow \delta$, která je dokazatelná podle věty o variantách. Je-li φ_i odvozena z nějakých předchozích φ_j, φ_k pomocí modus ponens, je φ_i^* odvozena z odpovídajících φ_j^*, φ_k^*, jak se snadno nahlédne. Je-li φ_i odvozena pravidlem generalizace z nějaké předchozí φ_j, tj. je-li φ_i tvaru $(\forall x)\varphi_j$, máme již $T \vdash \varphi_j^*$, přitom φ_i^* je formule $(\forall x)\varphi_j^*$, tedy je odvozena pravidlem generalizace z φ_j^*. □

Například jazyk aritmetiky běžně rozšiřujeme o nový binární predikát | pro dělitelnost a přidáváme definující axiom $x|y \leftrightarrow (\exists d)(y = x*d)$, podobně unární predikát p pro prvočíselnost a axiom $p(x) \leftrightarrow (\forall y)(\forall z)(p = x * y \to p = x \vee p = y)$. Podle právě dokázané věty jsou taková rozšíření konzervativní: obohacují

výrazové prostředky teorie, ale nepřidávají dokazatelné formule původního jazyka.

2.5.2 Věta (o nové funkci). *Buď φ formule s volnými proměnnými $x_1, \ldots, x_n, y$, a buď T teorie, která dokazuje formuli $(\forall x_1) \ldots (\forall x_n)(\exists y)\varphi$. Potom teorie T' v jazyce s novým n-árním funkčním symbolem f, která obsahuje nový axiom $\varphi_y[f(x_1, \ldots, x_n)]$, je konzervativním rozšířením T.*

Skolemizace Obrat použitý v předchozí větě lze použít k eliminaci existenčních kvantifikátorů, za cenu rozšíření jazyka o nové *Skolemovské funkce*.

2.5.3 Definice. Pro formuli φ v prenexním tvaru $(Qx_0) \ldots (Qx_n)\psi$ buď φ_S formule, která vznikne z φ vynecháním všech existenčních kvantifikátorů $\exists x_i$ z prefixu $(Qx_0) \ldots (Qx_n)$ a nahrazením (volných) výskytů proměnné x_i v otevřeném jádře ψ termem $f_i(x_0, \ldots, x_{i-1})$, kde f je nový funkční symbol četnosti i, pokud existenční kvantifikaci $\exists x_i$ předchází právě i univerzálních kvantifikátorů $\forall x_j, j < i$. Formule φ_S je pak *Skolemovská varianta* formule φ.

Pro otevřenou formuli či prenexní formuli φ bez existenčních kvantifikátorů je φ_S formule φ sama. Případných existenčních kvantifikátorů je v prefixu konečně mnoho, takže ke Skolemovskému tvaru lze dojít v konečně mnoha krocích.

2.5.4 Lemma. *Formule φ je splnitelná právě když φ_S je splnitelná.*

Důkaz. Pomocí 2.3.3 se snadno ukáže $\vdash \varphi_s \to \varphi$. V opačném směru buď $\mathfrak{M} \models \varphi$, máme najít model φ_S. Pro jednoduchost předpokládejme, že φ obsahuje jen jeden existenční kvantifikátor (v opačném případě pokračujeme indukcí). Je tedy $\mathfrak{M} \models (\forall x_1) \ldots (\forall x_n)(\exists y)\psi$, kde ψ už další existenční kvantifikace neobsahuje. Odpovídající Skolemovskou funkci f potom realizujeme přirozeným způsobem: v modelu $\mathfrak{M}$ existuje ke každým $a_1, \ldots, a_n \in M$ nějaké $b \in M$ tak, že $\mathfrak{M} \models \psi[a_1, \ldots, a_n, b]$; stačí tedy vzít za $f^{\mathfrak{M}}$ nějakou výběrovou funkci. Vzniklá struktura je pak modelem φ_S. □

Například: jelikož existuje nekonečně mnoho prvočísel, platí v $\mathbb{N}$ formule $(\forall x)(\exists y)(x < y \wedge p(y))$. Strukturu $\mathbb{N}$ tedy můžeme rozšířit o novou unární funkci $f : \mathbb{N} \to \mathbb{N}$, která každému číslu $n \in \mathbb{N}$ přiřazuje nějaké prvočíslo $f(n) > n$. Vzniklá struktura potom splňuje $(\forall x)(x < f(x) \wedge p(f(x)))$.

2.5.5 Věta (Skolem). *Každá teorie má konzervativní otevřené rozšíření.*

Důkaz. Je-li T_0 daná teorie, buď nejprve T_1 teorie v tomtéž jazyce, která jako axiomy obsahuje uzávěry prenexních tvarů formulí z T_0. Potom T_1 je ekvivalentní s T_0. Buď dále T_2 teorie, jejíž axiomy jsou Skolemovské varianty formulí z T_1. Podle předchozího je T_2 konzervativním rozšířením T_1. Buď konečně T_3 teorie, která jako axiomy obsahuje otevřená jádra formulí z T_2. Podle věty o uzávěru je T_3 ekvivalentní s T_2. Teorie T_3 je tedy hledané rozšíření. □

Například teorie pologrup jazyce s jediným binárním funkčním symbolem $*$ má jediný axiom $(\forall x)(\forall y)(\forall z)((x * y) * z = x * (y * z))$, nebo ekvivalentně jen $(x * y) * z = x * (y * z)$. Tedy teorie pologrup je otevřená teorie. V tomtéž jazyce lze formulovat teorii monoidů, pokud přidáme formuli $(\exists y)(\forall x)(y * x = x)$. Za cenu nové konstanty (třeba **1**) můžeme tuto formuli nahradit její Skolemovskou

variantou $(\forall x)(\mathbf{1} * x = x)$, resp. jejím otevřeným jádrem $\mathbf{1} * x = x$. V jazyce $\{*, \mathbf{1}\}$ pak můžeme zformulovat teorii grup přidáním axiomu $(\forall x)(\exists y)(x * y = \mathbf{1})$, a po přidání nové unární skolemovské funkce $^{-1}$ máme $(\forall x)(x * x^{-1} = \mathbf{1})$, resp. $x * x^{-1} = \mathbf{1}$. V jazyce $\{*, ^{-1}, \mathbf{1}\}$ můžeme konečně teorii grup zachytit obvyklou otevřenou axiomatikou $(x * y) * z = x * (y * z), \mathbf{1} * x = x, x * x^{-1} = \mathbf{1}$.

2.6 Rezoluční metoda

V kapitole o výrokové logice jsme popsali rezoluční metodu, která efektivně rozhoduje o splnitelnosti konečných výrokových teorií. Zde popíšeme zobecnění rezoluční metody pro jazyk predikátové logiky.[17]

Rezoluční kalkul je určitou formou predikátové logiky, ale liší se motivací i technickým přístupem: místo zkoumání *pravdivosti* a hledání *důkazu* ověřuje *nesplnitelnost* teorií a hledá jejich *zamítnutí*. Je-li T nějaká teorie a φ uzavřená formule, pak podle 2.3.14 je $T \vdash \varphi$ právě když teorie $T, \neg\varphi$ je sporná, což podle věty o úplnosti znamená právě tolik, že je nesplnitelná. Rezoluční metoda hledá místo důkazu formule φ v teorii T argument o nesplnitelnosti teorie $T, \neg\varphi$. Jednotlivé kroky takového zamítnutí přitom často postrádají přímočarost a průhlednost klasických důkazů, jsou však přístupné strojovému zpracování.

Nejprve přijmeme důležitá technická zjednodušení. Každá teorie má podle Skolemovy věty otevřené konzervativní rozšíření. Místo splnitelnosti dané teorie budeme tedy zkoumat splnitelnost tohoto rozšíření, v němž každá formule je otevřeným jádrem univerzální sentence v jazyce rozšířeném o nové Skolemovské konstanty a funkce. Navíc můžeme předpokládat, že toto jádro je v disjunktivní normální formě, tedy že je disjunkcí atomických formulí nebo jejich negací, které souhrně nazýváme *literály*. Na každou předloženou teorii můžeme tedy hledět jako na množinu klauzulí; na jejich pořadí ani na pořadí literálů v nich nezáleží.

2.6.1 Definice. Je-li S množina klauzulí, pak její *Herbrandovské univerzum* sestává ze všech konstantních termů jejího jazyka $\mathcal{L}$, tj. všech termů bez proměnných. Pokud jazyk $\mathcal{L}$ neobsahuje žádnou konstantu, je Herbrandovským univerzem množina konstantních termů jazyka $\mathcal{L} \cup \{a\}$, kde a je nová konstanta.

Například Herbrandovským univerzem teorie uspořádání je triviální množina $\{a\}$, neboť jazyk teorie uspořádání nemá žádné konstanty ani funkční symboly. Naopak Herbrandovské univerzum aritmetiky tvoří konstantní termy $0, 1, 0+0, 0+1, 1+0, 1+1, 0*0, 0*1, 1*0, 1*1, 0+(0+0), 0+(0+1), 0+(1+0), 0+(1+1), 0+(0*0)$, atd. Zřejmě pro teorii s alespoň jedním funkčním symbolem je Herbrandovské univerzum nekonečné.

2.6.2 Definice. Je-li S množina klauzulí a H její Hebrandovské univerzum, pak pro každou $P \subseteq H$ buď $P(S)$ *saturace* S *pomocí* P, totiž množina všech klauzulí bez proměnných, které vzniknou jako instance klauzulí z S dosazením konstantních termů z P za všechny proměnné.

[17] J. A. Robinson, *A Machine-Oriented Logic Based on the Resolution Principle*, Journal of the ACM, 12:1 (1965), 23–41

Rezoluce bez proměnných Klauzule z $H(S)$ neobsahují žádné proměnné, můžeme tedy na jejich literály hledět jako na prvotní (výrokové) formule a otázku splnitelnosti teorie $H(S)$ přenechat výrokové rezoluci. Vycházíme z následující podoby Herbrandovy věty, kterou uvádíme zatím bez důkazu.

2.6.3 Věta. *Buď S konečná množina klauzulí s Herbrandovským univerzem H. Je-li S nesplnitelná, pak už pro nějakou konečnou $P \subseteq H$ je $P(S)$ nesplnitelná.*

Spolu s větou o výrokové rezoluci potom máme

2.6.4 Důsledek. *Buď S konečná množina klauzulí s Herbrandovským univerzem H. Je-li S nesplnitelná, pak už pro nějakou konečnou $P \subseteq H$ a nějaké přirozené číslo n množina $R^n(P(S))$ obsahuje kontradikci.*

2.6.5 Příklad. Předvedeme důkaz elementárního tvrzení z teorie uspořádání: ostré uspořádání je antisymetrické. Standardní důkaz má dva řádky: kdyby v nějakém uspořádání $(X, <)$ existovaly prvky $a, b \in X$, pro které je zároveň $a < b$ i $b < a$, pak z transitivity je také $a < a$, což je ve sporu s antireflexivitou. Zajímá nás nyní, jaký argument dává výše popsaná metoda.

Formálně vzato chceme dokázat formuli $(\forall x)(\forall y)\neg(x < y \wedge y < x)$ v teorii $(\forall x)\neg(x < x), (\forall x)(\forall y)(\forall z)(x < y \wedge y < z \to x < z)$; ptáme se tedy, zda $(\forall x)\neg(x < x), (\forall x)(\forall y)(\forall z)(x < y \wedge y < z \to x < z), (\exists x)(\exists y)(x < y \wedge y < x)$ je splnitelná teorie. Její otevřenou Skolemovskou variantou je konečná množina klauzulí $S = \{\neg(x < x), \neg(x < y) \vee \neg(y < z) \vee (x < z), a < b, b < a\}$ v jazyce se dvěma novými Skolemovskými konstantami. Tato teorie nemá žádné funkční symboly, její Herbrandovské univerzum H tvoří jen konstanty a, b. Saturace $H(S)$ je potom následující konečná množina klauzulí bez proměnných:

$$
\begin{gathered}
(a < b), (b < a), \\
\neg(a < a), \neg(b < b), \\
\neg(a < a) \vee \neg(a < a) \vee (a < a), \neg(a < a) \vee \neg(a < b) \vee (a < b), \\
\neg(a < b) \vee \neg(b < a) \vee (a < a), \neg(a < b) \vee \neg(b < b) \vee (a < b), \\
\neg(b < a) \vee \neg(a < a) \vee (b < a), \neg(b < a) \vee \neg(a < b) \vee (b < b), \\
\neg(b < b) \vee \neg(b < a) \vee (b < a), \neg(b < b) \vee \neg(b < b) \vee (b < b)
\end{gathered}
$$

Po odstranění duplicit a tautologií, které splnitelnost neovlivní, zbyde

$$
\begin{gathered}
(a < b), (b < a), \neg(a < a), \neg(b < b), \\
\neg(a < b) \vee \neg(b < a) \vee (a < a), \neg(b < a) \vee \neg(a < b) \vee (b < b)
\end{gathered}
$$

Systematickou probírkou všech možných rezolvent, řekněme všechny dvojice klauzulí v pořadí zleva,[18] pak získáme následující přírůstky do $R^1(H(S))$:

$$
\begin{gathered}
\neg(b < a) \vee (a < a),\ \neg(b < a) \vee (b < b), \\
\neg(a < b) \vee (a < a),\ \neg(a < b) \vee (b < b), \\
\neg(a < b) \vee \neg(b < a)
\end{gathered}
$$

Do $R^2(H(S))$ pak přibudou $(a < a), (b < b), \neg(b < a), \neg(a < b)$, a $R^3(H(S))$ obsahuje kontradikci. Uvažovaná množina klauzulí tedy není splnitelná.

Posloupnost klauzulí použitých cestou ke kontradikci je příkladem *zamítnutí* formule $(\exists x)(\exists y)(x < y \wedge y < x)$ v teorii ostrého uspořádání (přesnou definici podáme později). Tento „strojový“ důkaz postrádá přímočarost „lidského“ důkazu, na druhou stranu se děje zcela mechanicky, nevyžaduje žádný vhled do struktury uspořádaných množin, a poskytuje stejný protipříklad.

[18] Způsob, jak přesně rezolventy hledat, je samostatná otázka, kterou prozkoumáme později.

Uvedený příklad je v několika ohledech speciální. Herbrandovské univerzum je dvouprvkové, obecně může být i v konečných případech neúnosně velké. Pokud jazyk obsahuje nějaký funkční symbol, je Herbrandovské univerzum nekonečné, a předem nevíme, která konečná část $P \subseteq H$, pokud vůbec nějaká, postačuje k zamítnutí. Pro formuli φ, která z dané teorie T neplyne, tak bude rezoluční metoda navždy marně hledat protipříklad na $T, \neg\varphi$.

Je-li Herbrandovské univerzum H spočetné, existuje posloupnost konečných množin $P_k \subseteq H$ tak, že $P_k \subseteq P_{k+1}$ a $H = \bigcup P_k$. Můžeme například za P_0 vzít konstanty, a indukcí přidávat do P_{k+1} konstantní termy vzniklé aplikací funkčních sybolů na termy z P_k. Nabízí se potom ověřit postupně splnitelnost konečných množin $P_k(S)$. Je-li totiž $P \subseteq H$ konečná, je $P \subseteq P_k$ pro nějaké k, takže je-li S nesplnitelná, musí už nějaká $P_k(S)$ být nesplnitelná. Růst množin P_k a odpovídajících saturací $P_k(S)$ je však kromě triviálních případů neúnosný. Ukazuje se, že efektivnější než saturovat klauzule z S pomocí hladin Hebrandova univerza a následně provádět výrokovou rezoluci je zobecnit samotnou rezoluci pro jazyk predikátové logiky, kdy klauzule mohou obsahovat proměnné.

Unifikace Atomické formule $((x*r)+1) < (y*z)$ a $\neg((p+q) < r*(s+1))$ jazyka aritmetiky nejsou navzájem opačnými literály. Vhodnými substitucemi lze ale vyrobit jejich instance, které opačnými literály jsou. Pokud za proměnnou p dosadíme term $x*y$, za proměnnou q konstantu 1, za proměnnou r proměnnou y, a za proměnnou z term $s+1$, vznikne formule $((x*y)+1) < (y*(s+1))$, která je instancí první i druhé formule, resp. její negace. Říkáme, že taková substituce obě formule *unifikuje*. Pokud dokážeme literály v klauzulích unifikovat, můžeme rezoluční metodu rozšířit na predikátovou logiku.

Již dříve jsme zavedli notaci $t_{x_1,\ldots,x_n}[t_1,\ldots,t_n]$ a $\varphi_{x_1,\ldots,x_n}[t_1,\ldots,t_n]$ pro instance termů a formulí. Termy a formule budeme pro účely tohoto oddílu nazývat souhrně *výrazy*. Potřebujeme se nyní detailně zabývat prováděnými substitucemi jako takovými. *Substituce* je tedy konečná množina párů x_i/t_i, kde x_i jsou navzájem různé proměnné a t_i jsou libovolné termy, přičemž t_i není proměnná x_i. Je-li potom e nějaký výraz (term nebo formule) a je-li $\sigma = \{x_1/t_1,\ldots,x_n/t_n\}$ nějaká substituce, budeme jako $e\sigma$ značit výraz, který vznikne současným nahrazením všech volných výskytů všech proměnných x_i odpovídajícími termy t_i. Například pro substituci $\sigma = \{p/(x*y), q/1, r/y, z/(s+1)\}$, je-li t je term $p+q$ a φ je formule $((x*r)+1) < (y*z)$, je $t\sigma$ term $(x*y)+1$ a $\varphi\sigma$ je instance $((x*y)+1) < (y*(s+1))$. Je-li $\mathcal{E}$ množina výrazů, položme $\mathcal{E}\sigma = \{e\sigma; e \in \mathcal{E}\}$.

Využijeme dvě speciální substituce: je-li e nějaký výraz, buď μ_e (resp. ν_e) substituce, která nahrazuje všechny (volné) proměnné ve výrazu e v abecedním pořadí[19] proměnnými $x_1, x_2, \ldots$ (resp. $y_1, y_2, \ldots$). Tyto substituce mají čistě technický význam, totiž odstranit případné kolize ve jménech proměnných.

[19]Symboly jazyka jsou předem nějak dobře uspořádány, například následovně. Proměnné předcházejí konstantám, následují unární funkční symboly, po nich binární, atd; následují unární predikátové symboly, po nich binární, atd; následují logické spojky; kvantifikátory se nás při práci s otevřenými Skolemovskými formulemi netýkají. V rámci jednotlivých tříd symbolů (proměnné, konstanty, ...) volíme buďto abecední nebo nějaké jiné (arbitrární) uspořádání. Například symboly jazyka aritmetiky jsou uspořádány v pořadí $a_1, a_2, \ldots, b_1, b_2, \ldots, y_1, y_2, \ldots, z_1, z_2, \ldots, 0, 1, S, *, +, <, =, \neg, \vee, \wedge$. Každé takové lineární uspořádání symbolů se přirozeným způsobem rozšiřuje na lineární uspořádání termů a formulí.

Například je-li ψ formule $((x*r)+1)<(r*(s+1))$, je $\psi\mu_\psi$ formule $((x_3*x_1)+1)<(x_1*(x_2+1))$ a $\psi\nu_\psi$ formule $((y_3*y_1)+1)<(y_1*(y_2+1))$. Někdy budeme psát stručněji $e\mu$ místo $e\mu_e$, i když pro výrazy e s různými proměnnými jsou μ_e různé substituce.

Jsou-li $\sigma=\{u_1/s_1,\ldots,u_m/s_m\}$ a $\tau=\{v_1/t_1,\ldots,v_n/t_n\}$ dvě substituce, definujeme jejich *složení* $\sigma\tau$ jako substituci $\sigma'\cup\tau'$, kde σ' tvoří všechny páry $u_i/s_i\tau$, pro které $s_i\tau$ je term různý od proměnné u_i, a τ' tvoří všechny páry v_j/t_j z τ, kde proměnná v_j není žádná z proměnných $u_1,\ldots,u_n$. Například substituce uvedená výše je složením substitucí $\{p/(x*r),q/1\}$ a $\{r/y,z/(s+1)\}$.

Snadno se ověří, že skládání substitucí je asociativní, prázdná substituce ϵ je vůči skládání neutrální, a pro každý výraz e a substituce σ,τ je $(e\sigma)\tau=e(\sigma\tau)$.

Je-li $\mathcal{E}$ množina výrazů, pak její *kolizní množinu* tvoří všechny podvýrazy (podtermy, podformule) výrazů z $\mathcal{E}$, které začínají na pozici,[20] na které v nějakém jiném výrazu z $\mathcal{E}$ stojí nějaký jiný podvýraz.

Řekneme, že substituce σ *unifikuje* množinu výrazů $\mathcal{E}$, pokud je $\mathcal{E}\sigma$ jednoprvková. Pokud k množině $\mathcal{E}$ existuje taková unifikující substituce, řekneme, že $\mathcal{E}$ je *unifikovatelná*. Zřejmě unifikovat $\mathcal{E}$ znamená unifikovat její kolizní množinu.

2.6.6 Definice (Unifikační algoritmus)**.** Vstupem následující procedury je jakákoli konečná množina výrazů $\mathcal{E}$. Výstupem je buďto unifikující substituce, pokud existuje, nebo informace o tom, že $\mathcal{E}$ není unifikovatelná.

(1) Polož $k=0,\sigma_0=\epsilon$ a jdi na (2).

(2) Je-li $\mathcal{E}\sigma_k$ jednoprvková, vrať σ_k a skonči.

(3) Buď v_k lexikograficky první a e_k první jemu odpovídající kolizní výraz množiny $\mathcal{E}\sigma_k$. Je-li v_k proměnná, která se nevyskytuje v e_k, buď $\sigma_{k+1}=\sigma_k\{v_k/e_k\}$, zvedni k o jedna, a pokračuj na (2). Jinak skonči neúspěchem.

2.6.7 Příklad. Unifikujeme následující aritmetické formule:

$$\begin{aligned}((x*r)+1)&<(y*z)\\(p+q)&<(r*(s+1))\end{aligned}$$

Term $(x*r)$ je kolizní, na téže pozici stojí ve druhé formuli proměnná p; podobně termy q a 1 jsou kolizní, stejně jako z a $(s+1)$. Začínáme tedy s kolizními výrazy

$$p,q,r,y,z,1,(x*r),(s+1).$$

Výraz p je proměnná, a nevyskytuje se v odpovídajícím kolizním termu $x*r$. Buď tedy $\sigma_1=\{p/(x*r)\}$. Množina $\mathcal{E}\sigma_1$ potom sestává z formulí

$$\begin{aligned}((x*r)+1)&<(y*z)\\((x*r)+q)&<(r*(s+1))\end{aligned}$$

[20] Při použití infixní notace je namístě určitá syntaktická opatrnost: ve formulích $((p*q)<r)$ a $p*q<s+1$ nezačínají termy r a $s+1$ na téže pozici, totiž r začíná na osmém zatímco $s+1$ na pátém symbolu. Můžeme trvat na důsledném závorkování infixních termů a formulí, nebo používat prefixní notaci, kde závorky nejsou potřeba a termy a formule začínají svým funkčním resp. predikátovým symbolem (ve formulích $<*pqr$ a $<*pq+s1$ pak stojí r a $+s1$ na stejném místě), nebo pracovat se syntaktickými stromy, kde termy a podformule tvoří podstromy. Věříme, že čtenáře tyto technikálie nezastaví, a používáme obvyklý infix.

a má kolizní termy $q, r, y, z, 1, s+1$. Lexikograficky první z nich je proměnná q, a nevyskytuje se v odpovídajícím kolizním termu 1, buď tedy $\sigma_2 = \sigma_1\{q/1\} = \{p/(x*r)\}\{q/1\} = \{p/(x*r), q/1\}$. Pak $\mathcal{E}\sigma_2$ obsahuje formule

$$((x*r)+1) < (y*z)$$
$$((x*r)+1) < (r*(s+1))$$

a má kolize $r, y, z, s+1$. Lexikograficky první je proměnná r, nevyskytuje se v odpovídajícím termu y, buď tedy $\sigma_3 = \sigma_2\{r/y\} = \{p/(x*r), q/1\}\{r/y\} = \{p/(x*y), q/1, r/y\}$. Množina $\mathcal{E}\sigma_3$ pak obsahuje formule

$$((x*y)+1) < (y*z)$$
$$((x*y)+1) < (y*(s+1)),$$

má jediný kolizní pár z a $(s+1)$, přitom proměnná z se nevyskytuje v $(s+1)$. Buď tedy konečně $\sigma_4 = \sigma_3\{z/(s+1)\} = \{p/(x*y), q/1, r/y\}\{z/(s+1)\} = \{p/(x*y), q/1, r/y, z/(s+1)\}$. Množina $\mathcal{E}\sigma_4$ pak obsahuje jedinou formuli

$$((x*y)+1) < (y*(s+1))$$

a substituce $\sigma = \sigma_4$ je hledaná unifikace.

2.6.8 Příklad. Aritmetické formule

$$x < y + z$$
$$y < z + (x + x)$$

nejsou unifikovatelné. Postupujeme-li podle unifikačního algoritmu, použijeme nejprve substituci $\{x/y\}$ a dostaneme

$$y < y + z$$
$$y < z + (y + y),$$

pak substituci rozšíříme na $\{x/y\}\{y/z\} = \{x/z, y/z\}$, máme

$$z < z + z$$
$$z < z + (z + z)$$

a končíme s chybou: proměnná z se *vyskytuje* v kolizním výrazu $(z+z)$. Ve skutečnosti jsme ukázali jen to, že k unifikaci nevede konkrétní použitý algoritmus. Podle věty o unifikaci níže to ale znamená, že neexistuje žádná unifikace.

Každá konečná množina klauzulí obsahuje jen konečně mnoho proměnných, takže unifikační algoritmus na takovém vstupu musí jednou skončit, neboť s každým dalším krokem má o jednu proměnnou méně. Ukážeme, že na unifikovatelném vstupu končí úspěchem, a nalezená unifikace je navíc nejobecnější možná.

2.6.9 Věta (o unifikaci)**.** *Buď $\mathcal{E}$ konečná unifikovatelná množina výrazů. Pak*

(i) *Unifikační algoritmus vrací substituci σ, která unifikuje $\mathcal{E}$.*

(ii) *Pro libovolnou jinou unifikaci τ existuje substituce λ tak, že $\tau = \sigma\lambda$.*

Důkaz. Ukážeme, že pro unifikovatelnou množinu $\mathcal{E}$ končí algoritmus úspěchem, a že pro každý krok $k \geq 0$ existuje substituce λ_k taková, že $\tau = \sigma_k\lambda_k$.

Pro $k = 0$ je $\sigma_0 = \epsilon$, takže stačí položit $\lambda_0 = \tau$. Pokud po k krocích známe substituce σ_k a λ_k takové, že $\tau = \sigma_k\lambda_k$, nastávají dva případy. Buďto je $\mathcal{E}\sigma_k$ jednoprvková, a končíme se $\sigma = \sigma_k$, nebo pokračujeme instrukcí (3).

Přitom λ_k unifikuje množinu $\mathcal{E}\sigma_k$, jelikož $(\mathcal{E}\sigma_k)\lambda_k = \mathcal{E}(\sigma_k\lambda_k) = \mathcal{E}\tau$. Jinými slovy, λ_k unifikuje kolize množiny $\mathcal{E}\sigma_k$, takže specielně $v_k\lambda_k = e_k\lambda_k$ pro v_k, e_k definované v instrukci (3). Výrazy z kolizní množiny nemohou všechny začínat[21] stejným symbolem (jinak by nebyly kolizní), a některý z nich musí být proměnná (jinak by $\mathcal{E}$ nebyla unifikovatelná). Tedy v_k je proměnná, neboť proměnné předcházejí ve zvoleném uspořádání všechny ostatní symboly. Pokud se proměnná v_k vyskytuje v odpovídajícím kolizním výrazu e_k, vyskytuje se také $v_k\lambda_k$ ve výrazu $e_k\lambda_k$, což pro dva různé výrazy splňující $v_k\lambda_k = e_k\lambda_k$ není možné. Proměnná v_k se tedy nevyskytuje ve výrazu e_k, a pokračujeme instrukcí (2) se substitucí $\sigma_{k+1} = \sigma_k\{v_k/e_k\}$. Stačí pak položit $\lambda_{k+1} = \lambda_k \setminus \{v_k/v_k\lambda_k\}$ a máme

$$\begin{aligned}
\lambda_k &= \{v_k/v_k\lambda_k\} \cup \lambda_{k+1} && \text{z definice } \lambda_{k+1}\\
&= \{v_k/e_k\lambda_k\} \cup \lambda_{k+1} && \text{díky } v_k\lambda_k = e_k\lambda_k\\
&= \{v_k/e_k\lambda_{k+1}\} \cup \lambda_{k+1} && \text{neboť } v_k \text{ se nevyskytuje v } e_k\\
&= \{v_k/e_k\}\lambda_{k+1} && \text{z definice složení}
\end{aligned}$$

Je tedy $\tau = \sigma_k\lambda_k = \sigma_k\{v_k/e_k\}\lambda_{k+1} = \sigma_{k+1}\lambda_{k+1}$. Tím je věta dokázána. □

Rezoluce pro predikátovou logiku Po zavedení unifikací můžeme rozšířit pojem rezolventy a rezoluce i na formule jazyka predikátové logiky. Takový přístup je obecnější než výroková rezoluce saturovaného Herbrandova univerza.

2.6.10 Definice. Buďte C, D klauzule a $L \subseteq C, M \subseteq D$ jejich neprázdné části. Buď dále N množina atomických podformulí z $L\mu \cup M\nu$. Je-li N unifikovatelná, s unifikací σ, a jsou-li $L\mu\sigma, M\nu\sigma$ jednoprvkové množiny obsahující navzájem opačné literály, je množina $(C \setminus L)\mu\sigma \cup (D \setminus M)\nu\sigma$ ***rezolventa*** klauzulí C a D.

Každé dvě klauzule mají zřejmě jen konečně mnoho rezolvent, a rezolventy C, D jsou právě rezolventy D, C, až na jména proměnných. Jsou-li navíc C, D bez proměnných, jsou jejich rezolventy právě výrokové rezolventy zavedené dříve.

Příklad 2.6.8 ukazuje, proč před unifikací nejprve standardizujeme jména proměnných. V kontextu rezoluční metody jsou klauzule otevřená jádra univerzálních sentencí, na jménech proměnných tedy nesejde, potřebujeme se jen zbavit případných umělých kolizí v těchto jménech. Po standardizaci jmen se z formulí $x < y + z$ a $y < z + (x + x)$ stanou $x_1 < x_2 + x_3$ a $y_2 < y_3 + (y_1 + y_1)$, které se snadno unifikují (dokonce druhá je instancí první).

Pro každou množinu klauzulí S označme jako $R(S)$ sjednocení S a množiny všech rezolvent klauzulí z S, a pro každé $n \in \mathbb{N}$ buď $R^{n+1}(S) = R(R^n(S))$. Pro S konečnou je i každá $R^n(S)$ konečná, ale rostoucí řetězec

$$S = R^0(S) \subseteq R^1(S) \subseteq \ldots \subseteq R^n(S) \subseteq R^{n+1}(S) \subseteq \ldots$$

[21] Upozorňujeme znovu, že výrazy jako $a < b$ a $x < y$ oba formálně „začínají“ symbolem $<$, a podobně pro ostatní predikáty a funkční symboly, které formálně píšeme $R(\ldots)$ a $f(\ldots)$.

se narozdíl od výrokové logiky nemusí stabilizovat ani pro konečnou S. Jednoduchým příkladem je $S = \{Q(a), \neg Q(x) \vee Q(f(x))\}$ v jazyce s konstantou a a unární funkcí f. Snadno se nahlédne, že mezi její rezolventy patří

$$Q(f(a)), Q(f(f(a))), Q(f(f(f(a)))), \ldots$$

takže rezoluční uzávěr je nekonečný.

Ukážeme předně, že rezoluce v jazyce predikátové logiky a následná saturace prvky Herbrandova univerza je obecnější než opačný postup, totiž saturace klauzulí konstantními termy a následná výroková rezoluce.

2.6.11 Věta. *Buď S libovolná množina klauzulí a buď P libovolná podmnožina jejího Herbrandova univerza. Potom $R(P(S)) \subseteq P(R(S))$.*

Důkaz. Buď $A \in R(P(S))$. Pokud je dokonce $A \in P(S)$, je také $A \in P(R(S))$, neboť $S \subseteq R(S)$. V opačném případě je A výrokovou rezolventou nějakých klauzulí $C\alpha, D\beta$ bez proměnných pro $C, D \in S$, $\alpha = \{v_1/t_1, \ldots, v_k/t_k\}$, kde v_i jsou právě všechny proměnné z C v abecedním pořadí, $\beta = \{w_1/u_1, \ldots, w_l/u_l\}$, kde w_i jsou právě všechny proměnné z D v abecedním pořadí, a $t_1, \ldots, t_k$ a $u_1, \ldots, u_l$ jsou konstantní termy z P. To jest, A je tvaru $(C \setminus L)\alpha \cup (D \setminus M)\beta$, kde $L \subseteq C, M \subseteq D$ jsou neprázdné, a $L\alpha, M\beta$ jsou jednoprvkové množiny obsahující navzájem opačný literál. V tom případě položme

$$\tau = \{x_1/t_1, \ldots, x_k/t_k, y_1/u_1, \ldots, y_l/u_l\}.$$

Pak bude $A = (C \setminus L)\mu\tau \cup (D \setminus M)\nu\tau$, $L\alpha = L\mu\tau$ a $M\beta = M\nu\tau$. To znamená, že τ unifikuje množinu N atomických podformulí z $L\mu \cup M\nu$. Existuje tedy unifikace σ a substituce λ termů z P tak, že $\tau = \sigma\lambda$. Je tedy $L\mu\sigma\lambda = M\nu\sigma\lambda$, takže $L\mu\sigma$ a $M\nu\sigma$ jsou jednoprvkové množiny obsahující navzájem opačný literál. To znamená, že $B = (C \setminus L)\mu\sigma \cup (D \setminus M)\nu\sigma$ je rezolventou $C, D \in S$, neboli $B \in R(S)$. Přitom $\tau = \sigma\lambda$, takže $A = B\lambda$ a máme $A \in P(R(S))$. □

Opačná inkluze nemusí platit ani v jednoduchých případech. Například pro množinu klauzulí $S = \{Q(x, f(y)), \neg Q(g(y), x)\}$ a $P = \{a\} \subseteq H$ je $P(S) = \{Q(a, f(a)), \neg Q(g(a), a)\} = R(P(S))$, zatímco $P(R(S))$ obsahuje kontradikci.

2.6.12 Důsledek. *Buď S libovolná množina klauzulí a buď P podmnožina jejího Herbrandova univerza. Potom $R^n(P(S)) \subseteq P(R^n(S))$ pro každé $n \in \mathbb{N}$.*

Z výše uvedeného ihned dostáváme následující verzi Herbrandovy věty.

2.6.13 Důsledek. *Buď S konečná množina klauzulí s Herbrandovským univerzem H. Je-li S nesplnitelná, pak už pro nějakou konečnou $P \subseteq H$ a nějaké přirozené číslo n množina $P(R^n(S))$ obsahuje kontradikci.*

Znění je stejné jako ve větě 2.6.4, jen pořadí rezoluce a saturace je opačné. Pokud si nyní uvědomíme, že pouhým dosazením konstantních termů do klauzulí nová kontradikce nevznikne, vidíme, že $P(R^n(S))$ obsahuje kontradikci právě když $R^n(S)$ obsahuje kontradikci. Odtud získáváme hlavní větu.

2.6.14 Věta (o rezoluci). *Buď S konečná množina klauzulí. Potom S je nesplnitelná, právě když pro nějaké $n \in \mathbb{N}$ množina $R^n(S)$ obsahuje kontradikci.*

2.6.15 Příklad. Množina klauzulí $\{Q(x,f(y)),\neg Q(g(y),x)\}$ zmíněná výše je nesplnitelná. Obě klauzule jsou jednoprvkové, do rezoluce tedy vstupují jen literály $Q(x,f(y))$ a $\neg Q(g(y),x)$, resp. $Q(x_1,f(x_2))$ a $\neg Q(g(y_2),y_1)$, po standardizaci jmen. Tyto literály potřebujeme nejprve unifikovat (až na případnou negaci). Lexikograficky prvním kolizním výrazem je proměnná x_1, která se nevyskytuje v odpovídajícím kolizním výrazu $g(y_2)$. Klademe tedy $\sigma_1 = \{x_1/g(y_2)\}$ a zbývá unifikovat $Q(g(y_2),f(x_2))$ s $\neg Q(g(y_2),y_1)$. Jedinou kolizí je nyní proměnná y_1 a výraz $f(x_2)$, ve kterém se y_1 nevyskytuje. Buď tedy $\sigma_2 = \sigma_1\{y_1/f(x_2)\} = \{x_1/g(y_2), y_1/f(x_2)\}$. Substituce σ_2 pak unifikuje literály $Q(x_1,f(x_2))$ a $\neg Q(g(y_2),y_1)$ na vzájemně opačné literály $Q(g(y_2),f(x_2))$ resp. $\neg Q(g(y_2),f(x_2))$. Jiné literály v použitých klauzulích nejsou, jejich rezolventa je tedy prázdná, a uvažovaná množina klauzulí je nesplnitelná.

Zároveň si můžeme opět všimnout, jak standardizace jmen umožňuje unifikaci. V původních literálech $Q(x,f(y))$ a $\neg Q(g(y),x))$ je třeba substituovat $\{x/g(y)\}$, po substituci však s výrazy $Q(g(y),f(y))$ a $\neg Q(g(y),g(y)))$ není jak pokračovat: kolizní term $f(y)$ žádná substituce nepřepíše na $g(y)$. Standardizace jmen využívá jen toho, že proměnné x,y ve formuli $(\forall x)(\forall y)Q(x,f(y))$ nemají nic společného s proměnnými x,y ve formuli $(\forall x)(\forall y)\neg Q(g(y),x)$.

2.6.16 Příklad. Ukážeme, jak rezoluční metoda dojde k následujícímu tvrzení z teorie pologrup: pologrupa s dělením má zprava neutrální prvek.

Připomeňme, že *pologrupa* je množina opatřená asociativní operací. Pologrupa $(X,*)$ má *dělení zleva*, pokud pro každé $x,y\in X$ existuje $a\in X$ takové, že $a*x=y$; podobně má *dělení zprava*, pokud pro každé $x,y\in X$ existuje $b\in X$ tak, že $x*b=y$. Pokud splňuje obě podmínky, jde o *pologrupu s dělením*.

Důkaz není těžký. Zvolme $x\in X$ libovolně. Potom díky dělení zprava existuje $f\in X$ takové, že $x*f=x$. Ukážeme, že prvek $f\in X$ je zprava neutrální. Pro $y\in X$ totiž díky dělení zleva existuje $a\in X$ tak, že $a*x=y$, načež $y*f=(a*x)*f=a*(x*f)=a*x=y$ a jsme hotovi.[22]

Zajímá nás nyní, jak k témuž závěru dojde rezoluční metoda, totiž jak odhalí kontradikci obsaženou v teorii $(\forall x)(\forall y)(\forall z)((x*y)*z = x*(y*z))$, $(\forall x)(\forall y)(\exists a)(a*x=y)$, $(\forall x)(\forall y)(\exists b)(x*b=y)$, $\neg(\exists x)(\forall y)(y*x=y)$.

Zjednodušíme předně jazyk, ve kterém formulujeme předpoklady i závěr. Ve výše uvedeném argumentu několikrát mlčky používáme axiomy rovnosti, které jsou tak samozřejmé, že je ani nepřipomínáme. Ve formálním důkaze či v chodu rezoluční metody bychom museli každou použitou instanci explicitně rozepsat. Použijeme místo toho jazyk s jediným ternárním predikátem Q; atomická formule $Q(x,y,z)$ má značit, že součinem x,y je z. Formulace v takovém jazyce působí na první pohled kostrbatě, ale díky tomu, že se jedná o jazyk bez rovnosti, je následující argument ve skutečnosti jednodušší.

Axiom asociativity pak v tomto jazyce nabývá podobu univerzální sentence $(\forall x)(\forall y)(\forall z)(\forall u)(\forall v)(\forall w)(Q(x,y,u)\to Q(y,z,v)\to Q(x,v,w)\to Q(u,z,w))$, otevřeným jádrem je klauzule $\neg Q(x,y,u)\vee\neg Q(y,z,v)\vee\neg Q(x,v,w)\vee Q(u,z,w)$ v normálním tvaru. Dělení zleva je po překladu formule $(\forall x)(\forall y)(\exists a)Q(a,x,y)$, jejímž otevřeným jádrem je po Skolemizaci klauzule $Q(g(x,y),x,y)$, a podobně klauzule $Q(x,h(x,y),y)$ vyjadřuje dělení zprava. Negací $(\exists x)(\forall y)(y*x=y)$ je

[22] Analogicky se najde zleva neutrální prvek $e\in X$. Z definice je potom $e=e*f=f$, takže $(X,*,e)$ je monoid. Pro libovolné $x\in X$ navíc máme $a\in X$ takové, že $a*x=e$, a zároveň $b\in X$ takové, že $x*b=e$, neboli levý a pravý inverz. Potom opět $a=a*e=a*(x*b)=(a*x)*b=e*b=b$. Pologrupa s dělením je tedy ve skutečnosti grupa.

konečně formule $(\forall x)(\exists y)(y{*}x \neq y)$ se Skolemovským jádrem $\neg Q(k(x), x, k(x))$.

Při značení z 2.6.10 volíme pro první krok rezoluce za klauzuli C axiom asociativity ve výše uvedené podobě, za $L \subseteq C$ množinu $\{\neg Q(x,y,u), \neg Q(x,v,w)\}$, a za $M = D$ jednoprvkovou $\{Q(g(x,y),x,y)\}$. Po standardizaci jmen[23] je tedy

$$\begin{aligned} L\mu &= \{\neg Q(x_4, x_5, x_1), \neg Q(x_4, x_2, x_3)\} \\ M\nu &= \{Q(g(y_1,y_2), y_1, y_2)\} \\ N &= \{Q(x_4, x_5, x_1), Q(x_4, x_2, x_3), Q(g(y_1,y_2), y_1, y_2)\} \end{aligned}$$

a hledáme unifikaci pro N podle unifikačního algoritmu. Lexikograficky prvním kolizním výrazem je proměnná x_1 s kolizí x_3, což je výraz, ve kterém se x_1 nevyskytuje. Klademe tedy $\sigma_1 = \{x_1/x_3\}$ a zbývá unifikovat množinu výrazů $N\sigma_1 = \{Q(x_4,x_5,x_3), Q(x_4,x_2,x_3), Q(g(y_1,y_2),y_1,y_2)\}$. Jejím prvním kolizním výrazem je proměnná x_2 s kolizí x_5, klademe tedy $\sigma_2 = \sigma_1\{x_2/x_5\} = \{x_1/x_3, x_2/x_5\}$ a máme $N\sigma_2 = \{Q(x_4,x_5,x_3), Q(g(y_1,y_2),y_1,y_2)\}$. Zde je první kolizí x_3/y_2, buď tedy $\sigma_3 = \sigma_2\{x_3/y_2\} = \{x_1/y_2, x_2/x_5, x_3/y_2\}$ a zbývá unifikovat $N\sigma_3 = \{Q(x_4,x_5,y_2), Q(g(y_1,y_2),y_1,y_2)\}$. Proměnná x_4 je v kolizi s výrazem $g(y_1,y_2)$, buď tedy $\sigma_4 = \sigma_3\{x_4/g(y_1,y_2)\} = \{x_1/y_2, x_2/x_5, x_3/y_2, x_4/g(y_1,y_2)\}$; zbývá unifikovat $N\sigma_4 = \{Q(g(y_1,y_2),x_5,y_2), Q(g(y_1,y_2),y_1,y_2)\}$. Jedinou kolizí je x_5/y_1, takže $\sigma = \sigma_5 = \sigma_4\{x_5/y_1\} = \{x_1/y_2, x_2/y_1, x_3/y_2, x_4/g(y_1,y_2), x_5/y_1\}$ unifikuje původní množinu N na jednoprvkovou $N\sigma = \{Q(g(y_1,y_2),y_1,y_2)\}$. Získáváme rezolventu $(C \setminus L)\mu\sigma \cup (D \setminus M)\nu\sigma = \{\neg Q(y_1,x_6,y_1), Q(y_2,x_6,y_2)\}$.

Zbylé rezoluční kroky zapíšeme již stručněji. Držíme se stále značení z 2.6.10, podtržením značíme kolizní výrazy a rezolventu. Standardizace μ, ν se týkají proměnných v celé klauzuli C resp. D (a správně bychom je měli značit μ_C, ν_D); viz podrobně např. jména proměnných v klauzuli $M\nu$ následujícího kroku.

$$\begin{aligned} L &= \{\neg Q(k(x),x,k(x))\} & C &= \{\neg Q(k(x),x,k(x))\} \\ M &= \{Q(y_2,x_6,y_2)\} & D &= \{\neg Q(y_1,x_6,y_1), Q(y_2,x_6,y_2)\} \\ L\mu &= \{\neg Q(k(x_1),x_1,k(x_1))\} & C\mu &= \{\neg Q(k(x_1),x_1,k(x_1))\} \\ M\nu &= \{Q(y_3,y_1,y_3)\} & D\nu &= \{\neg Q(y_2,y_1,y_2), Q(y_3,y_1,y_3)\} \\ N &= \{Q(k(x_1),\underline{x_1},k(x_1)), Q(y_3,\underline{y_1},y_3)\} & \sigma_1 &= \{x_1/y_1\} \\ N\sigma_1 &= \{Q(\underline{k(y_1)},y_1,k(y_1)), Q(\underline{y_3},y_1,y_3)\} & \sigma_2 &= \{x_1/y_1, y_3/k(y_1)\} \\ N\sigma_2 &= \{Q(k(y_1),y_1,k(y_1))\} & & \underline{\neg Q(y_2,y_1,y_2)} \end{aligned}$$

Ke kontradikci nyní zbývá poslední krok.

$$\begin{aligned} L &= \{Q(x,h(x,y),y)\} = C \\ M &= \{\neg Q(y_2,y_1,y_2)\} = D \\ L\mu &= \{Q(x_1,h(x_1,x_2),x_2)\} \\ M\nu &= \{\neg Q(y_2,y_1,y_2)\} \\ N &= \{Q(\underline{x_1},h(x_1,x_2),x_2), Q(\underline{y_2},y_1,y_2)\} & \sigma_1 &= \{x_1/y_2\} \\ N\sigma_1 &= \{Q(y_2,h(y_2,x_2),\underline{x_2}), Q(y_2,y_1,\underline{y_2})\} & \sigma_2 &= \{x_1/y_2, x_2/y_2\} \\ N\sigma_2 &= \{Q(y_2,\underline{h(y_2,y_2)},y_2), Q(y_2,\underline{y_1},y_2)\} & \sigma_3 &= \{x_1/y_2, x_2/y_2, y_1/h(y_2,y_2)\} \\ N\sigma_3 &= \{Q(y_2,h(y_2,y_2),y_2)\} \end{aligned}$$

[23] Přejmenování μ, ν se týká celých klauzulí C, D, i rezolventu obdržíme již v nových jménech.

Jednoprvkové klauzule C, D posledního kroku se unifikují na opačné literály. Jejich rezolventa je prázdná, uvažovaná teorie je nesplnitelná.

2.6.17 Cvičení. Ukažte, že následující dvě klauzule tvoří nesplnitelnou teorii.

$$Q(x, g(x), y, h(x, y), z, k(x, y, z))$$
$$\neg Q(u, v, e(v), w, f(v, w), x)$$

Všimněte si, že unifikovaná množina kolizí používá jedinou proměnnou, takže dosazením libovolného prvku Herbrandova univerza dostaneme i zamítnutí pomocí výrokové rezoluce. Použité termy se však vyskytují teprve na páté hladině $P_5 \subseteq H$ Herbrandova univerza, která má pro daný jazyk řádově 10^{64} prvků.

Rezoluční metoda jako logický systém

2.6.18 Definice. Buď S libovolná množina klauzulí. Potom *zamítnutí* S je každá konečná posloupnost klauzulí $\varphi_1, \ldots, \varphi_n$, ve které každá φ_i je buďto klauzule z S, nebo je rezolventou některých předchozích dvou, a φ_n je prázdná.

Jako důsledek věty o rezoluci ihned získáváme, že množina klauzulí S je nesplnitelná, právě když existuje její zamítnutí. V tomto smyslu je věta o rezoluci větou o úplnosti pro *rezoluční kalkul* jehož jediné odvozovací pravidlo je *pravidlo rezoluce*: z libovolných dvou klauzulí odvoď libovolnou jejich rezolventu.

Kapitola 3

Teorie množin

Dosud jsme používali základní množinové pojmy jako sjednocení, podmnožina, zobrazení, kartézský součin a další, aniž bychom se pozastavovali nad jejich „samozřejmým“ významem a existencí. Takový přístup je pro jistou část matematiky dostačující. Setkáme se však i s otázkami, na které lze odpovědět až po vyjasnění základních vlastností množin.

Ve skutečnosti samotná představa množiny jako souboru nějakých objektů s nějakou vlastností je sporná. To je obsahem známého Russelova paradoxu: označme jako r množinu všech těch množin, které neobsahují samy sebe jako prvek, formálně $r = \{x; x \notin x\}$, a ptejme se, zda je $r \in r$. Pokud je $r \in r$, pak má množina r onu vlastnost, kterou prvky množiny r mají, totiž $r \notin r$. Pokud naopak $r \notin r$, pak je z definice $r \in r$. Oba případy vedou ke sporu.

Podobné sémantické paradoxy vedly ke vzniku *axiomatické* teorie množin. Její jazyk obsahuje jediný speciální symbol, totiž binární predikát $\in$ pro náležení. Jak jsme naznačili již v úvodu, ukazuje se, že na tomto pojmu lze uvnitř univerza množin vybudovat základy veškeré běžné matematiky. Matematické objekty lze pak nahlížet jako množiny opatřené nějakou strukturou.

Například metrický prostor je uspořádaná dvojice (X, d), kde X je nějaká neprázdná množina a d je zobrazení z kartézského součinu $X \times X$ do množiny reálných čísel, splňující trojúhelníkovou nerovnost a další podmínky. Zbývá říct, *co je* neprázdná množina, uspořádaná dvojice, kartézský součin, zobrazení, uspořádání (pokud chceme mluvit o nerovnostech), a reálné číslo.

Většina matematických oborů postupně přijala množinový základ, a teorii množin tak připadla v moderní matematice podobná úloha, jakou v řecké matematice hrála geometrie: axiomy teorie množin popisují „svět matematiky“, tak jako axiomy geometrie popisují všeobjímající prostor.

Většině matematického provozu přitom stačí základní objekty a konstrukce, které axiomy teorie množin v tomto světě garantují, a jazyk teorie množin jako vyjadřovací prostředek. To je obsahem prvních několika odstavců.

Kromě této meta-matematické úlohy má teorie množin i svůj vlastní předmět studia, totiž aktuální nekonečno. Uvidíme, jak v teorii množin zavést pojem *mohutnosti* a popíšeme základy kombinatoriky nekonečných množin a počítání s nekonečnými mohutnostmi. Ukazuje se, že některé problémy v analýze, topologii, algebře, logice i jinde jsou ve skutečnosti otázkami po mohutnostech či kombinatorických vlastnostech nekonečných množin. Některé z nich jsou přitom tak jemné, že na ně ani sama teorie množin nemá jednoznačnou odpověď.

3.1 Axiomy teorie množin

Nejrozšířenější axiomatikou je Zermelo-Fraenkelova teorie množin, označovaná jako ZF. Její axiomy na jednu stranu zaručují, že univerzum množin je dostatečně bohaté, na druhou stranu vylučují existenci množin, které by vedly ke známým sporům obsaženým v naivní teorii množin. Uvedeme nejprve všechny axiomy. Ke každému jednotlivě se pak vrátíme a ukážeme, které z obvyklých množinových konstrukcí zaručuje.

Axiom extenzionality. Množiny se stejnými prvky se rovnají.

$$(\forall x)(\forall y)((\forall z)(z \in x \leftrightarrow z \in y) \rightarrow x = y)$$

Schema axiomů vydělení. Pro každou formuli $\varphi(z)$ jazyka teorie množin, která neobsahuje volně proměnnou y, je následující formule instancí axiomu vydělení: z každé množiny lze vydělit množinu prvků splňujících φ.

$$(\forall x)(\exists y)(\forall z)(z \in y \leftrightarrow (z \in x \wedge \varphi(z))$$

Axiom dvojice. Každé dvě množiny určují dvouprvkovou množinu.

$$(\forall x)(\forall y)(\exists d)(\forall z)(z \in d \leftrightarrow (x = z \vee y = z))$$

Axiom sumy. Každá množina má sjednocení.

$$(\forall x)(\exists s)(\forall z)(z \in s \leftrightarrow (\exists y)(y \in x \wedge z \in y))$$

Axiom potence. Každá množina má potenční množinu.

$$(\forall x)(\exists p)(\forall y)(y \in p \leftrightarrow y \subseteq x)$$

Schema axiomů nahrazení. Pro každou formuli $\varphi(u, v)$ jazyka teorie množin, která nemá volné proměnné w, y, je následující formule instancí axiomu nahrazení: obraz množiny při definovatelném zobrazení je množina.

$$\begin{aligned}(\forall u)(\forall v)(\forall w)(&\varphi(u,v) \wedge \varphi(u,w) \rightarrow v = w) \\ &\rightarrow (\forall x)(\exists y)(\forall v)(v \in y \leftrightarrow (\exists u)(u \in x \wedge \varphi(u,v))\end{aligned}$$

Axiom nekonečna. Existuje nekonečná množina.

$$(\exists x)(\emptyset \in x \wedge (\forall y)(y \in x \rightarrow y \cup \{y\} \in x))$$

Axiom regularity. V každé neprázdné množině existuje $\in$-minimální prvek.

$$x \neq \emptyset \rightarrow (\exists y)(y \in x \wedge y \cap x = \emptyset)$$

Axiomy jsou formulovány v jazyce, který kromě predikátu $\in$ používá další speciální symboly $\subseteq$, $\emptyset$, $\cup$, $\cap$, $\{x\}$ pro podmnožinu, prázdnou množinu, sjednocení, průnik, a jednoprvkovou množinu. Uvidíme, že tyto pojmy lze definovat již v základním jazyce teorie množin. Budeme také používat obvyklé zkratky $(\exists x \in y)\varphi$ za $(\exists x)(x \in y \wedge \varphi)$ a $(\forall x \in y)\varphi$ za $(\forall x)(x \in y \rightarrow \varphi)$.

Axiom extenzionality $(\forall x)(\forall y)((\forall z)(z \in x \leftrightarrow z \in y) \rightarrow x = y)$ zachycuje základní představu o množinách: množina je určena svými prvky, takže množiny se stejnými prvky jsou identické. Opačná implikace plyne z axiomů rovnosti, takže platí dokonce $(\forall x)(\forall y)((\forall z)(z \in x \leftrightarrow z \in y) \leftrightarrow x = y)$.

Schema axiomů vydělení je formálním protějškem intuitivní představy množiny jakožto souboru objektů s nějakou vlastností, respektive ústupkem z takové představy, která je podle Russelova paradoxu sporná: množinou není nutně *každý* takový soubor — jeho prvky musí pocházet z nějaké předem již dané množiny. Soubor, který není takto omezen, může být „příliš velký“ na to, aby byl množinou. Například $\{z; z = z\}$ není množina,[1] jinak aplikuj axiom vydělení pro formuli $z \notin z$ jako v Russelově paradoxu.

Schematem vydělení je pro každou množinu x a každou formuli $\varphi(z)$ zaručena existence množiny těch prvků $z \in x$, které splňují $\varphi(z)$. Tato množina je podle axiomu extenzionality jednoznačně určena, budeme ji značit $\{z; z \in x \wedge \varphi(z)\}$ nebo stručněji $\{z \in x; \varphi(z)\}$.

Formulí $z \neq z$ můžeme z jakékoli množiny x vydělit *prázdnou množinu* $\{z \in x; z \neq z\}$, která nemá žádné prvky. Taková množina je podle axiomu extenzionality jediná, budeme ji značit $\emptyset$.

Jsou-li a, b dvě množiny, pak formule $z \in b$ vyděluje z množiny a množinu $a \cap b = \{z; z \in a \wedge z \in b\}$, kterou nazýváme *průnik* množin a, b. Obecněji, pro neprázdnou množinu x je *průnik* $\bigcap x = \{z; (\forall y \in x)(z \in y)\}$ množina vydělená z libovolné $y \in x$. Množiny a, b jsou *disjunktní*, pokud je $a \cap b = \emptyset$. Formule $z \notin b$ vyděluje z množiny a *rozdíl* $a \setminus b = \{z \in a; z \notin b\}$ množin a, b.

Rozšířili jsme základní jazyk teorie množin o definice nových symbolů $\emptyset, \cap, \setminus$. Z kapitoly o predikátové logice víme, že se jedná o konzervativní rozšíření, a že každou formuli rozšířeného jazyka lze ekvivalentně zapsat již v základním jazyce. Například $x \cap y = z$ lze ekvivalentně zapsat jako $(\forall t)(t \in z \leftrightarrow (t \in x \wedge t \in y))$. V dalším budeme již bez komentáře volně používat nově zavedené symboly.

Je-li každý prvek množiny a prvkem množiny b, tj. $(\forall z)(z \in a \rightarrow z \in b)$, řekneme, že množina a je *podmnožinou* množiny b a píšeme $a \subseteq b$. Je-li $a \subseteq b$ a přitom $a \neq b$, píšeme obvykle $a \subset b$ a říkáme, že a je *vlastní podmnožinou* b. Vidíme, že všechny množiny vydělené z množiny x jsou jejími podmnožinami.

Snadno se nahlédne, že pro jakékoli množiny x, y, z platí $\emptyset \subseteq x$, $x \subseteq x$; $x \subseteq y \wedge y \subseteq z \rightarrow x \subseteq z$; $x \subseteq y \wedge y \subseteq x \leftrightarrow x = y$; $x \cap y = y \cap x$, $x \cap \emptyset = \emptyset$; $x \subseteq y \leftrightarrow x \cap y = x$; $x \subseteq y \leftrightarrow x \setminus y = \emptyset$; $x \cap y = \emptyset \leftrightarrow x \setminus y = x$.

Axiomy vydělení nejsou samy dostatečně silné na to, aby zaručily například existenci sjednocení dvou množin. K tomu a k dalším konstrukcím je potřeba přijmout další axiomy. Předpokládáme přitom, že čtenáři je většina těchto elementárních pojmů známa; naším záměrem není popisovat např. vlastnosti průniku, nýbrž ukázat, jak přijaté axiomy zaručují existenci a vlastnosti standardních množinových konstrukcí.

Podle *axiomu dvojice* $(\forall x)(\forall y)(\exists d)(\forall z)(z \in d \leftrightarrow (x = z \vee y = z))$ existuje ke každým dvěma množinám x, y další množina, jejímiž jedinými prvky jsou právě množiny x, y. Podle axiomu extenzionality je taková množina jednoznačně určena množinami x, y. Tím je definována operace $\{x, y\}$, kterou nazýváme *neu-*

[1] Takovéto „příliš velké“ soubory, tzv. *vlastní třídy*, nemají v ZF žádnou formální existenci, ale přesto s nimi lze rozumně pracovat. Zavedeme je v příštím oddíle jako užitečné zkratky.

spořádaná dvojice množin x, y. Jsou-li x a y jedna a tatáž množina, píšeme místo $\{x, x\}$ krátce $\{x\}$, a jednoprvkovou množinu $\{x\}$ nazýváme *singleton*.

Pomocí pojmu neuspořádané dvojice lze zavést *uspořádanou dvojici* množin x, y jako $(x, y) = \{\{x\}, \{x, y\}\}$. To je dvouprvková množina zaručená axiomem dvojice, přičemž prvek $\{x\}$ vyznačuje, která z obou množin x, y je první.

3.1.1 Lemma. *Pro každé množiny* x, y, u, v *je* (i) $\{x, y\} = \{u, v\}$ ***právě když*** $(x = u \wedge y = v) \vee (x = v \wedge y = u)$*;* (ii) $(x, y) = (u, v)$ ***právě když*** $(x = u \wedge y = v)$.

Obecněji pak můžeme zavést *uspořádanou* n*-tici* množin $x_1, \ldots, x_n$. Položme $(x_1) = x_1$, a je-li již definována n-tice $(x_1, \ldots, x_n)$, buď $(x_1, \ldots, x_n, x_{n+1})$ uspořádaná dvojice $((x_1, \ldots, x_n), x_{n+1})$. Pro dvojici x_1, x_2 se jedná o uspořádanou dvojici zavedenou výše. Pro množiny $x_1, \ldots, x_n, y_1, \ldots, y_n$ pak platí $(x_1, \ldots, x_n) = (y_1, \ldots, y_n) \leftrightarrow x_1 = y_1 \wedge \cdots \wedge x_n = y_n$.

Podle *axiomu sumy* $(\forall x)(\exists s)(\forall z)(z \in s \leftrightarrow (\exists y)(y \in x \wedge z \in y))$ existuje ke každé množině x její *sjednocení*, tj. taková množina, jejímiž prvky jsou právě prvky prvků množiny x. Podle axiomu extenzionality je suma jednoznačně určena množinou x. Tím je definována operace $\bigcup x = \{z; (\exists y)(y \in x \wedge z \in y)\}$. V případě $x = \{a, b\}$ pak sjednocení $\bigcup\{a, b\} = \{z; z \in a \vee z \in b\}$ značíme $a \cup b$.

Pomocí axiomu dvojice jsme definovali jednoprvkové a dvouprvkové množiny. Pomocí operace sjednocení můžeme nyní pro množiny a, b, c definovat tříprvkovou množinu $\{a, b, c\}$ jako $\{a, b\} \cup \{c\}$, a je-li pro nějaké $a_1, \ldots, a_n$ již definována $\{a_1, \ldots, a_n\}$, položíme $\{a_1, \ldots, a_n, a_{n+1}\} = \{a_1, \ldots, a_n\} \cup \{a_{n+1}\}$.

Pomocí množinového rozdílu a sjednocení definujeme pro množiny a, b jejich *symetrickou diferenci* jako $a \triangle b = (a \setminus b) \cup (b \setminus a)$.

Podle *axiomu potence* $(\forall x)(\exists p)(\forall y)(y \in p \leftrightarrow y \subseteq x)$ existuje ke každé množině x její *potenční množina*, tj. množina všech jejích podmnožin. Podle axiomu extenzionality je potenční množina jednoznačně učena množinou x. Tím je definována operace $P(x) = \{y; y \subseteq x\}$.

Jsou-li a, b množiny, je $a \times b = \{(x, y); x \in a, y \in b\}$ jejich *kartézský součin*. To je množina, neboť $a \times b = \{u; (\exists x \in a)(\exists y \in b) u = (x, y)\} \subseteq P(P(a \cup b))$. Obecně pak můžeme definovat $a_1 \times \cdots \times a_{n-1} \times a_n = (a_1 \times \cdots \times a_{n-1}) \times a_n$ jako množinu $\{(x_1, \ldots, x_n); x_1 \in a_1, \ldots, x_n \in a_n\}$ uspořádaných n-tic; speciálně $a^n = a \times \cdots \times a$ (n-krát) je *mocnina* množiny a.

Axiom nekonečna $(\exists x)(\emptyset \in x \wedge (\forall y)(y \in x \rightarrow y \cup \{y\} \in x))$ postuluje existenci nekonečné množiny. Pojem *nekonečna* resp. *konečnosti* obvykle používá nějaký pojem *čísla* nebo *mohutnosti*, které jsme dosud nezavedli.[2] Uvedený axiom žádný pojem mohutnosti nepoužívá; množiny s uvedenou vlastností se nazývají *induktivní*. Uvidíme později, že nejmenší induktivní množinou je množina přirozených čísel.

Všimněme si, že axiom nekonečna je první, který zaručuje existenci vůbec nějaké množiny. Různé alternativní axiomatiky, které axiom nekonečna neobsahují,[3] proto obvykle přijímají ještě *axiom existence* $(\exists x)(x = x)$.

Schema axiomů nahrazení zaručuje, že obrazem množiny při definovatelném zobrazení je opět množina. Je-li totiž $\varphi(u, v)$ formule taková, že

$$(\forall u)(\forall v)(\forall w)(\varphi(u, v) \wedge \varphi(u, w) \rightarrow v = w),$$

[2] Pojem konečné množiny lze zavést i bez pojmu mohutnosti, i když taková definice může působit uměle. Ve cvičeních ukážeme některé ekvivalentní definice konečnosti.

[3] Například teorie konečných množin ZF_{Fin}, která obsahuje *negaci* axiomu nekonečna.

existuje ke každé množině u nanejvýš jedna množina v tak, že $\varphi(u,v)$; tím je množina v množině u jednoznačně přiřazena. Axiom potom zaručuje, že pokud prvky $u \in x$ nahradíme odpovídajícími v, výsledkem bude opět množina.

Zatímco předchozí axiomy zaručují existenci rozličných množin a množinových operací, *axiom regularity* množinové univerzum omezuje. V každé neprázdné množině požaduje existenci $\in$-minimálního prvku, tj. takového $y \in x$, že pro žádné další $z \in x$ není $z \in y$. Tím vylučuje například množiny tvaru $x = \{x\}$, nebo obecněji případy $x \in x$ a cykly náležení jako $x_1 \in x_2 \in x_3 \in x_1$. Uvidíme později, že množinové univerzum splňující axiom regularity lze vystavět „odspoda“ z prázdné množiny iterováním operací potence a sjednocení.

Rozšířením teorie ZF o *axiom výběru* vznikne teorie ZFC, nejrozšířenější axiomatika teorie množin. Axiomu výběru se budeme věnovat v oddíle 3.7.

3.1.2 Cvičení. Axiomy jsme zformulovali v jazyce, který kromě predikátu $\in$ obsahuje další symboly (které jsme nicméně postupně definovali). Napište axiomy v základním jazyce teorie množin.

3.1.3 Cvičení. Díky axiomu vydělení je možné ostatní axiomy zeslabit a místo existence té které množiny požadovat jen množinu, která ji obsahuje jako část. Například axiom dvojice lze nahradit formulí $(\forall x)(\forall y)(\exists d)(x \in d \wedge y \in d)$ a samotnou dvojici $\{x,y\}$ pak z množiny d vydělit formulí $z = x \vee z = y$. Ukažte, že podobně je možné rozvolnit ostatní axiomy.

3.1.4 Cvičení. Ve schematech axiomů vydělení a nahrazení není možné vynechat podmínku o volných výskytech proměnných.

3.1.5 Cvičení. Ze schematu axiomů vydělení plyne silnější schema axiomů vydělení s parametrem: pro každou formuli $\varphi(z,p)$, která neobsahuje volně proměnnou y, je formule $(\forall x)(\forall p)(\exists y)(\forall z)(z \in y \leftrightarrow (z \in x \wedge \varphi(z,p))$ axiomem. Takové schema lze pak rozšířit i na formule s více parametry.

3.1.6 Cvičení. Každá instance axiomu vydělení plyne z nějaké instance axiomu nahrazení. (Návod: vhodnou formulí zobrazte vyhovující prvky na sebe.)

3.1.7 Cvičení. Určete prvky množin $\bigcup x$, $\bigcap x$, $\bigcup\bigcup x$, $\bigcap\bigcap x$, $\bigcup\bigcap x$, $\bigcap\bigcup x$, kde (a) $x = \{\{a,c\},\{b,c\},\{b,d\}\}$; (b) $x = \{\{\{a,b\},\{b\}\},\{\{b,c\}\}\}$.

3.2 Třídy

Jediné objekty, které v teorii množin existují, jsou množiny. Daná množina může být prvkem v jiných množinách, a naopak jediné její prvky jsou další množiny. Viděli jsme, že ne každý soubor tvaru $\{x;\varphi(x)\}$ je množinou. Z praktických důvodů je však výhodné i takové soubory používat.

3.2.1 Definice. Je-li $\varphi(x)$ formule jazyka teorie množin s volnou proměnnou x, řekneme, že $C = \{x;\varphi(x)\}$ je *definovatelná třída*. Obecněji, je-li $\varphi(x,p)$ formule se dvěma volnými proměnnými, je třída $C = \{x;\varphi(x,p)\}$ *definovatelná z* p. *Vlastní třída* je taková, která není množinou.[4]

[4]Tvrzení „$\{x;\varphi(x)\}$ je vlastní třída“ lze vyjádřit formulí $\neg(\exists u)(\forall x)(x \in u \leftrightarrow \varphi(x))$.

Třídy jsou zkratkami za odpovídající formule. Jsou-li x, y množiny, a jsou-li C, D třídy definované formulemi φ resp. ψ, jsou následující výrazy zkratkami za formule jazyka teorie množin.

$$\begin{array}{rll} x \in C & \text{je zkratka za} & \varphi(x) \\ y = C & \text{je zkratka za} & (\forall x)(x \in y \leftrightarrow \varphi(x)) \\ C = D & \text{je zkratka za} & (\forall x)(\varphi(x) \leftrightarrow \psi(x)) \\ C \subseteq D & \text{je zkratka za} & (\forall x)(\varphi(x) \rightarrow \psi(x)) \end{array}$$

Indukcí podle složitosti se snadno ověří, že každou formuli, která obsahuje výše uvedené výrazy, lze ekvivalentně rozepsat v základním jazyce teorie množin. Přitom pracovat se třídami je praktičtější než pracovat s formulemi. Podstatné je, že nedovolujeme *kvantifikaci* třídových proměnných. To by znamenalo kvantifikovat formule; takto zůstáváme v jazyce prvního řádu.

Pro třídy můžeme zavést operace analogické základním množinovým operacím. Jsou-li C, D třídy definované formulemi φ resp. ψ, buď

$$\begin{aligned} C \cup D &= \{x; \varphi(x) \vee \psi(x)\}\,; \\ C \cap D &= \{x; \varphi(x) \wedge \psi(x)\}\,; \\ C \setminus D &= \{x; \varphi(x) \wedge \neg\psi(x)\}\,. \end{aligned}$$

Každá množina $y = \{x; x \in y\}$ je třídou, ale nikoli naopak; zatím známe jednu vlastní třídu, totiž *univerzální třídu* neboli *univerzum* $V = \{x; x = x\}$. Je-li C třída a y množina, je $y \cap C$ množina: je-li totiž $\varphi(x)$ formule definující třídu $C = \{x; \varphi(x)\}$, je $y \cap C = \{x \in y; \varphi(x)\}$ množina podle axiomu vydělení.

Existují teorie množin, které vycházejí ze základního pojmu třídy, a množiny definují jako speciální typ tříd. V Zermelo-Fraenkelově teorii množin jsou třídy jen pomocným vyjadřovacím prostředkem, nejsou to objekty teorie.

3.3 Relace a funkce

3.3.1 Definice. Množina $R \subseteq X_1 \times \cdots \times X_n$ uspořádaných n-tic je *n-ární relace*. V případě $R \subseteq X^n$ je R relace *na množině* X.

Relace umožňují zachytit vztahy mezi objekty: $x_1, \ldots, x_n$ jsou v uvažovaném vztahu, pokud $(x_1, \ldots, x_n) \in R$. Volbou různých relací lze pak zkoumat různé vztahy. Pro *binární* relaci $R \subseteq X \times X$ píšeme obvykle xRy místo $(x, y) \in R$.

Pro zachycení vztahů mezi objekty tedy není potřeba zavádět nový typ objektů: relace jsou opět množiny. Z pohledu predikátové logiky je výstavba matematiky na množinovém základě způsob, jak formulovat vše podstatné v jazyce prvního řádu. Kvantifikace objektů, tj. množin, umožňuje zároveň kvantifikaci množin objektů, vlastností a vztahů mezi objekty.

3.3.2 Definice. Buď $R \subseteq X \times Y$. Potom množina $dom(R) = \{x; (\exists y)xRy\}$ je *definiční obor* relace R a $rng(R) = \{y; (\exists x)xRy\}$ její *obor hodnot*. Pro množinu $A \subseteq X$ je $R[A] = \{y \in Y; (\exists x \in A)xRy\}$ *obraz* množiny A při relaci R a pro množinu $B \subseteq Y$ je $R^{-1}[B] = \{x \in X; (\exists y \in A)xRy\}$ *vzor* množiny B. Relace $R^{-1} = \{(y, x); (x, y) \in R\} \subseteq Y \times X$ je relace *inverzní* k R. Jsou-li R, S relace, je $S \circ R = \{(x, z); (\exists y)(xRy \wedge ySz)\}$ *složení* relací R a S.

Obory $dom(R)$ a $rng(R)$ jsou skutečně množinami, neboť $dom(R) \subseteq \bigcup\bigcup R$ a $rng(R) \subseteq \bigcup\bigcup R$. Přímo z definice je $(R^{-1})^{-1} = R$, $dom(R) = rng(R^{-1})$, $rng(R) = dom(R^{-1})$. Pro složené relace je $dom(S \circ R) \subseteq dom(R)$, $rng(S \circ R) \subseteq S$ a $(S \circ R)^{-1} = R^{-1} \circ S^{-1}$.

3.3.3 Definice. Relace $f \subseteq X \times Y$ je ***funkce z*** X ***do*** Y, pokud pro každé $x \in X$ existuje nejvýše jedno $y \in Y$ takové, že $(x, y) \in f$. V takovém případě píšeme obvykle $f : X \to Y$ a $f(x) = y$ a říkáme, že $y \in Y$ je ***funkční hodnota*** či ***obraz*** $x \in X$, a x je ***vzorem*** y. Množinu všech funkcí z X do Y pak značíme jako Y^X.

Množina $dom(f) = \{x \in X; (\exists y \in Y) f(x) = y\}$ je ***definiční obor*** a množina $rng(f) = \{y \in Y; (\exists x \in X) f(x) = y\}$ je ***obor hodnot*** funkce $f : X \to Y$. Pro $A \subseteq X$ je $f \restriction A = \{(x, y) \in f; x \in A\}$ ***zúžení*** funkce f na množinu A. Funkce $f : X^n \to Y$ je n***-ární***, specielně $f : X^n \to X$ je n***-ární operace*** na X.

Někdy se místo ***funkce*** říká ***zobrazení***, a pojem ***funkce*** se vyhrazuje pro speciální případ zobrazení z $\mathbb{R}$ do $\mathbb{R}$, tedy pro ***reálné*** funkce, nebo alespoň funkce s reálnými hodnotami. Pro nás však budou oba pojmy synonymní. Soubor Y^X všech funkcí z X do Y je množina, neboť $Y^X \subseteq P(X \times Y)$.

3.3.4 Definice. Je-li $f : X \to Y$ funkce, pak pro $A \subseteq X$ je $f[A] = \{f(a); a \in A\}$ ***obraz množiny*** A, a pro $B \subseteq Y$ je $f^{-1}[B] = \{a \in A; f(a) \in B\}$ ***vzor*** množiny B.

Pokud pro $x_1, x_2 \in X$ různé jsou také $f(x_1), f(x_2) \in Y$ různé, je f ***prostá***. To je právě když i $f^{-1} : Y \to X$ funkce, totiž ***inverzní funkce*** k funkci f.

Je-li $f[X] = Y$, řekneme, že f je zobrazení z X ***na*** Y. Prostá funkce z X na Y je ***bijekce*** neboli ***vzájemně jednoznačné zobrazení*** mezi X a Y. V takovém případě píšeme někdy $f : X \approx Y$. Bijekce z X na X je ***permutace*** množiny X.

3.3.5 Definice. Jsou-li $f \in Y^X$ a $g \in Z^Y$ funkce, pro které $rng(f) \subseteq dom(g)$, buď $g \circ f = \{(x, z); (\exists y \in Y)(f(x) = y \wedge g(y) = z\} \subseteq X \times Z$ funkce z X do Z ***složená*** z funkcí f a g (v tomto pořadí).

Snadno se ověří, že složení funkcí je opět funkce. Značení $g \circ f$ je jedna ze dvou možných konvencí. Má tu výhodu, že odpovídá značení „po argumentech," totiž obrazem x při funkci $g \circ f$ je $g(f(x))$.

3.3.6 Definice. Je-li $\{X_i; i \in I\}$ soubor množin, pak ***kartézský součin*** $\prod_{i \in I} X_i$ tohoto souboru je systém všech funkcí $f : I \to \bigcup X_i$ splňujících $f(i) \in X_i$ pro každé $i \in I$. Je-li specielně $X_i = X$ pro každé $i \in I$, pak součin $\prod_{i \in I} X = X^I$ je ***mocnina*** množiny X. Prvky $f \in X^I$, tedy funkce $f : I \to X$, obvykle zapisujeme jako $(f_i \in X; i \in I)$ nebo stručněji $(f_i; i \in I)$ či $(f_i)_{i \in I}$.

Definice kartézského součinu je kompatibilní s představou konečného součinu $X_1 \times \cdots \times X_n$ jako množiny všech uspořádaných n-tic $(x_1, \ldots, x_n)$, kde $x_i \in X_i$. Vskutku, prvky právě zavedeného součinu $\prod_{i \in I} X_i$ jsou právě takové I-tice.

3.3.7 Cvičení. Ukažte, že pro každé zobrazení $f : X \to Y$ a systém množin $\mathcal{A} \subseteq P(Y)$ je $f^{-1}[\bigcup \mathcal{A}] = \bigcup \{f^{-1}[A]; A \in \mathcal{A}\}$ a $f^{-1}[\bigcap \mathcal{A}] = \bigcap \{f^{-1}[A]; A \in \mathcal{A}\}$; jinými slovy, vzor sjednocení je sjednocení vzorů, a vzor průniku je průnik vzorů. Podobně pro systém $\mathcal{A} \subseteq P(X)$ je $f[\bigcup \mathcal{A}] = \bigcup \{f[A]; A \in \mathcal{A}\}$, tedy obraz sjednocení je sjednocením obrazů. Ukažte na příkladě, že obraz průniku obecně nemusí být průnikem obrazů, a to ani v konečném případě.

3.3.8 Cvičení. (i) Jsou-li f, g funkce, pak také $f \cap g$ a $f \setminus g$ jsou funkce, a $f \cup g$ je funkce právě tehdy, když $f \restriction (dom(f) \cap dom(g)) = g \restriction (dom(f) \cap dom(g))$. (ii) Buď $\mathcal{F}$ taková množina funkcí, že pro každé $f, g \in \mathcal{F}$ je buďto $f \subseteq g$ nebo $g \subseteq f$. Potom také $\bigcup \mathcal{F}$ je funkce. (iii) Obecněji, pro systém funkcí $\mathcal{F}$, který je direktně uspořádán inkluzí, je také $\bigcup \mathcal{F}$ je funkce.

3.3.9 Cvičení. Složení prostých funkcí je prosté a složení bijekcí je bijekce. Inverzní funkce k bijekci je opět bijekce. Složení bijekce a jejího inverzu je identita. Systém všech permutací dané množiny tvoří grupu vůči skládání. Systém všech funkcí z dané množiny *na* sebe tvoří monoid.

3.3.10 Cvičení. (i) Funkce $h : Y \to Z$ je prostá právě tehdy, když pro každé dvě funkce $f, g : X \to Y$ platí $h \circ f = h \circ g \to f = g$. (ii) Funkce $f : X \to Y$ je na Y právě tehdy, když pro každé dvě funkce $g, h : Y \to Z$ platí $g \circ f = h \circ f \to g = h$.

3.3.11 Cvičení. Buďte $f : A \to B$ a $g : C \to D$ zobrazení. Potom také $f \times g : A \times C \to B \times D$ definované předpisem $(f \times g)(x, y) = (f(x), g(y))$ je zobrazení, a platí (i) $f \times g$ je prosté právě když f, g jsou prostá; (ii) $f \times g$ je na právě když f, g jsou na; (iii) $f \times g$ má inverz právě když f, g mají inverz, a v tom případě je $(f \times g)^{-1} = f^{-1} \times g^{-1}$.

3.4 Ekvivalence a uspořádání

V tomto oddíle se budeme věnovat dvěma typům relací, které mají v teorii množin i v ostatní matematice výsadní postavení: ekvivalence a uspořádání.

3.4.1 Definice. Binární relace R na množině X je

(a) *reflexivní*, pokud pro každé $x \in X$ je xRx.

(b) *antireflexivní*, pokud pro žádné $x \in X$ není xRx.

(c) *symetrická*, pokud pro každé xRy je také yRx.

(d) *antisymetrická*, pokud pro každé xRy je $\neg yRx$.

(e) *slabě antisymetrická*, pokud $xRy \wedge yRx$ platí jen pro $x = y$.

(f) *transitivní*, pokud při $xRy \wedge yRz$ je také xRz.

3.4.2 Definice. *Ekvivalence* na množině X je relace, která je reflexivní, symetrická a transitivní. Je-li $\equiv$ nějaká ekvivalence na X, pak pro prvek $x \in X$ je množina $[x] = \{y \in X; x \equiv y\}$ jeho *ekvivalenční třída*. Množina $X/\equiv = \{[x]; x \in X\}$ je potom *kvocient* neboli *faktorizace* množiny X podle ekvivalence $\equiv$.

Pojem ekvivalence je způsob, jak vyjádřit stejnost či podobnost: zavedeme-li na dané množině nějakou ekvivalenci, vyjadřujeme tím, které její prvky považujeme v nějakém ohledu za „stejné“. Z tohoto pohledu jsou požadavky na reflexivitu, transitivitu a symetrii velmi přirozené. Relaci ekvivalence na množině je obvyklé značit nějakým sugestivním symbolem jako $\equiv$ či $\approx$ apod.

Ekvivalenční třída nějakého prvku sestává právě z těch prvků, které jsou s ním ekvivalentní; zřejmě dva prvky jsou ekvivalentní právě tehdy, když jejich ekvivalenční třídy jsou totožné, tj. $x \equiv y$ právě když $[x] = [y]$.

3.4.3 Příklad. (a) Krajním příkladem ekvivalence je rovnost: každý prvek je ekvivalentní jen sám se sebou, všechny ekvivalenční třídy jsou jednoprvkové. Druhým extrémem je relace $X \times X$, ve které každé dva prvky jsou navzájem ekvivalentní, a celá množina X je jedinou ekvivalenční třídou. (b) Pro výrokové formule φ a ψ položme $\varphi \equiv \psi$ právě tehdy, když jsou splněny při stejných pravdivostních ohodnoceních. (c) Pro celá čísla[5] x, y buď $x \equiv y$ právě tehdy, když mají stejný zbytek po dělení sedmi. Tím se množina $\mathbb{Z}$ celých čísel rozpadne na sedm disjunktních tříd. (d) Pro konečné automaty A a B buď $A \equiv B$ právě když přijímají stejný jazyk. Pro generativní gramatiky G_1, G_2 buď $G_1 \equiv G_2$ právě tehdy, když generují stejný jazyk. Pro Turingovy stroje S a T buď $S \equiv T$ právě tehdy, když na stejném vstupu dávají stejný výstup.

3.4.4 Definice. Systém $\mathcal{A} \subseteq P(X)$ tvoří *rozklad* množiny X, pokud množiny $A \in \mathcal{A}$ jsou neprázdné, navzájem disjunktní, a platí $X = \bigcup \mathcal{A}$.

3.4.5 Věta. *Buď E ekvivalence na X. Potom $\mathcal{A}(E) = \{[x]; x \in X\}$ je rozklad množiny X. Naopak je-li $\mathcal{A}$ rozklad množiny X, je relace $E(\mathcal{A})$ sestávající z dvojic $(x, y) \in X \times X$ splňujících $(\forall A \in \mathcal{A})(x \in A \leftrightarrow y \in A)$ ekvivalence na X.*

Důkaz. Buď E ekvivalence na X. Pro každé $x \in X$ je $x \in [x]$ díky reflexivitě, takže všechny třídy $[x]$ jsou neprázdné a jejich sjednocením je množina X. Zbývá ukázat, že různé ekvivalenční třídy jsou navzájem disjunktní. Buďte tedy $[x]$ a $[y]$ dvě různé ekvivalenční třídy. Pokud $[x]$ a $[y]$ nejsou disjunktní, znamená to, že $z \in [x] \cap [y]$ pro nějaké $z \in X$. Pak je z definice xEz a yEz, takže ze symetrie také zEy, a z transitivity pak xEy. Tedy $[x] = [y]$, spor.

Buď naopak $\mathcal{A}$ nějaký rozklad. Snadno se nahlédne, že *náležeti do stejné části rozkladu* je reflexivní, symetrický a transitivní vztah, neboli ekvivalence. □

Navíc platí, že pro každou ekvivalenci E na množině X je $E(\mathcal{A}(E)) = E$, a pro každý rozklad $\mathcal{A}$ množiny X je $\mathcal{A}(E(\mathcal{A})) = \mathcal{A}$. Mezi ekvivalencemi na množině a jejími rozklady tedy existuje jednoznačná korespondence.

3.4.6 Definice. Binární relace $<$ na množině X je *uspořádání* množiny X, pokud je antireflexivní a transitivní. Dva různé prvky $x, y \in X$ jsou *porovnatelné*, pokud platí buďto $x < y$ nebo $y < x$; jinak jsou *neporovnatelné*. Uspořádání, ve kterém každé dva prvky jsou porovnatelné, je *lineární*. Podmnožina $C \subseteq X$ je *řetězec*, pokud je $<$ lineární uspořádání na C.

Uspořádání budeme značit jako $<$, $\prec$, $\sqsubset$ či podobně, uspořádanou množinu spolu se zvoleným uspořádáním jako $(X, <)$.

3.4.7 Příklad. (a) Nejjednodušší možné uspořádání je *diskrétní uspořádání* prázdnou relací. (b) Tradiční číselné obory $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ jsou lineárně uspořádané. (c) Potenční množina $P(X)$ množiny X je uspořádána inkluzí. Pokud X obsahuje alespoň dva prvky, pak toto uspořádání není lineární. (d) Rozklad $\mathcal{A}$ množiny X *zjemňuje* rozklad $\mathcal{B}$, pokud pro každou $A \in \mathcal{A}$ existuje právě jedna $B \in \mathcal{B}$ tak, že $A \subseteq B$ (a zároveň není $\mathcal{A} = \mathcal{B}$). V takovém případě píšeme $\mathcal{A} \prec \mathcal{B}$. Systém všech rozkladů množiny X je potom uspořádán relací zjemnění. (e) Je-li

[5] Řekli jsme, že v teorii množin lze vybudovat základy veškeré běžné matematiky, ale například okruh celých čísel jsme dosud formálně nezavedli. V příkladech ale budeme i nadále čerpat z oblastí, které čtenář zná (elementární algebra, geometrie, analýza) ještě předtím, než příslušné objekty definujeme v teorii množin.

$(A, \leq)$ nějaká konečná lineárně uspořádaná množina (*abeceda*), buď A^* množina všech konečných posloupností prvků z A (všech *slov* v abecedě A). Lineární uspořádání abecedy se pak rozšiřuje na *lexikografické uspořádání* slov stejným způsobem, jako uspořádání slov ve slovníku: pro slova $u = a_1a_2a_3 \dots a_k$ a $v = b_1b_2b_3 \dots b_l$ je $u \preceq v$ právě tehdy, když pro každé $i \leq \min\{k, l\}$ je $a_i \leq b_i$. (f) Množina $Seq = \bigcup_{n\in\mathbb{N}} 2^n$ všech konečných binárních posloupností je uspořádána relací *prodloužení*, kde $(x_1, \dots, x_k) \sqsubseteq (y_1, \dots, y_l)$ právě tehdy, když $k \leq l$ a $(\forall i \leq k)(x_i = y_i)$.

Uspořádání tak jak je definováno výše se obšírněji nazývá *ostré uspořádání*. Každému ostrému uspořádání $<$ odpovídá právě jedno *neostré* uspořádání $\leq$ na téže množině, položíme-li $x \leq y$ právě když $x < y$ nebo $x = y$. Takové neostré uspořádání je potom reflexivní, transitivní, a slabě antisymetrické.

Naopak každému neostrému uspořádání $\leq$ odpovídá právě jedno ostré uspořádání, položíme-li $x < y$ právě když $x \leq y$ a $x \neq y$. V dalším budeme volně přecházet mezi ostrým a odpovídajícím neostrým uspořádáním, podle toho, která formulace bude momentálně výhodnější.

3.4.8 Definice. Buď $(X, <)$ uspořádání. Prvek $a \in X$ je *nejmenší* (*největší*), pokud pro každé $x \in X$ je $a \leq x$ ($a \geq x$). Prvek $a \in X$ je *minimální* (*maximální*), pokud pro žádné $x \in X$ není $x < a$ ($x > a$).

3.4.9 Příklad. (a) Pokud číslo $m \in \mathbb{N}$ dělí číslo $n \in \mathbb{N}$, budeme psát $m|n$. Relace dělitelnosti je pak uspořádání na množině $\mathbb{N}$ přirozených čísel. Jeho nejmenším prvkem je číslo 1, které dělí všechna ostatní čísla, největším prvkem je číslo 0. Minimálními prvky množiny $\mathbb{N}\setminus\{1\}$ uspořádané dělitelností jsou právě všechna prvočísla; žádné z nich není nejmenším prvkem.

(b) Prázdná posloupnost je zřejmě nejmenším prvkem v $(Seq, \sqsubseteq)$, žádný maximální prvek v Seq neexistuje. Množina $\bigcup_{n\leq m} 2^n$ má 2^m maximálních prvků.

(c) Úplná teorie je právě taková, která je maximální mezi všemi bezesporných teoriemi (v daném jazyce) při uspořádání inkluzí.

Nejmenší prvek $a \in X$, pokud existuje, je zároveň (jediný) minimální, ale nikoli naopak, jak ukazuje předchozí příklad. Uspořádání s více než jedním minimálním prvkem tedy nemůže mít žádný nejmenší prvek. Navíc minimální prvky jsou navzájem neporovnatelné, takže takové uspořádání nemůže být lineární. Analogická tvrzení můžeme vyslovit o největších a maximálních prvcích.

Řekneme, že uspořádání $\preceq$ množiny X *rozšiřuje* uspořádání $\leq$, pokud pro každé $x \leq y$ je také $x \preceq y$. Například obvyklé uspořádání množiny $\mathbb{N}$ přirozených čísel rozšiřuje uspořádání podle dělitelnosti, a lexikografické uspořádání konečných binárních posloupností rozšiřuje relaci prodlužování na Seq.

3.4.10 Lemma. *Buď $(X, \leq)$ uspořádání, ve kterém prvky x, y jsou neporovnatelné. Pak existuje rozšíření $(X, \preceq)$, ve kterém je $x \preceq y$.*

Důkaz. Pro prvky $p, q \in X$ položme $p \preceq q$ právě tehdy, když je buďto $p \leq q$ nebo $p \leq x$ a $y \leq q$. Relace $(X, \preceq)$ zřejmě rozšiřuje relaci $(X, \leq)$ a máme $x \preceq y$. Zbývá ověřit, že $\preceq$ je vskutku uspořádáním. To přenecháváme čtenáři. □

3.4.11 Lemma. *Systém $\mathcal{X}$ všech uspořádání na množině X je uspořádán inkluzí, a jeho maximální prvky jsou právě lineární uspořádání množiny X.*

Důkaz. Snadno se nahlédne, že inkluze je uspořádáním na $\mathcal{X} \subseteq P(P(X \times X))$. Uspořádání množiny X, které není lineární, není podle předchozího lemmatu maximální v $\mathcal{X}$. Je-li naopak $\leq$ lineární uspořádání na X, je pro každé dva prvky $x, y \in X$ buďto $x \leq y$ nebo $y \leq x$. Uspořádání $\leq$ tedy nelze rozšířit přidáním žádné další porovnatelné dvojice $x \preceq y$ ani $y \preceq x$: je-li $x \leq y$ (opačný případ je podobný), pak $x \preceq y$ nic nepřidává a $y \preceq x$ znamená $x = y$. To znamená, že $\leq$ je maximální prvek $\mathcal{X}$ vůči inkluzi. □

Lineární uspořádání jsou taková, která již nelze rozšířit. Proto se někdy lineární uspořádání nazývají *totální*, kdežto ostatní uspořádání jsou *částečná*.

3.4.12 Cvičení. (a) Které z vlastností 3.4.1 mají následující relace na množině přirozených čísel? m dělí n; m je menší než n; $m + n \geq 1000$; $m + n$ je sudé; $m + n$ je násobkem sedmi; $m * n$ je sudé; $m = n^k$ pro nějaké k přirozené. (b) Které z uvedených vlastností má relace $|x - y| \in \mathbb{Q}$ mezi reálnými čísly? (c) Které z těchto vlastností má relace kolmosti (rovnoběžnosti, mimoběžnosti) mezi přímkami v rovině (v prostoru)?

3.4.13 Cvičení. Popište všechna uspořádání na množině se dvěma, třemi, čtyřmi prvky. Kolik různých uspořádání existuje na n-prvkové množině?

3.4.14 Cvičení. (a) Popište nejmenší, největší, minimální a maximální prvky uspořádání z 3.4.7. (b) Maximální prvek lineárního uspořádání je jediný a největší. (c) Popište uspořádání s jediným maximálním prvkem, který není největší.

3.4.15 Cvičení. Popište nějakou množinu $A \subseteq \mathbb{N}$, která má při uspořádání dělitelností právě m minimálních a právě n maximálních prvků. Pro která čísla $m, n \in \mathbb{N}$ taková množina existuje?

3.4.16 Cvičení. Maximální řetězec v $(Seq, \sqsubseteq)$ je *větev*. Lexikografické uspořádání všech větví v Seq je lineární, zatímco $(Seq, \sqsubseteq)$ lineární není.

3.4.17 Cvičení. Konečnou aplikací lemmatu 3.4.10 můžeme každé konečné uspořádání rozšířit do lineárního, neboť případných neporovnatelných dvojic je pouze konečně mnoho. Implementujte algoritmus, který pro konečné uspořádání efektivně nalezne nějaké lineární rozšíření.

3.5 Ordinální čísla

3.5.1 Definice. Buď $(X, <)$ uspořádaná množina. Řekneme, že uspořádání $(X, <)$ je *dobré*, pokud každá jeho neprázdná část má nejmenší prvek.

Každá část dobře uspořádané množiny je sama dobře uspořádána toutéž relací, speciálně každý *počáteční úsek* $(\leftarrow, x) = \{y \in X; y < x\}$ určený prvkem $x \in X$. Dobré uspořádání je nutně lineární, neboť každá dvouprvková množina $\{x, y\}$ má nejmenší prvek, takže každé dva prvky jsou porovnatelné.

3.5.2 Definice. Zobrazení $f : (X, <) \to (Y, \prec)$ mezi uspořádanými množinami je *monotónní*, pokud pro $x_1 < x_2$ je $f(x_1) \prec f(x_2)$. Monotónní bijekce je *isomorfismus* mezi uspořádáními $(X, <)$ a $(Y, \prec)$. Pokud mezi dvěma uspořádáními $(X, <)$ a $(Y, \prec)$ nějaký isomorfismus existuje, řekneme, že jsou *isomorfní* a píšeme $(X, <) \simeq (Y, \prec)$. Isomorfismus $(X, <)$ se sebou je *automorfismus*.

Složení monotónních zobrazení je monotónní, složení automorfismů je automorfismus, a inverzní zobrazení k automorfismu je rovněž automorfismem. *Býti isomorfní* je relace ekvivalence na třídě uspořádaných množin.

3.5.3 Lemma. *Je-li $(X,<)$ dobré uspořádání a $f:(X,<)\to(X,<)$ je monotónní, je $f(x)\geq x$ pro každé $x\in X$. Jediným automorfismem $(X,<)$ je identita. Jsou-li dvě dobrá uspořádání isomorfní, pak mezi nimi existuje jediný isomorfismus. Dobré uspořádání není isomorfní s žádným svým počátečním úsekem.*

Důkaz. Je-li množina $\{x\in X; f(x)<x\}$ neprázdná, buď x její nejmenší prvek. Potom pro $y=f(x)<x$ je $f(y)<f(x)=y$, spor. Je-li $f:(X,<)\to(X,<)$ automorfismus, je i f^{-1} automorfismus, takže je $f(x)\geq x$ i $f^{-1}(x)\geq x$ pro každé $x\in X$; tedy také $x=f(f^{-1}(x))\geq f(x)$. Jsou-li $f,g:(X,<)\to(Y,\prec)$ dva různé isomorfismy, je složené zobrazení $g^{-1}\circ f$ neidentickým automorfismem na X, spor. Je-li f isomorfismus mezi uspořádáním X a nějakým jeho počátečním úsekem $(\leftarrow,x)$, je $f(x)<x$, spor. □

3.5.4 Věta. *Buďte $(X,<)$ a $(Y,\prec)$ dobrá uspořádání. Pak nastává právě jedna z následujících možností:*

(i) *$(X,<)$ a $(Y,\prec)$ jsou isomorfní.*

(ii) *$(X,<)$ je isomorfní s nějakým počátečním úsekem $(Y,\prec)$.*

(iii) *$(Y,\prec)$ je isomorfní s nějakým počátečním úsekem $(X,<)$.*

Důkaz. Položme $f=\{(x,y)\in X\times Y; (\leftarrow,x)$ je isomorfní s $(\leftarrow,y)\}$. Z předchozího lemmatu plyne, že f je prostá funkce z X do Y. Navíc f je monotónní: je-li $x_1<x_2$ a $g:(\leftarrow,x_2)\simeq(\leftarrow,y_2)$ je isomorfismus, jsou i $(\leftarrow,x_1)$ a $(\leftarrow,g(x_1))$ isomorfní, přitom $g(x_1)\prec y_2$.

Pokud je $dom(f)=X$ a $rng(f)=Y$, nastává případ (i). Pokud $rng(f)\neq Y$, buď y nejmenší prvek množiny $Y\setminus rng(f)$. Pak je $rng(f)=(\leftarrow,y)$, neboť $rng(f)$ je uzavřená dolů. V tom případě je ale $dom(f)=X$, neboť jinak pro nejmenší prvek x množiny $X\setminus dom(f)$ máme $f(x)=y$, spor. Tedy nastává případ (ii). Podobně se ukáže, že pokud $dom(f)\neq X$, nastává případ (iii). Podle předchozího lemmatu se tyto případy vzájemně vylučují. □

Právě dokázaná věta říká, že každé dvě dobře uspořádané množiny jsou porovnatelné podle délky: jedna je počátečním úsekem druhé. Pokud jsou dvě dobře uspořádané množiny isomorfní, řekneme, že jejich uspořádání jsou *stejného typu*. Ordinální čísla jsou pak právě typy dobrých uspořádání.

3.5.5 Definice. (i) Množina je *transitivní*, pokud každý její prvek je i její částí.
(ii) *Ordinální číslo* je transitivní množina dobře uspořádaná relací náležení.

Prázdná množina $\emptyset$ je ordinál, a stejně tak množiny

$$1=\emptyset\cup\{\emptyset\}=\{0\},$$
$$2=1\cup\{1\}=\{0,1\}=\{\emptyset,\{\emptyset\}\},$$
$$3=2\cup\{2\}=\{0,1,2\}=\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\},$$
$$4=3\cup\{3\}=\{0,1,2,3\}=\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\},\{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}\}.$$

Níže zavedeme přirozená čísla právě jako typy konečných dobrých uspořádání. Induktivní množina zaručená axiomem nekonečna obsahuje jako prvky

všechny ordinály, které lze získat z prázdné množiny výše naznačeným způsobem. Třída ordinálních čísel pak tuto škálu prodlužuje. Ordinální čísla budeme značit písmeny $\alpha, \beta, \gamma, \ldots$ z řecké abecedy. Třídu ordinálních čísel budeme značit On.

3.5.6 Lemma. *Množina x je transitivní právě když $\bigcup x \subseteq x$, což je právě když $x \subseteq P(x)$. Jsou-li množiny x, y transitivní, jsou také $x \cap y$ a $x \cup y$ transitivní. Jsou-li všechny prvky množiny x transitivní, jsou také $\bigcap x$ a $\bigcup x$ transitivní.*

3.5.7 Lemma. *Buďte α, β ordinální čísla. Potom* (i) *$\alpha \notin \alpha$;* (ii) *je-li $x \in \alpha$, je také x ordinál;* (iii) *$\alpha \in \beta$ právě když $\alpha \subset \beta$;* (iv) *je buďto $\alpha \in \beta$ nebo $\beta \in \alpha$.*

Důkaz. (i) Pokud $\alpha \in \alpha$, není $\in$ uspořádáním množiny α. (ii) Pro $x \in \alpha$ je $x \subseteq \alpha$, takže i množina x je dobře uspořádaná relací náležení. Pro $z \in y \in x$ je $z \in x$, neboť α je transitivní množina a $\in$ je transitivní relace na α. (iii) Jeden směr plyne z transitivity. Je-li naopak $\alpha \subset \beta$, buď γ nejmenší prvek neprázdné množiny $\beta \setminus \alpha$. Potom z transitivity je α právě počáteční úsek ordinálu β určený prvkem γ. Tedy $\alpha = \{\xi \in \beta; \xi \in \gamma\} = \gamma \in \beta$. (iv) Množina $\alpha \cap \beta = \gamma$ je ordinál, přitom $\gamma \subseteq \alpha$ a $\gamma \subseteq \beta$. Kdyby $\gamma \subset \alpha$ a $\gamma \subset \beta$, máme $\gamma \in \alpha \cap \beta = \gamma$ podle (iii), což je spor s (i). Je tedy $\gamma = \alpha$ nebo $\gamma = \beta$, takže $\alpha \in \beta$ nebo $\beta \in \alpha$. □

Pro ordinály α, β budeme obvykle psát $\alpha < \beta$ místo $\alpha \in \beta$. Každý ordinál β je tedy právě množinou $\{\alpha; \alpha < \beta\}$ svých předchůdců.

3.5.8 Věta. *On je transitivní vlastní třída dobře uspořádaná relací náležení.*

Důkaz. Podle (ii) předchozího lemmatu je On transitivní a podle (i) je $\in$ antireflexivní relace na On. Zároveň $\in$ je transitivní relace na On, neboť všechny prvky On jsou transitivní množiny. Tedy $\in$ je ostré uspořádání třídy On. Podle (iv) předchozího lemmatu je toto uspořádání lineární. Ukážeme, že je dobré. Buď $X \subseteq On$ nějaká množina ordinálů. Zvolme $\alpha \in X$ libovolně. Pokud je α nejmenší prvek množiny X, jsme hotovi. V opačném případě je $\alpha \cap X$ neprázdná podmnožina ordinálu α, takže má nejmenší prvek, označme ho β. Snadno se ověří, že β je zároveň nejmenším prvkem množiny X. Ukázali jsme, že transitivní třída On je dobře uspořádaná relací náležení. Z toho plyne, že On nemůže být množinou, tedy ordinálem, neboť pak by platilo $On \in On$, což je spor. □

Zároveň je třída On jediná s uvedenými vlastnostmi. Je-li totiž X transitivní vlastní třída dobře uspořádaná relací náležení, jsou všechny prvky $x \in X$ ordinály, takže $X \subseteq On$. Je-li $X \subset On$, buď $\alpha \in On \setminus X$. Potom z transitivity je $X \subseteq \alpha$, takže X je množina, spor.

3.5.9 Lemma. *Buď $X \subseteq On$ neprázdná množina. Potom $\bigcap X$ je nejmenší prvek a $\bigcup X$ je supremum množiny X v dobrém uspořádání třídy On.*

Důkaz. Buď $\alpha \in X$ nejmenší prvek. Z definice je $\bigcap X \subseteq \alpha$. Přitom pro $\beta \in \alpha$ platí $\beta \in x$ pro každé $x \in X$, neboli $\beta \in \bigcap X$, takže je zároveň $\alpha \subseteq \bigcap X$.

Množina $\bigcup X$ je transitivní, takže $\bigcup X \subseteq X$ je dobře uspořádaná náležením. Tedy $\bigcup X$ je ordinální číslo. Je-li $\beta \in X$ největší prvek, je dokonce $\bigcup X = \beta$. Pokud X nemá největší prvek, je $X \subseteq \bigcup X$, takže $\bigcup X$ je majorantou všech $\alpha \in X$. Naopak libovolná majoranta $\gamma \in On$ množiny X obsahuje každé $\alpha \in X$ jako prvek, tedy i jako část $\alpha \subseteq \gamma$, a máme $\bigcup X \subseteq \gamma$. □

3.5.10 Věta. *Každá dobře uspořádaná množina je isomorfní právě s jedním ordinálním číslem. Navíc isomorfismus mezi nimi je jednoznačný.*

Důkaz. Buď $(X, <)$ dané dobré uspořádání. Každý ordinál je podle 3.5.4 buďto isomorfní s X (a jsme hotovi), nebo je isomorfní s nějakým počátečním úsekem X, nebo naopak. Přitom On je vlastní třída, takže není možné, aby každý ordinál byl isomorfní s nějakým počátečním úsekem X. Buď tedy $\beta \in On$ ordinál, pro který je naopak $(X, <)$ isomorfní s nějakým počátečním úsekem ordinálu β. Takový počáteční úsek $(\leftarrow, \alpha) \subseteq \beta$ je ovšem právě ordinál $\alpha < \beta$. □

3.5.11 Definice. Pro ordinál α buď $\alpha + 1$ množina $\alpha \cup \{\alpha\}$. Ordinál β je *izolovaný*, je-li $\beta = 0$ nebo je tvaru $\beta = \alpha + 1$ pro nějaké $\alpha < \beta$. Jinak je *limitní*.

Je-li α ordinální číslo, je také $\alpha \cup \{\alpha\}$ ordinální číslo: je transitivní, neboť α je transitivní, a jakožto množina ordinálů je dobře uspořádaná relací náležení. Zároveň je $\alpha \cup \{\alpha\}$ nejmenší ordinál nad α: pro $\beta < \alpha \cup \{\alpha\}$ je buďto $\beta \in \alpha$ nebo $\beta = \alpha$, čili $\beta \leq \alpha$; pod $\alpha \cup \{\alpha\}$ tedy leží jen ordinály $\leq \alpha$. To ospravedlňuje značení $\alpha + 1$. Ordinál $\alpha + 1$ je *následníkem* ordinálu α ve třídě On.

Ordinál $\beta \in On$ je limitní, právě když není následníkem, tedy když pro $\alpha < \beta$ je také $\alpha + 1 < \beta$. Induktivní množina zaručená axiomem nekonečna obsahuje $\emptyset$ jako prvek, a s každým prvkem x obsahuje také $x \cup \{x\}$; obsahuje tedy celý počáteční úsek třídy On. Formule $\varphi(z)$, která říká *z je izolovaný ordinál a všechny jeho prvky rovněž,* pak z této množiny vyděluje první limitní ordinál.

3.5.12 Definice. Buď ω nejmenší limitní ordinální číslo. Předchůdci ordinálu ω ve třídě ordinálních čísel jsou *přirozená čísla*.

Přirozené číslo je tedy právě izolovaný ordinál s izolovanými předchůdci, a axiom nekonečna zaručuje existenci limitních ordinálů. Nad každým ordinálem α existuje limitní ordinál $\beta > \alpha$: stačí položit $\alpha_0 = \alpha$ a $\alpha_{n+1} = \alpha_n + 1$, ordinál $\beta = \bigcup \alpha_n$ je pak limitní. Třída On obsahuje neomezeně mnoho isolovaných i neomezeně mnoho limitních ordinálů.

3.5.13 Definice. Funkce s definičním oborem ω je *posloupnost*, v případě zobrazení z ω do X pak *posloupnost v množině* X, kterou obvykle značíme $(x_n; n < \omega)$. Analogicky definujeme *konečnou posloupnost* $(x_i; i < n)$ pro $n \in \omega$ přirozené a *transfinitní posloupnost* $(x_\xi; \xi < \alpha)$ pro ordinály $\alpha > \omega$.

Transfinitní indukce Následník $\omega + 1$ limitního ordinálu ω je isolovaný, ale není to přirozené číslo. Třída ordinálů překračuje hranici přirozených čísel. Pro třídu On platí následující indukční princip, silnější než princip indukce pro přirozená čísla: dovoluje učinit indukční krok i pro limitní ordinály.

3.5.14 Věta (o transfinitní indukci). *Buď $X \subseteq On$ třída ordinálních čísel taková, že pro každý ordinál $\alpha \in On$ platí $\alpha \subseteq X \to \alpha \in X$. Potom $X = On$.*

Důkaz. Pokud $X \subset On$, buď α nejmenší prvek třídy $On \setminus X$. Pak pro každé $\xi < \alpha$ je $\xi \in X$, tj. $\alpha \subseteq X$, ale není $\alpha \in X$, spor. □

Transfinitní rekurze V matematice jsou běžné různé rekurzivní konstrukce. Typicky se v každém n-tém kroku (pro n přirozené) provádí nějaká stále stejná

operace nad předchozími kroky $i < n$. Dokážeme větu o transfinitní rekurzi, která zobecňuje takové konstrukce rekurzí na třídu všech ordinálních čísel.

Řekneme, že třída $F = \{(x, y); \varphi(x, y)\}$ je (třídové) zobrazení, pokud pro každou množinu x existuje nanejvýš jedna množina y tak, že $\varphi(x, y)$. V takovém případě píšeme $F(x) = y$. Třídové zobrazení F definované na On je *ordinální funkce*. Podobně jako u množinových zobrazení definovaných na nějakém ordinálu, značíme třídové *ordinální posloupnosti* jako $(x_\alpha; \alpha \in On)$.

3.5.15 Věta (o transfinitní rekurzi)**.** *Buď G třídové zobrazení. Potom existuje jediná ordinální funkce F taková, že pro každý ordinál α je $F(\alpha) = G(F \restriction \alpha)$.*

Důkaz. Jednoznačnost takové funkce F se dokáže indukcí: v bodě $\alpha = 0$ musí být $F(0) = G(F \restriction 0) = G(\emptyset)$, a jsou-li již známy hodnoty $F(\xi)$ pro $\xi < \alpha$, existuje jediná přípustná hodnota pro $F(\alpha)$. Definujeme tedy F jediným možným způsobem: $F(\alpha) = x$ právě když existuje α-posloupnost $(x_\xi; \xi < \alpha)$, pro kterou (i) pro každé $\xi < \alpha$ je $x_\xi = G((x_\eta; \eta < \xi))$ a (ii) $x = G((x_\xi; \xi < \alpha))$.

Posloupnost splňující (i) může existovat jen jedna, což se dokáže indukcí: pokud je $(y_\xi; \xi < \alpha)$ jiná taková posloupnost, je $x_\xi = y_\xi$ pro každé $\xi < \alpha$. Hodnota $F(\alpha) = x$ je pak jednoznačně určena podmínkou (ii), tedy F je funkce. Z věty o indukci rovněž plyne, že pro každé $\alpha \in On$ taková posloupnost existuje, takže $F(\alpha)$ je definováno pro každé $\alpha \in On$. □

3.5.16 Důsledek. *Buď X množina a buď α ordinál. Je-li G funkce, která zobrazuje posloupnosti v X délky $< \alpha$ do X, pak existuje jediná α-posloupnost $(x_\xi; \xi < \alpha)$ v množině X taková, že pro každé $\xi < \alpha$ je $x_\xi = G((x_\eta; \eta < \xi))$.*

Ordinální aritmetika Zavedli jsme pojem ordinálního čísla a ukázali jsme, že ordinály jsou právě typy dobře uspořádaných množin. Nyní ukážeme, že s ordinálními čísly lze rozumně počítat: popíšeme ordinální aritmetiku. Speciálně na oboru přirozených čísel tak získáme model Peanovy aritmetiky v teorii množin.

3.5.17 Definice (ordinální součet)**.** Pro ordinály α, β položme

(i) $\alpha + 0 = \alpha$

(ii) $\alpha + (\beta + 1) = (\alpha + \beta) + 1$

(iii) $\alpha + \beta = \sup_{\xi<\beta} \alpha + \xi$ pro β limitní.

3.5.18 Definice (ordinální součin)**.** Pro ordinály α, β položme

(i) $\alpha \cdot 0 = 0$

(ii) $\alpha \cdot (\beta + 1) = (\alpha \cdot \beta) + \alpha$

(iii) $\alpha \cdot \beta = \sup_{\xi<\beta} \alpha \cdot \xi$ pro β limitní.

Operace ordinálního součtu a součinu mají některé vlastnosti známé z elementární aritmetiky. Indukcí podle β lze například dokázat, že operace $\alpha + \beta$ i $\alpha \cdot \beta$ jsou asociativní. Ani jedna z operací však není komutativní, například $1 + \omega = \sup_{n\in\omega} 1 + n = \omega \neq \omega + 1$ a $2 \cdot \omega = \sup_{n\in\omega} 2 \cdot n = \omega \neq \omega \cdot 2 = \omega + \omega$.

Ordinály jsme zavedli jako typy dobrých uspořádání, nabízí se tedy otázka, v jakém vztahu je ordinální aritmetika k dobrým uspořádáním.

3.5.19 Definice. Buďte $(A, <_A)$ a $(B, <_B)$ dvě navzájem disjunktní lineární uspořádání. Potom jejich *sumou* je následující lineární uspořádání na $A \cup B$: pro $x, y \in A \cup B$ buď $x < y$ právě tehdy, když je buďto (i) $x, y \in A$ a $x <_A y$, nebo (ii) $x, y \in B$ a $x <_B y$, nebo (iii) $x \in A$ a $x \in B$. Jejich *produktem* je následující lineární uspořádání na $A \times B$: pro $(a_1, b_1), (a_2, b_2) \in A \times B$ buď $(a_1, b_1) < (a_2, b_2)$ právě když je buďto (i) $a_1 <_A a_2$, nebo (ii) $a_1 = a_2$ a $b_1 <_B b_2$.

Snadno se ověří, že suma i produkt lineárních uspořádání je opět lineární, a že suma i produkt dobrých uspořádání jsou opět dobré. Můžeme obrazně říci, že suma $A \cup B$ vznikne položením uspořádání A a B za sebe (v tomto pořadí), zatímco produkt $A \times B$ vznikne položením několika kopií uspořádání B za sebe, totiž pro každý prvek množiny A jednou. Indukcí se pak dokáže:

3.5.20 Věta. *Pro každé dva ordinály α, β je ordinální součet $\alpha + \beta$ právě typ dobrého uspořádání disjunktní sumy $(\{0\} \times \alpha) \cup (\{1\} \times \beta)$, a ordinální součin $\alpha \cdot \beta$ je právě typ dobrého uspořádání produktu $\beta \times \alpha$.*

3.5.21 Definice. Ordinální funkce $F : On \to On$ je *normální*, právě když je *neklesající*, tj. pro $\alpha \leq \beta$ je a $F(\alpha) \leq F(\beta)$, a zároveň *spojitá*,[6] tj. pro β limitní je $F(\beta) = \sup_{\alpha<\beta} F(\alpha)$. Analogicky se zavede *normální posloupnost* $(x_\alpha; \alpha \in On)$.

Operace ordinálního součtu a součinu jsou přímo z definice normálními funkcemi druhého argumentu. Následující věta říká, že každá normální funkce má neomezeně mnoho *pevných bodů*.

3.5.22 Věta. *Buď F normální funkce a buď α libovolný ordinál. Potom existuje ordinální číslo $\beta \geq \alpha$, pro které je $F(\beta) = \beta$.*

Důkaz. Buď $\alpha_0 = \alpha$ daný ordinál. Známe-li již α_n, položme $\alpha_{n+1} = F(\alpha_n)$. Funkce F je neklesající, je tedy $\alpha_{n+1} \geq \alpha_n$. Pokud je $\alpha_{n+1} = F(\alpha_n) = \alpha_n$ pro nějaké n, jsme hotovi. V opačném případě máme nekonečnou rostoucí posloupnost $(\alpha_n; n \in \omega)$. Pro ordinál $\beta = \sup \alpha_n$ je $\beta \geq \alpha$, a ze spojitosti máme $F(\beta) = \sup_n F(\alpha_n) = \sup_n \alpha_{n+1} = \beta$. Navíc β je nejmenší pevný bod $> \alpha$. □

Pro operace ordinálního součtu a součinu to znamená, že pro každé α existuje neomezeně mnoho takových β, že $\alpha + \beta = \beta$, a neomezeně takových, že $\alpha \cdot \beta = \beta$.

3.5.23 Cvičení. Napište formuli základního jazyka teorie množin, která říká, že α je tranzitivní množina; α je dobře uspořádaná náležením; α je izolovaný ordinál; α je přirozené číslo.

3.6 Přirozená čísla

3.6.1 Věta. *Ordinální následník, ordinální součet a ordinální součin přirozených čísel je opět přirozené číslo. Množina ω přirozených čísel opatřená těmito operacemi je modelem Peanovy aritmetiky.*

[6] Třída ordinálních čísel nese přirozenou topologii, při které jsou spojitá právě taková zobrazení, která respektují suprema; to je důvodem pro zavedený název. Isolované a limitní ordinály jsou v této topologii právě isolované a hromadné body. Topologií On se zabývat nebudeme.

Důkaz. První tvrzení získáme snadno indukcí přímo z definice následníka, součtu a součinu: je-li n izolovaný ordinál sestávající z izolovaných ordinálů, má $n \cup \{n\}$ tutéž vlastnost; podobně pro součet $n+0$ a dále indukcí, podobně pro součin. Schema axiomů indukce plyne ihned z věty o indukci. Vlastnosti součtu a součinu jsou zachyceny přímo v definici. Následník libovolného ordinálu je z definice neprázdná množina, a je-li $m \cup \{m\} = n \cup \{n\}$ pro nějaká $m, n \in \omega$, je nutně $m = n$: jinak by muselo být $m \in n$ a zároveň $n \in m$, což není možné. □

Struktura $\mathbb{N} = (\omega, +, \cdot, 0, 1, \in)$ přirozených čísel tvoří tzv. *standardní model* aritmetiky v teorii množin. Z věty o kompaktnosti víme, že aritmetika má i jiné, *nestandardní* modely. V kapitole o teorii modelů takový model sestrojíme.

3.6.2 Příklad. Struktura přirozených čísel je odrazovým můstkem ke konstrukci dalších tradičních oborů algebry. Popíšeme nejprve, jak z přirozených čísel sestrojit okruh celých čísel.

Pro uspořádané dvojice $(a, b), (c, d) \in \mathbb{N} \times \mathbb{N}$ položme $(a, b) \equiv (c, d)$ právě když $a + d = b + c$, kde $+$ je sčítání přirozených čísel, tedy ordinální součet. Snadno se ověří, že relace $\equiv$ je ekvivalence.[7] Označme množinu ekvivalenčních tříd jako $\mathbb{Z}$, a pro $[(a, b)], [(p, q)] \in \mathbb{Z}$ položme

$$[(a, b)] \oplus [(p, q)] = [(a + p, b + q)]$$
$$[(a, b)] \otimes [(p, q)] = [(ap + bq, aq + bp)]$$

Musíme předně ukázat, že operace zavedené na ekvivalenčních třídách nezávisí na jejich reprezentantech; to jest, že pro $(a, b) \equiv (c, d)$ a $(p, q) \equiv (r, s)$ je také $[(a, b)] \oplus [(p, q)] = [(c, d)] \oplus [(r, s)]$ a $[(a, b)] \otimes [(p, q)] = [(c, d)] \otimes [(r, s)]$. To se však snadno ověří z definice.

Ukážeme nejprve, že $(\mathbb{Z}, \oplus, \otimes)$ spolu s konstantami $\mathbf{0} = [(0, 0)]$ a $\mathbf{1} = [(1, 0)]$ tvoří komutativní okruh s jednotkou. Asociativita a komutativita operací $\oplus$ a $\otimes$ na $\mathbb{Z}$ plyne z asociativity a komutativity operací na $\mathbb{N}$. Pro každé $[(a, b)] \in \mathbb{Z}$ máme $[(a, b)] \oplus [(0, 0)] = [(a, b)]$, takže $\mathbf{0}$ je neutrální vůči $\oplus$, a zároveň platí $[(a, b)] + [(b, a)] = [(a + b, b + a)] = [(0, 0)] = \mathbf{0}$. Tedy $(\mathbb{Z}, \oplus, \mathbf{0})$ je komutativní grupa. Je také $[(a, b)] \otimes [(1, 0)] = [(a, b)]$, takže $\mathbf{1} = [(1, 0)]$ je neutrální vůči $\otimes$ a $(\mathbb{Z}, \otimes, \mathbf{1})$ je komutativní monoid. Distributivitu $\otimes$ vůči $\oplus$ přenecháváme čtenáři.

Je-li $[(a, b)] \neq \mathbf{0} \neq [(p, q)]$, je $a \neq b, p \neq q$, a tedy $ap + bq \neq aq + bp$, takže $[(a, b)] \otimes [(p, q)] \neq \mathbf{0}$. To znamená, že okruh $(\mathbb{Z}, \oplus, \otimes, \mathbf{0}, \mathbf{1})]$ je oborem integrity.

Položme dále $[(a, b)] < [(p, q)]$ právě když $a + q \in p + b$. Podobně jako výše se ověří, že definice $<$ nezávisí na volbě reprezentantů. Relace $<$ je lineárním uspořádáním na $\mathbb{Z}$, což se ukáže z linearity uspořádání na $\mathbb{N}$. Tím je okruh $\mathbb{Z}$ rozdělen na kladná $x > \mathbf{0}$ a záporná $x < \mathbf{0}$. Přitom pro $x < y$ je $x \oplus z < y \oplus z$, a pro $x, y > \mathbf{0}$ je také $x \otimes y > \mathbf{0}$. To znamená, že $\mathbb{Z}$ je uspořádaný okruh.

Zobrazení $n \mapsto [(n, 0)]$ je potom isomorfním vnořením $(\omega, +, *, 0, 1, \in)$ do $(\mathbb{Z}, \oplus, \otimes, \mathbf{0}, \mathbf{1}, <)$; obrazem $\mathbb{N}$ je právě množina všech nezáporných celých čísel.

3.6.3 Příklad. Popíšeme konstrukci tělesa $\mathbb{Q}$ racionálních čísel z oboru integrity $\mathbb{Z}$ celých čísel. Obecně se jedná o algebraickou konstrukci *podílového nadtělesa.*

[7] Dvojice (a, b) budou hrát roli chybějících rozdílů $a - b$. Je potom přirozené požadovat, aby např. dvojice $(3, 7)$ reprezentovala stejný objekt jako $(5, 9)$, což je právě zavedená ekvivalence. Operace na ekvivalenčních třídách jsou pak zavedeny přirozeným způsobem.

Pro uspořádané dvojice $(a,b),(c,d) \in \mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$ položme $(a,b) \equiv (c,d)$ právě když $a \otimes d = b \otimes c$, kde $\otimes$ je násobení celých čísel, zavedené výše. Snadno se ověří, že $\equiv$ je relace ekvivalence.[8] Označme množinu ekvivalenčních tříd jako $\mathbb{Q}$, a pro $[(a,b)],[(p,q)] \in \mathbb{Q}$ položme

$$[(a,b)] \boxplus [(p,q)] = [((a \otimes q) \oplus (p \otimes b), b \otimes q)]$$
$$[(a,b)] \boxtimes [(p,q)] = [(a \otimes p, b \otimes q)]$$

Musíme opět ukázat, že operace zavedené na ekvivalenčních třídách nezávisí na zvolených reprezentantech; to však plyne snadno z definice.

Ukážeme, že $(\mathbb{Q}, \boxplus, \boxtimes)$ spolu s konstantami $\mathbf{0} = [(0,1)]$ a $\mathbf{1} = [(1,1)]$, kde 0 a 1 značí nulu a jednotku okruhu celých čísel, je komutativní těleso, do kterého se okruh $(\mathbb{Z}, \oplus, \otimes, 0, 1)$ isomorfně vnořuje.

Asociativita a komutativita operací $\boxplus$ a $\boxtimes$ na $\mathbb{Q}$ plyne snadno z asociativity a komutativity operací na $\mathbb{Z}$. Pro každé $[(a,b)] \in \mathbb{Q}$ je $[(a,b)] \boxplus [(0,1)] = [(a,b)]$, takže $\mathbf{0}$ je neutrální vůči $\boxplus$, a zároveň platí $[(a,b)] + [(-a,b)] = [(0,1)] = \mathbf{0}$, kde $-a$ je opačný prvek v $(\mathbb{Z}, \oplus)$. Tedy $(\mathbb{Q}, \boxplus, \mathbf{0})$ je komutativní grupa. Je také $[(a,b)] \boxtimes [(1,1)] = [(a,b)]$, takže $\mathbf{1} = [(1,1)]$ je neutrální vůči $\boxtimes$ a $(\mathbb{Q}, \boxtimes, \mathbf{1})$ je komutativní monoid. Navíc pro $[(a,b)] \neq \mathbf{0}$ je $[(a,b)] \boxtimes [(b,a)] = \mathbf{1}$, tedy každý nenulový prvek má inverzní prvek vůči násobení. Distributivita plyne snadno z distributivity v $\mathbb{Z}$. Struktura $(\mathbb{Q}, \boxplus, \boxtimes, \mathbf{0}, \mathbf{1})$ je tedy komutativní těleso.

Položíme-li $[(a,b)] \prec [(p,q)]$ právě když $a \otimes q < p \otimes b$, kde $<$ je uspořádání celých čísel, ověří se opět, že definice $\prec$ nezáleží na zvolených representantech, zavádí lineární uspořádání na $\mathbb{Q}$, a že se jedná o uspořádané těleso. Zároveň uspořádání $(\mathbb{Q}, \prec)$ je husté.

Zobrazení $k \mapsto [(k,1)]$ je potom isomorfním vnořením uspořádaného okruhu $(\mathbb{Z}, \oplus, \otimes, 0, 1, <)$ do uspořádaného tělesa $(\mathbb{Q}, \boxplus, \boxtimes, \mathbf{0}, \mathbf{1}, \prec)$.

Zkonstruovali jsme okruh celých čísel a těleso racionálních čísel v teorii množin. Algebraik pochopitelně nepracuje s racionálními čísly jako s ekvivalenčními třídami dvojic celých čísel — důležité je, že sestrojené obory mají vlastnosti, které od nich očekáváme; vnitřní podoba jejich prvků nás pak trápit nemusí.

3.7 Axiom výběru

3.7.1 Definice. Buď x neprázdná množina. Řekneme, že množina $c \subseteq \bigcup x$ je *výběrová množina* pro x, pokud pro každou neprázdnou $y \in x$ je množina $c \cap y$ jednoprvková. Zobrazení $f : x \to \bigcup x$ je *výběrová funkce* neboli *selektor* pro x, pokud pro každou neprázdnou $y \in x$ je $f(y) \in y$. *Axiom výběru* (AC) je pak následující tvrzení: na každé množině existuje výběrová funkce.

Axiom výběru se od axiomů teorie ZF liší v tom, že postuluje existenci množin, aniž by popisoval, co přesně mají být jejich prvky, narozdíl třeba od axiomu dvojice nebo potence. V tomto smyslu není *konstruktivní*. Podle AC například existuje výběrová funkce na $P(\mathbb{R})$, kterou těžko explicitně popsat.

[8] Dvojicemi (a,b) modelujeme zlomky a/b. Zavedená ekvivalence znamená právě tolik, že (a,b) reprezentuje tentýž zlomek jako (c,d). Pro důkaz transitivity potřebujeme fakt, že $\mathbb{Z}$ je obor integrity; například v okruhu $\mathbb{Z}_4$ relace $\equiv$ není transitivní.

Přijetím axiomu výběru se teorie ZF rozšiřuje do teorie ZFC. Dá se ukázat, že pokud je ZF bezesporná, je bezesporná i ZFC. Říkáme, že ZFC je *relativně bezesporná* vůči ZF. Zároveň je relativně bezesporná i teorie ZF rozšířená o negaci AC. To znamená, že teorie ZF není úplná: AC je nerozhodnutelná formule.

3.7.2 Věta. *Následující tvrzení jsou ekvivalentní:*

(i) *Na každé množině existuje výběrová funkce.*

(ii) *Každý systém navzájem disjunktních množin má výběrovou množinu.*

(iii) *Neprázdný soubor neprázdných množin má neprázdný kartézský součin.*

(iv) *Na každé množině existuje dobré uspořádání.*

(v) *Pro množiny x, y existuje buďto prosté vnoření x do y nebo naopak.*

Důkaz. (i $\leftrightarrow$ ii) Buď x daná množina. Bez újmy na obecnosti předpokládáme, že je neprázdná, a všechny $y \in x$ jsou neprázdné. Uvažme systém $\{\{y\} \times y; y \in x\}$. To je systém navzájem disjunktních množin, podle (ii) pro něj tedy existuje výběrová množina c. Ta je ovšem výběrovou funkcí na x. Naopak, je-li x uvažovaný systém a f výběrová funkce na x, je $c = rng(f)$ výběrová množina pro x.

(i $\leftrightarrow$ iii) Buď x neprázdná množina sestávající z neprázdných $y \in x$. Potom prvky kartézského součinu $\prod \{y; y \in x\}$ jsou právě selektory na x.

(i $\leftrightarrow$ iv) Buď x daná množina a buď δ ordinál, který nelze prostou funkcí zobrazit do x. Takový ordinál musí existovat, neboť On je vlastní třída. Buď f výběrová funkce na množině $P(x) \setminus \{\emptyset\}$. Ordinální indukcí pak definujeme funkci $g : \delta \to x$ tak, že $g(\alpha) = f(x \setminus g[\alpha])$, dokud je množina $x \setminus g[\alpha]$ neprázdná. Funkce g je z definice prostá, takže pro nějaké $\alpha < \delta$ musí být $x \setminus g[\alpha]$ prázdná. Pak ale $g \restriction \alpha : \alpha \to x$ je prosté zobrazení α na x, které na množině x indukuje dobré uspořádání typu α. Je-li naopak $\prec$ dobré uspořádání množiny $\bigcup x$, je $f(y) = \min_{\prec} y$ výběrová funkce na x.

(iv $\leftrightarrow$ v) Pokud se množiny x, y dají dobře uspořádat, zvolme nějaká jejich dobrá uspořádání. Potom podle 3.5.4 máme buďto vnoření x do y (pišme $x \preceq y$) nebo $y \preceq x$. Naopak, pokud je δ ordinál takový, že $\delta \not\preceq x$, je podle (v) nutně $x \preceq \delta$, takže množina x nese dobré uspořádání typu $\leq \delta$. □

V dalším zformulujeme tři *principy maximality*, často používané v různých partiích matematiky, a ukážeme, že jsou ekvivalentní s axiomem výběru.

3.7.3 Definice. Systém $\mathcal{A} \subseteq P(X)$ je *konečného charakteru*, pokud pro každou $A \subseteq X$ je $A \in \mathcal{A}$ právě když platí $B \in \mathcal{A}$ pro všechny konečné $B \subseteq A$.

Pro systém $\mathcal{A}$ konečného charakteru platí při $B \subseteq A \in \mathcal{A}$ také $B \in \mathcal{A}$.

Systémy konečného charakteru se v matematice přirozeně vyskytují, například systém lineárně nezávislých podmnožin vektorového prostoru, systém řetězců v dané uspořádané množině či systém splnitelných teorií v daném jazyce.

3.7.4 Lemma (Tukey)**.** *Je-li $\mathcal{A} \subseteq P(X)$ systém konečného charakteru, pak pro každou množinu $A \in \mathcal{A}$ existuje $Y \supseteq A$, která je maximální v $(\mathcal{A}, \subseteq)$.*

Z Tukeyova lemmatu ihned plyne tvrzení Lindenbaumovy věty 2.4.16: každá bezesporná teorie má bezesporné zúplnění. Bezesporné teorie tvoří systém konečného charakteru, a úplná teorie je právě maximální bezesporná.

3.7.5 Lemma (Hausdorff). *Každé uspořádání obsahuje maximální řetězec.*

Maximální řetězec je taková lineárně uspořádaná podmnožina $C \subseteq (X, <)$, kterou nelze rozšířit do žádné větší lineárně uspořádané $D \supset C$. Ekvivalentně, řetězec $C \subset X$ je maximální právě když pro každý prvek $x \in X$ porovnatelný se všemi $y \in C$ je $x \in C$. Speciálně nejmenší a největší prvek uspořádání $(X, <)$, pokud existují, musí ležet v každém maximálním řetězci.

3.7.6 Lemma (Zorn). *Buď $(X, <)$ uspořádaná množina, ve které každý řetězec má horní mez. Potom pro každé $x \in X$ existuje maximální prvek $y \geq x$.*

3.7.7 Věta. *Každé z následujících tvrzení je ekvivalentní s axiomem výběru:*

(vi) *Tukeyovo lemma.*

(vii) *Hausdorffovo lemma.*

(viii) *Zornovo lemma.*

Důkaz. (iv $\to$ vi) Buď $\mathcal{A} \subseteq P(X)$ systém konečného charakteru a buď $A \in \mathcal{A}$. Nosná množina X má podle (iv) nějaké dobré uspořádání $\{x_\alpha; \alpha < \kappa\}$. Položme $Y_0 = A$, a dále indukcí: pokud je $Y_\alpha \cup \{x_\alpha\} \in \mathcal{A}$, buď $Y_{\alpha+1} = Y_\alpha \cup \{x_\alpha\}$, jinak buď $Y_{\alpha+1} = Y_\alpha$; pro $\beta < \kappa$ limitní buď $Y_\beta = \bigcup_{\alpha<\beta} Y_\alpha$; nakonec položme $Y = \bigcup_{\alpha<\kappa} Y_\alpha$. Zřejmě $Y \supseteq A$, zbývá ukázat, že Y je maximální v $\mathcal{A}$. Předně každá $Y_\alpha \in \mathcal{A}$: pro isolované ordinály z definice, pro limitní ordinály díky tomu, že $\mathcal{A}$ je konečného charakteru. Ze stejného důvodu je $Y \in \mathcal{A}$, neboť každá konečná $K \subset Y$ je podmnožinou nějaké Y_α, a tedy $K \in \mathcal{A}$. Zároveň je $Y \in \mathcal{A}$ maximální: kdyby $Y \subset Z \in \mathcal{A}$, buď $x_\alpha \in Z \setminus Y$ první ve zvoleném očíslování. Potom je $Y_\alpha \cup \{x_\alpha\} \subseteq Z \in \mathcal{A}$, tedy také $Y_\alpha \cup \{x_\alpha\} \in \mathcal{A}$, čili $Y_\alpha \cup \{x_\alpha\} = Y_{\alpha+1} \subseteq Y$, spor.

(vi $\to$ vii) Buď $(X, <)$ dané uspořádání. Systém všech řetězců v X je konečného charakteru: je-li $A \subseteq X$ řetězec, je i každé $B \subseteq A$ řetězec, a naopak pokud A není řetězec, jsou nějaké $x, y \in A$ neporovnatelné, takže konečná $\{x, y\} \subseteq A$ není řetězec. Podle (vi) má pak systém řetězců maximální prvek.

(vii $\to$ viii) Buď $(X, <)$ uspořádaná množina a $x \in X$ daný prvek. Pokud každý řetězec v X má horní mez, má horní množina $(x, \to) = \{y \in X; y \geq x\}$ tutéž vlastnost. Podle (vii) existuje v $(x, \to)$ maximální řetězec C; ten obsahuje x, jinak není maximální. Je-li potom y horní mez C, je $y \geq x$ maximální v $(X, <)$: kdyby $z > y$ pro nějaké z, bylo by $C \cup \{z\}$ prodloužení maximálního řetězce C.

(viii $\to$ vi) Je-li $\mathcal{A} \subseteq P(X)$ systém konečného charakteru, pak pro každý řetězec $\mathcal{C} \subseteq \mathcal{A}$ je také $\bigcup \mathcal{C} \in \mathcal{A}$. Tedy každý řetězec v $\mathcal{A}$ má horní mez. Nad zvoleným $A \in \mathcal{A}$ tedy podle Zornova lemmatu existuje maximální $Y \in \mathcal{A}$.

(viii $\to$ ii) Je-li x daný systém neprázdných, navzájem disjunktních množin, buď $\mathcal{S}$ množina *částečných selektorů* na x, tj. množin $c \subseteq \bigcup x$ takových, že pro každou $y \in x$ je $c \cap y$ nejvýše jednoprvková. Pak $(\mathcal{S}, \subseteq)$ splňuje předpoklady Zornova lemmatu, a každá maximální $c \in \mathcal{S}$ je výběrovou množinou pro x. □

3.7.8 Věta. *Každé z následujících tvrzení je důsledkem axiomu výběru.*

(a) *Každá bezesporná teorie má bezesporné zúplnění.*

(b) *Každé uspořádání má lineární rozšíření.*

(c) *Každý vektorový prostor má bázi.*

Důkaz. (a) Systém všech bezesporných teorií (v daném jazyce) je konečného charakteru, neboť případný spor v teorii je dokazatelný už z nějaké její konečné části. Přitom úplná teorie je právě maximální bezesporná. Tvrzení tedy plyne z Tukeyova lemmatu. (b) Lineární rozšíření daného uspořádání $(X, <)$ je podle 3.4.11 právě maximální prvek nad $<$ v systému $(\mathcal{X}, \subseteq)$ všech uspořádání na X. Snadno se ověří, že každý řetězec $\mathcal{C} \subseteq \mathcal{X}$ má horní mez $\bigcup \mathcal{C} \in \mathcal{X}$. Hledané rozšíření tedy existuje podle Zornova lemmatu. (c) Systém lineárně nezávislých podmnožin daného prostoru je konečného charakteru, neboť o případné lineární závislosti svědčí už nějaká konečná část. Přitom báze je právě maximální lineárně nezávislá podmnožina. Existence tedy plyne z Tukeyova lemmatu. □

3.7.9 Věta. *AC platí právě když každý souvislý graf má kostru.*

Důkaz. Buď $G = (V, E)$ souvislý graf. Pro stromy $S, T \subseteq G$ položme $S \prec T$ pokud $S \subseteq T$ je podgraf. Systém všech stromů G uspořádaný touto relací splňuje předpoklady Zornova lemmatu: snadno se ukáže, že sjednocení řetezce stromů je strom. To znamená, že existuje maximální strom $T \subseteq G$. Takový ale nutně obsahuje všechny vrcholy, neboť G je souvislý. Tedy T je kostrou G.

V opačném směru ukážeme, že pro každý systém neprázdných množin existuje výběrová množina. Buď $\mathcal{A} = \{A_i; i \in I\}$ takový systém; můžeme předpokládat, že množiny A_i jsou navzájem disjunktní. Popíšeme graf $G = (V, E)$. Buď $V = \bigcup \mathcal{A} \cup \{y_i, z_i; i \in I\} \cup \{x\}$, kde $x, y_i, z_i \notin \bigcup \mathcal{A}$ jsou nějaké navzájem různé množiny, a položme $E = \{xy_i; i \in I\} \cup \{ay_i, az_i; a \in A_i\}$. Potom $G = (V, E)$ je souvislý graf, podle předpokladu tedy má nějakou kostru $(T, F) \subseteq (V, E)$. Snadno se nahlédne, že každá $xy_i \in F$, že v každé množině A_i existuje právě jeden prvek $a_i \in A_i$ takový, že $ay_i, az_i \in F$, a že pro každé $a \in A_i \setminus \{a_i\}$ je $ay_i \in F$ právě když $az_i \notin F$. Potom $\{a_i; i \in I\}$ je výběrová množina pro $\mathcal{A}$. □

3.7.10 Věta. *AC platí právě když každá množina nese grupovou operaci.*

Důkaz. Buď X libovolná neprázdná množina. Stačí najít grupu mohutnosti $|X|$. Pro X konečnou máme cyklickou grupu $\mathbb{Z}_{|X|}$. Pro X nekonečnou má množina $[X]^{<\omega}$ mohutnost $|X|$ a nese přirozenou grupovou operaci symetrické diference.

Nechť naopak každá množina nese grupovou operaci. Je-li X neprázdná, najdeme pro ni dobré uspořádání. Buď δ nějaký ordinál, který se nevnořuje do X; takový nutně existuje, neboť On je vlastní třída. Podle předpokladu existuje na množině $G = X \cup \delta$ nějaká grupová operace, značme ji krátce $ab \in G$. Pro každé $x \in G$ je zobrazení $a \mapsto xa$ injektivní, neboť při $xa = xb$ je $a = x^{-1}xa = x^{-1}xb = b$. Speciálně pro každé $x \in X$ musí existovat nějaké $\alpha \in \delta$ tak, že $x\alpha \in \delta$, jinak by $\alpha \mapsto x\alpha$ bylo vnoření δ do X. Pro $x \in X$ buď $f(x)$ lexikograficky první pár $(\alpha, \beta) \in \delta \times \delta$ takový, že $x\alpha = \beta \in \delta$. Zobrazení f je pak vnořením X do $\delta \times \delta$, neboť při $f(x) = (\alpha, \beta) = f(y)$ je $x\alpha = \beta = y\alpha$, takže $x = y$ jako výše. Množina X tak dědí dobré uspořádání typu $\leq \delta \times \delta$. □

3.7.11 Cvičení. Napište axiom výběru v základním jazyce teorie množin.

3.7.12 Cvičení. Ukažte, že následující tvrzení je ekvivalentní s axiomem výběru: pro každou neprázdnou množinu X a každé zobrazení $f : X \to Y$ existuje zobrazení $g : Y \to X$ takové, že $f \circ g \circ f = f$.

3.7.13 Cvičení. Lineární uspořádání je dobré, právě když neobsahuje nekonečnou klesající posloupnost $x_1 > x_2 > \cdots > x_n > x_{n+1} > \ldots$

3.7.14 Cvičení. Popište nějaká lineární rozšíření $(2^\omega, \sqsubset)$ a $(P(\mathbb{N}), \subset)$.

3.7.15 Cvičení. Uspořádání $<$ množiny X je *definovatelné*, pokud existuje formule $\varphi(x, y)$ jazyka teorie množin tak, že pro $x, y \in X$ je $x < y$ právě když $\varphi(x, y)$. Definujte formulí nějaké dobré uspořádání množiny reálných čísel.

3.8 Kardinální čísla

Mohutnosti Pojem vzájemně jednoznačného zobrazení je klíčem k porovnávání množin podle mohutnosti — množiny, mezi kterými existuje bijekce, mají v dobrém smyslu „stejně prvků“, aniž bychom zatím hovořili o jejich „počtu“.

3.8.1 Definice. Pokud existuje bijekce $f : A \to B$, řekneme, že A a B mají *stejnou mohutnost* a píšeme $A \approx B$. Existuje-li prosté zobrazení z A do B, řekneme, že A má *nanejvýš takovou mohutnost* jako B, a píšeme $A \preceq B$. Pokud existuje prosté zobrazení z A do B, ale žádné zobrazení z A na B, řekneme, že množina A má *menší mohutnost* než B, a píšeme $A \prec B$.

3.8.2 Příklad. (a) Funkce $f(n) = 2n$ je prosté zobrazení $\mathbb{N}$ na množinu sudých čísel, takže množina přirozených čísel má stejnou mohutnost jako její vlastní část. (b) Každé dva otevřené intervaly na reálné přímce mají stejnou mohutnost. (c) Funkce $arctg : \mathbb{R} \to (-\frac{\pi}{2}, \frac{\pi}{2})$ je bijekce, takže interval $(-\frac{\pi}{2}, \frac{\pi}{2})$ má stejnou mohutnost jako reálná přímka. Tutéž vlastnost má potom i každý jiný interval.

Tyto příklady by nás měly varovat. S pojmy menší a stejné mohutnosti musíme zacházet opatrně: nejprve je třeba ukázat, že mají vlastnosti, které od nich očekáváme. Každá množina je bijektivní sama se sebou; je-li f bijekce z A na B, je f^{-1} bijekce z B na A; a složení bijekcí je bijekce. Tedy relace $A \approx B$ je reflexivní, symetrická a transitivní, neboli je ekvivalencí mezi množinami.

Podobně bychom rádi dokázali, že $A \preceq B$ je (třídové) uspořádání. Reflexivita a transitivita je zřejmá. Slabá antisymetrie je zaručena následující větou.

3.8.3 Věta (Cantor, Bernstein)**.** *Pokud existuje prosté zobrazení z A do B, a zároveň existuje prosté zobrazení z B do A, pak existuje bijekce mezi A a B. Jinými slovy, je-li $A \preceq B$ a $B \preceq A$, je $A \approx B$.*

Důkaz. Buďte $f : A \to B$ a $g : B \to A$ prosté funkce. Potom složená funkce $g \circ f$ je prosté zobrazení A na $g[f[A]] \subseteq g[B] \subseteq A$. Bez újmy na obecnosti tedy stačí dokázat následující: je-li $A_0 \supseteq B_0 \supseteq A_1$ a $A_0 \approx A_1$, pak je také $A_0 \approx B_0$.

Buď tedy $f : A_0 \approx A_1$ bijekce a $A_0 \supseteq B_0 \supseteq A_1$. Položme dále

$$A_{n+1} = f[A_n], \quad B_{n+1} = f[B_n].$$

Potom pro každé $n \in \omega$ jsou množiny $A_n \setminus B_n$ a $A_{n+1} \setminus B_{n+1}$ navzájem disjunktní a bijektivní; podobně $B_n \setminus A_{n+1}$ jsou disjunktní navzájem i se všemi $A_n \setminus B_n$. Položíme-li tedy $g(x) = f(x)$ pro $x \in \bigcup_{n\in\omega}(A_n \setminus B_n)$ a $g(x) = x$ jinak, je tím definováno zobrazení $g : A_0 \to B_0$. Snadno se ověří, že g je bijekce. □

3.8.4 Definice. Množina, která je bijektivní s nějakým přirozeným číslem, je *konečná*; v opačném případě je *nekonečná*. Množina, která je bijektivní s množinou přirozených čísel, je *spočetná*; v opačném případě je *nespočetná*.

3.8.5 Příklad. (a) Isolovaný ordinál $\omega + 1$ je spočetná množina. Stačí položit $f(\omega) = 0$ a $f(n) = n + 1$ pro $n \in \omega$ a máme bijekci mezi $\omega \cup \{\omega\}$ a jeho ordinálním předchůdcem ω. Podobně se ukáže, že každý ordinální následník spočetného ordinálu je spočetný, takže všechny ordinály $\omega + n$ jsou spočetné. (b) Limitní ordinální číslo $\omega + \omega = 2\omega$ je rovněž spočetné, stačí poslat sudé $2n$ na $n \in \omega$ a liché $2n + 1$ na $\omega + n$. Podobně se nahlédne, že také 3ω a obecně každý ordinál tvaru $k\omega$ je spočetný. (c) Prosté zobrazení z ω do $\omega \times \omega$ najdeme triviálně, a $g(m, n) = 2^m \cdot 3^n$ je prosté zobrazení z $\omega \times \omega$ do ω. Podle Cantor-Bernsteinovy věty je tedy $\omega \times \omega$ spočetná množina. Indukcí se nyní snadno předvede, že všechny ordinály tvaru ω^k jsou spočetné. (d) $\mathbb{Z}$ a $\mathbb{Q}$ jsou spočetné.

3.8.6 Věta. *Sjednocení konečně mnoha konečných množin je konečné. Sjednocení spočetně mnoha nanejvýš spočetných množin je spočetné.*

3.8.7 Příklad. (a) Všechny množiny ω^k jsou spočetné, tedy i jejich spočetné sjednocení $\omega^{<\omega} = \bigcup_{k\in\omega} \omega^k$ je spočetná množina. Existuje tedy jen spočetně mnoho konečných posloupností přirozených čísel.

(b) Je-li A konečná nebo nejvýše spočetná *abeceda*, pak každá konečná posloupnost prvků z A je *slovo* nad A. *Jazyk* je potom jakákoli množina takových slov. Každý jazyk nad spočetnou abecedou je tedy spočetný.

(c) Polynomů daného stupně n s celočíselnými koeficienty je právě tolik jako n-tic celých čísel (koeficientů), tedy spočetně mnoho. Pro každý stupeň $n \in \omega$ je tedy množina $\mathcal{P}_n$ všech polynomů s celočíselnými koeficienty spočetná, takže i množina $\mathcal{P} = \bigcup \mathcal{P}_n$ polynomů všech stupňů je spočetná.

(d) Reálné číslo, které je kořenem nějakého reálného polynomu s celočíselnými koeficienty, je *algebraické*; jinak je *transcendentní*. Zřejmě každé racionální číslo je algebraické. Číslo $\sqrt{2}$ je jak známo iracionální, ale je algebraické, neboť je kořenem polynomu $x^2 - 2$. Podle (c) je všech polynomů s celočíselnými koeficienty jen spočetně mnoho, a podle známé věty z algebry má každý z nich jen konečně mnoho kořenů, totiž nejvýše tolik, kolik je jeho stupeň. Tedy algebraických čísel je jen spočetně mnoho.

3.8.8 Věta (Cantor)**.** *Pro každou množinu x je $x \prec P(x)$.*

Důkaz. Zobrazení svědčící o $x \preceq P(x)$ najdeme snadno, například $y \mapsto \{y\}$. Zbývá ukázat, že neexistuje žádné zobrazení množiny x *na* potenci $P(x)$. Dokážeme to sporem: buď $f : x \to P(x)$ takové zobrazení. V tom případě položme $z = \{y \in x; y \notin f(y)\} \subseteq x$. Podle předpokladu je $z = f(y)$ pro nějaké $y \in x$; ptejme se, jestli je $y \in f(y)$. Pokud $y \in f(y) = z$, je z definice $y \notin f(y)$. Pokud $y \notin f(y)$, je z definice $y \in z = f(y)$. V každém případě dostáváme spor. □

Kardinální čísla Pomocí prostých zobrazení můžeme porovnávat mohutnosti množin, aniž bychom tyto mohutnost vyjadřovali nějakým číslem udávajícím „počet prvků.“ Je užitečné si uvědomit, že pojem stejné mohutnosti je základnější než pojem čísla, které tuto mohutnost reprezentuje. Zavedeme nyní kardinální čísla jako reprezentanty mohutností.

3.8.9 Definice. Ordinální číslo α je *kardinál*, pokud není bijektivní s žádným ordinálem $\beta < \alpha$. Pro množinu x je $|x| = \min\{\alpha \in On; x \approx \alpha\}$ její *mohutnost*.

Pracujeme v teorii ZFC, takže každá množina se dá dobře uspořádat, a je tedy bijektivní s nějakým ordinálem. To znamená, že mohutnost $|x|$ je dobře definovaná pro všechny množiny. Navzájem bijektivní množiny $x \approx y$ mají z definice stejnou mohutnost $|x| = |y|$. V tomto smyslu jsou kardinální čísla reprezentanti mohutností. Kardinály budeme obvykle značit písmeny $\kappa, \lambda, \mu, \ldots$ z prostředku řecké abecedy, a Cn bude značit třídu všech kardinálních čísel.

3.8.10 Lemma. *(i) Každé přirozené číslo je kardinál. (ii) Je-li A množina kardinálů, je také $\bigcup A$ kardinál. (iii) Ordinální číslo ω je kardinál. (iv) Nekonečné kardinální číslo je nutně limitní ordinál.*

Důkaz. (i) Pro $n = 0$ není co dokazovat. Dále pokračujeme indukcí: je-li $n \in \omega$ kardinál, je také $n \cup \{n\}$ kardinál. Pokud ne, pak pro nějaké $m < n$ existuje bijekce $f : m \cup \{m\} \approx n \cup \{n\}$. Je-li $f(m) = n$, máme $f \restriction m : m \approx n$, spor. Je-li $f(m) = j < n$, musí být $f(i) = n$ pro nějaké $i < m$. Pak ale stačí položit $g(i) = j$ a $g(k) = k$ pro ostatní $k < m$ a máme bijekci $g : m \approx n$, spor.

(ii) Buď A množina kardinálů. Víme, že $\bigcup A$ je ordinál, supremum množiny A v dobrém uspořádání třídy On. Pokud $\bigcup A$ není kardinál, pak existuje bijekce $f : \beta \approx \bigcup A$ pro nějaké $\beta < \bigcup A$. V tom případě je $\beta < \alpha$ pro nějaké $\alpha \in A$, neboť $\bigcup A = \sup A$. Podle Cantor-Bernsteinovy věty je potom $\beta \approx \alpha$, spor.

(iii) plyne z (i) a (ii), neboť $\omega = \sup \omega$ je supremum množiny kardinálů.

(iv) Buď $\alpha \cup \{\alpha\}$ nekonečný kardinál; je tedy $\alpha \geq \omega$. Položme $f(\alpha) = 0$, $f(n) = n + 1$ pro $n \in \omega$, a $f(\beta)$ pro $\omega \leq \beta < \alpha$. Snadno se ověří, že f je bijekce $\alpha \cup \{\alpha\}$ na α, spor. Kardinál tedy nemůže být ordinálním následníkem. □

Uvědomme si rozdíl mezi ideou ordinálního a kardinálního čísla. Ordinální čísla jsou reprezentanti dobrých uspořádání (viz věta 3.5.10), kdežto kardinální čísla jsou reprezentanti velikostí. Například spočetnou množinu lze uspořádat podle typu ω nebo $\omega + \omega$ nebo jiného spočetného ordinálu, což jsou navzájem různá uspořádání, jejich mohutnost je však stejná, totiž spočetná.

Na přirozených číslech, tj. konečných kardinálech, představa mohutnosti a dobrého uspořádání splývají. Podle 3.7.13 je lineární uspořádání dobré právě tehdy, když neobsahuje nekonečnou klesající posloupnost. Z toho plyne, že každé lineární uspořádání konečné množiny je dobré. Podle 3.5.10 jsou tedy všechna lineární uspořádání konečné množiny navzájem isomorfní. To znamená, že přirozené číslo $n \in \omega$ lze lineárně uspořádat jediným způsobem, totiž podle ordinálního typu n, který je zároveň jeho mohutností.

3.8.11 Cvičení. Pro funkci $f : X \to X$ z konečné množiny X do sebe jsou následující podmínky ekvivalentní: (i) f je prostá (ii) f je na (iii) f je bijekce.

3.8.12 Cvičení. (a) Popište nějakou bijekci z $\omega \times \omega$ na ω. (b) Pro bijekci $f(x) = x/3$ z $[0, 1]$ na $[0, 1/3] \subseteq [0, 1/2) \subseteq [0, 1]$ popište bijekci $g : [0, 1] \approx [0, 1/2)$ tak jak je sestrojena v důkaze Cantorovy-Bernsteinovy věty a nakreslete její graf.

3.8.13 Cvičení. Triviálně platí $2^\omega \preceq \omega^\omega$. Ukažte, že platí i opačná nerovnost. (Návod: funkční hodnoty z ω lze binárně kódovat hodnotami z $\{0, 1\}$.)

3.8.14 Cvičení. (a) Každý systém neprázdných, otevřených, navzájem disjunktních intervalů v $\mathbb{R}$ je nanejvýš spočetný. (b) Každá monotónní reálná funkce má nanejvýš spočetně mnoho bodů nespojitosti. (c) Každá reálná funkce má nanejvýš spočetně mnoho ostrých lokálních extrémů.

3.8.15 Cvičení. (a) Množina X je T-*konečná* právě když každý neprázdný systém $\mathcal{A} \subseteq P(X)$ obsahuje maximální prvek; V opačném případě je T-*nekonečná*. Ukažte, že každé přirozené číslo je T-konečná množina; kardinální číslo ω je T-nekonečná množina; každá konečná množina je T-konečná; každá nekonečná množina je T-nekonečná. (b) Množina je D-*konečná*, pokud není bijektivní s žádnou svou vlastní částí. Ukažte, že množina je konečná právě tehdy, když je D-konečná. To jsou dva způsoby, jak zavést pojem konečnosti bez pojmu čísla.

3.8.16 Cvičení. Popište nějakou množinu $A \subset \mathbb{R}$, pro kterou je $\mathbb{N} \prec A \prec \mathbb{R}$ — nebo ukažte, že žádná taková množina neexistuje.

3.9 Filtry a ideály

3.9.1 Definice. Buď X neprázdná. Neprázdný systém $\mathcal{F} \subset P(X)$ je *filtr* na X, pokud pro $A, B \in \mathcal{F}$ je $A \cap B \in \mathcal{F}$ a pro $A \in \mathcal{F}, A \subseteq B$ je $B \in \mathcal{F}$. Jinými slovy, filtr na X je horní, dolů usměrněná množina v uspořádání $(P(X), \subseteq)$.

Filtr je způsob, jak zachytit představu „velké“ podmnožiny: celá množina je velká, prázdná nikoli; průnik dvou velkých množin je stále velký, a množina větší než nějaká velká je sama velká. Volbou různých filtrů pak zkoumáme různé pojmy „velikosti“ podmnožin. Definice vylučuje nezajímavý případ $\mathcal{F} = P(X)$. Uvažujeme pouze *vlastní* filtry, splňující $\emptyset \notin \mathcal{F}$.

3.9.2 Příklad. (a) Pro pevně zvolenou $A \subseteq X$ je $\mathcal{F} = \{B \subseteq X; A \subseteq B\}$ filtr na X. Filtry tohoto typu jsou *hlavní filtry*. Krajními případy jsou *triviální filtr* $\{X\}$ a filtry tvaru $\{B \subseteq X; x \in B\}$ pro nějaké $x \in X$.

(b) Systém $\mathcal{F} = \{A \subseteq \mathbb{N}; \mathbb{N} \setminus A$ je konečná$\}$ je netriviálním filtrem na $\mathbb{N}$, nazývá se *Fréchetův filtr*. Vskutku, $\mathcal{F}$ není tvaru $\{B \subseteq \mathbb{N}; A \subseteq B\}$ pro žádnou $A \subseteq N$: je-li $A \in \mathcal{F}$, pak také $B = A \setminus \{\min A\} \in \mathcal{F}$, přitom $A \not\subseteq B$.

(c) Buď $\mathcal{F}$ filtr na $\mathbb{N}$. Řekneme, že číslo $a \in \mathbb{R}$ je *$\mathcal{F}$-limitou* posloupnosti (a_n), pokud pro každé $\varepsilon > 0$ je $\{n \in \mathbb{N}; |a_n - a| < \varepsilon\} \in \mathcal{F}$. Klasický pojem limity je pak právě $\mathcal{F}$-limita podle Fréchetova filtru.

(d) Systém množin $A \subseteq \mathbb{R}$, které obsahují číslo 0 i s nějakým otevřeným intervalem, tvoří filtr na $\mathbb{R}$, který není hlavní: žádná A z filtru není podmnožinou všech ostatních. Tento filtr se nazývá *filtr okolí nuly*. Obecněji v každém topologickém prostoru X tvoří okolí zvoleného bodu filtr na X.

3.9.3 Definice. Buď X nějaká neprázdná množina. Systém $\mathcal{I} \subset P(X)$ je *ideál* na X, pokud pro $A, B \in \mathcal{I}$ je také $A \cup B \in \mathcal{I}$ a pro $A \subseteq B \in \mathcal{I}$ je také $A \in \mathcal{I}$. Jinými slovy, ideál je nahoru usměrněná dolní množina v $(P(X), \subseteq)$.

Pojem ideálu je duální k pojmu filtru: tak jako filtr zachycuje představu o „velkých“ podmnožinách, zachycuje ideál představu „malých“ podmnožin. Například konečné podmnožiny dané množiny X tvoří ideál $Fin(X)$, nebo prostě Fin v případě $Fin(\mathbb{N})$. Ideál Fin je duální k Fréchetovu filtru.

3.9.4 Cvičení. $\mathcal{F}$ je filtr na X právě když $\mathcal{F}_* = \{X \setminus F; F \in \mathcal{F}\}$ je ideál, a naopak $\mathcal{I}$ je ideál na X právě když $\mathcal{I}^* = \{X \setminus I; I \in \mathcal{I}\}$ je filtr. Takové filtry a ideály jsou navzájem *duální*.

3.9.5 Příklad. Harmonická řada $\sum \left\{\frac{1}{n}; n \in \mathbb{N}\right\}$ jak známo diverguje. *Sumační ideál* $\mathcal{I}_{\frac{1}{n}}$ na $P(\mathbb{N})$ sestává z těch množin $A \subseteq \mathbb{N}$, pro které $\sum \left\{\frac{1}{n}; n \in A\right\}$ konverguje. Ideál $\mathcal{I}_{\frac{1}{n}} \supset Fin$ rozšiřuje Fréchetův ideál o některé nekonečné množiny.

3.9.6 Cvičení. Pro množinu $A \subseteq \mathbb{N}$ uvažme posloupnost $|A \cap \{0, 1, \ldots, n\}|/n$. To je posloupnost reálných čísel z intervalu $[0, 1]$. Má-li tato posloupnost limitu $d(A)$, řekneme, že množina A má *hustotu* $d(A)$. Například každá konečná množina má nulovou hustotu, množina sudých čísel má hustotu $1/2$, a každá nekonečná aritmetická posloupnost s diferencí k má hustotu $1/k$. Existují i množiny, které hustotu nemají. Uvažte systém $\mathcal{Z}$ všech podmnožin $A \subseteq \mathbb{N}$ s nulovou hustotou. Ukažte, že $\mathcal{Z}$ je ideál na $P(\mathbb{N})$. Patří množina prvočísel do $\mathcal{Z}$?

3.9.7 Cvičení. Uvažte systém $\mathcal{W}$ všech podmnožin $A \subseteq \mathbb{N}$, které obsahují jen aritmetické posloupnosti omezené délky. Například množina $\{2^n; n \in \mathbb{N}\}$ obsahuje jen aritmetické posloupnosti délky 2, a padne tedy do $\mathcal{W}$. Naopak množina sudých čísel obsahuje aritmetické posloupnosti libovolné konečné délky. Dá se ukázat, že systém $\mathcal{W}$ je ideál na $P(\mathbb{N})$: je zřejmě uzavřen na podmnožiny, obsahuje prázdnou množinu, a neobsahuje $\mathbb{N}$; uzavřenost na sjednocení je tvrzení van der Waerdenovy věty z algebry. Podle Szemerediho věty je $\mathcal{W} \subseteq \mathcal{Z}$, tedy každá množina s nenulovou hustotou obsahuje aritmetickou posloupnost libovolné konečné délky. Opačná inkluze neplatí, jak ukazuje již množina $\bigcup [n^3, n^3 + n]$. Patří množina prvočísel do $\mathcal{W}$?

Kapitola 4

Booleovy algebry

Booleovy algebry zavedl G. Boole[1] v půli devatenáctého století jako algebraickou strukturu vhodnou pro matematický popis chování výrokových spojek. Postupně se ukázalo, že Booleovy algebry jsou relevantní nejen pro logiku, ale že se přirozeně vyskytují v teorii množin, topologii, informatice, analýze, teorii míry a jinde. Základní literaturou pro hlubší studium Booleových algeber je [HBA].

4.1 Booleovské operace

4.1.1 Definice. *Booleova algebra* je struktura $\mathbb{B} = (B, \wedge, \vee, -, \mathbf{0}, \mathbf{1})$, kde B je neprázdná množina, opatřená binárními operacemi $\wedge$ (*průsek*) a $\vee$ (*spojení*), unární operací $-$ (*komplement*) a konstantami $\mathbf{0}$ a $\mathbf{1}$, přičemž platí:

$$\begin{aligned} x \wedge \mathbf{1} &= x & x \vee \mathbf{0} &= x \\ x \wedge -x &= \mathbf{0} & x \vee -x &= \mathbf{1} \\ x \wedge y &= y \wedge x & x \vee y &= y \vee x \\ x \wedge (y \wedge z) &= (x \wedge y) \wedge z & x \vee (y \vee z) &= (x \vee y) \vee z \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) & x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \end{aligned}$$

Operace průseku a spojení jsou komutativní, asociativní, navzájem distributivní, a prvky $\mathbf{0}$ a $\mathbf{1}$ jsou vůči operacím $\vee$ a $\wedge$ neutrální. Díky asociativitě a komutativitě můžeme pro konečnou $\{x_1, \ldots, x_n\} \subseteq \mathbb{B}$ psát stručně $x_1 \wedge \ldots \wedge x_n$ a $x_1 \vee \ldots \vee x_n$ bez závorek a bez ohledu na pořadí. Místo $x \wedge -y$ budeme psát stručně $x - y$. Množinu $\mathbb{B} \setminus \{\mathbf{0}\}$ nenulových prvků algebry $\mathbb{B}$ budeme značit $\mathbb{B}^+$.

Jednoprvková množina s operacemi definovanými jediným možným způsobem zřejmě splňuje všechny výše uvedené axiomy, je však zcela nezajímavá. Budeme uvažovat jen *nedegenerované* algebry, splňující $\mathbf{0} \neq \mathbf{1}$.

4.1.2 Příklad. Nejjednodušší Booleovou algebrou je dvouprvková množina $\{0, 1\}$, kde operace se chovají jako výrokové spojky, hledíme-li na 0 a 1 jako na pravdivostní hodnoty. Tuto algebru budeme označovat stručně 2.

[1] G. Boole, *The mathematical analysis of logic*, Cambridge, 1847

Množina 2^n tvořená všemi n-bitovými posloupnostmi je Booleovou algebrou vůči operacím definovaným *po složkách*: v posloupnosti $x \wedge y \in 2^n$ stojí na i-tém místě hodnota $x_i \wedge y_i$, a podobně definujeme i ostatní operace. Tedy například v algebře 2^5 je $10110 \wedge 01010 = 00010$, $10110 \vee 01010 = 11111$ a $-10110 = 01001$. Snadno se ověří, že tato struktura splňuje Booleovské axiomy. Existují tedy Booleovy algebry neomezeně velkých konečných mohutností. Později ukážeme, že konečné Booleovy algebry jsou právě tvaru 2^n.

4.1.3 Příklad. (a) Je-li X neprázdná množina, pak množina $P(X)$ všech jejích podmnožin tvoří vůči obvyklým množinovým operacím průniku, sjednocení a doplňku Booleovu algebru, takzvanou *potenční algebru* množiny X.

(b) Buď X libovolná neprázdná množina. Potom systém všech těch podmnožin množiny X, které jsou buďto konečné, nebo mají konečný doplněk, tvoří vůči obvyklým množinovým operacím Booleovu algebru. Existují tedy Booleovy algebry všech nekonečných mohutností.

(c) Je-li $(X, \leq)$ lineární uspořádání, uvažme *polouzavřené* intervaly tvarů $[x, y)$ a $[x, \rightarrow)$ a $(\leftarrow, y)$. Systém všech konečných sjednocení takových intervalů tvoří spolu s množinovými operacemi *intervalovou algebru* lineárního uspořádání $(X, \leq)$, kterou budeme značit $Int(X, \leq)$.

Algebry z předchozího příkladu sestávají z podmnožin nějaké předem dané množiny, a operace na nich jsou obvyklé množinové operace. Takové algebry nazýváme *algebry množin*. Později dokážeme Stoneovu větu o reprezentaci, podle které je každá Booleova algebra isomorfní s nějakou algebrou množin.

4.1.4 Příklad. Buď X topologický prostor. (a) Množina $U \subseteq X$ je *obojetná*, je-li současně otevřená i uzavřená. Systém $CO(X)$ všech obojetných podmnožin pak tvoří algebru množin. Tato algebra je triviální právě když prostor X je souvislý. (b) Podmnožina $U \subseteq X$ je *regulární otevřená*, pokud je vnitřkem svého uzávěru. Soubor $RO(X)$ všech regulárních otevřených množin pak tvoří Booleovu algebru, definujeme-li operace následovně. Spojením $U \vee V$ je vnitřek uzávěru množiny $U \cup V$; průsekem $U \wedge V$ je průnik $U \cap V$; Booleovským doplňkem U je vnitřek množiny $X \setminus U$. Algebra $RO(X)$ není algebrou množin, neboť Booleovské operace nesplývají s množinovými operacemi.

4.1.5 Příklad. Buď $\mathcal{A}$ množina výrokových proměnných. Pro výrokovou formuli φ nad $\mathcal{A}$ buď $[\varphi] = \{\varrho; \vdash \varphi \leftrightarrow \varrho\}$ její *ekvivalenční třída*. Na množině těchto ekvivalenčních tříd definujeme Booleovské operace následovně:

$$\mathbf{0} = [\bot], \mathbf{1} = [\top], [\varphi] \wedge [\psi] = [\varphi \wedge \psi], [\varphi] \vee [\psi] = [\varphi \vee \psi], -[\varphi] = [\neg\varphi]$$

Přísně vzato se dopouštíme nejasnosti ve značení: na levé straně uvedených definic značí symboly $\wedge$ a $\vee$ jistou nově zavedenou binární operaci, vpravo pak výrokovou spojku. Tento slang je však běžný, a zdůrazňuje souvislost mezi chováním výrokových spojek a Booleovských operací.

Booleovské operace na ekvivalenčních třídách jsou definovány pomocí zvolených reprezentantů, ale snadno se přesvědčíme, že výsledek operace na volbě reprezentantů nezávisí: pro $[\varphi] = [\varphi']$ je $-[\varphi] = [\neg\varphi] = [\neg\varphi'] = -[\varphi']$ apod.

Ověřit nyní, že tato struktura je vskutku Booleovou algebrou, znamená ověřit jisté vlastnosti výrokových spojek, které jsou však čtenáři již známé. Tuto algebru budeme nazývat *algebra výroků* nad $\mathcal{A}$ a značit ji $\mathbb{B}(\mathcal{A})$. Pro konečnou $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ se jedná o *konečnou algebru výroků* — z 1.2.5 víme, že má

právě 2^{2^n} prvků. Pro $\mathcal{A}$ nekonečnou je $\mathbb{B}(\mathcal{A})$ nekonečná algebra mohutnosti $|\mathcal{A}|$.

4.1.6 Příklad. Pro uzavřenou formuli φ jazyka $\mathcal{L}$ predikátové logiky označme její ekvivaleční třídu jako $[\varphi] = \{\varrho; \vdash \varphi \leftrightarrow \varrho\}$. Ekvivalenční třídy $[\varphi]$ pak tvoří *Lindenbaumovu algebru* $\mathbb{B}(\mathcal{L})$ jazyka $\mathcal{L}$, s operacemi definovanými jako výše. Obecněji, je-li T teorie v jazyce $\mathcal{L}$, buď $[\varphi] = \{\psi; T \vdash \varphi \leftrightarrow \psi\}$ ekvivalenční třída sentence φ jazyka $\mathcal{L}$. Třídy $[\varphi]$ pak tvoří *Lindenbaumovu algebru* $\mathbb{B}(T)$ teorie T. Algebra $\mathbb{B}(T)$ je triviální právě tehdy, když teorie T je úplná.

Axiomy Booleovy algebry vykazují určitou dualitu: pokud v nějakém axiomu zaměníme všechny výskyty $\vee$ a $\wedge$, a všechny výskyty $\mathbf{0}$ a $\mathbf{1}$, vznikne opět axiom. To znamená, že je-li $(B, \wedge, \vee, -, \mathbf{0}, \mathbf{1})$ Booleova algebra, je $(B, \vee, \wedge, -, \mathbf{1}, \mathbf{0})$ rovněž Booleova algebra. Budeme vždy dokazovat jen jedno z duálních tvrzení.

4.1.7 Lemma. *V každé Booleově algebře platí*

(a) $x \wedge \mathbf{0} = \mathbf{0}$, $x \vee \mathbf{1} = \mathbf{1}$

(b) $x \wedge x = x$, $x \vee x = x$

(c) $x \wedge (x \vee y) = x$, $x \vee (x \wedge y) = x$

(d) $x \wedge y = x$ *právě tehdy, když* $x \vee y = y$

(e) $x = -y$ *právě tehdy, když* $x \vee y = \mathbf{1}$ *a* $x \wedge y = \mathbf{0}$

(f) $-(x \wedge y) = -x \vee -y$, $-(x \vee y) = -x \wedge -y$

(g) $-(-x) = x$, $-\mathbf{0} = \mathbf{1}$, $-\mathbf{1} = \mathbf{0}$

Vlastnosti (a), (b), (c) se nazývají *anihilace*, *idempotence* a *absorpce*. Vlastnost (e) charakterizuje komplementy a vlastnosti (f) jsou *de Morganovy formule*.

Důkaz. (a) $x \wedge \mathbf{0} = (x \wedge \mathbf{0}) \vee \mathbf{0} = (x \wedge \mathbf{0}) \vee (x \wedge -x) = x \wedge (\mathbf{0} \vee -x) = x \wedge -x = \mathbf{0}$. (b) $x = x \wedge \mathbf{1} = x \wedge (x \vee -x) = (x \wedge x) \vee (x \wedge -x) = (x \wedge x) \vee \mathbf{0} = x \wedge x$. (c) $x = x \wedge \mathbf{1} = x \wedge (\mathbf{1} \vee y) = (x \wedge \mathbf{1}) \vee (x \wedge y) = x \vee (x \wedge y)$. (d) Je-li $x \wedge y = x$, je $x \vee y = (x \wedge y) \vee y = y$. (e) Jeden směr plyne přímo z axiomů. V opačném směru máme $x = x \wedge \mathbf{1} = x \wedge (y \vee -y) = (x \wedge y) \vee (x \wedge -y) = \mathbf{0} \vee (x \wedge -y) = (y \wedge -y) \vee (x \wedge -y) = (y \vee x) \wedge -y = \mathbf{1} \wedge -y = -y$. (f) Podle (e) stačí ukázat, že $(x \vee y) \vee (-x \wedge -y) = (x \vee y \vee -x) \wedge (x \vee y \vee -y) = \mathbf{1} \wedge \mathbf{1} = \mathbf{1}$ a $(x \vee y) \wedge (-x \wedge -y) = (x \wedge -x \wedge -y) \vee (y \wedge -x \wedge -y) = \mathbf{0} \vee \mathbf{0} = \mathbf{0}$. □

4.1.8 Cvičení. Proč neexistuje žádná tříprvková Booleova algebra?

4.1.9 Cvičení. Množina $A \subseteq X$, kde X je $\mathbb{Z}$ nebo $\mathbb{Q}$ nebo $\mathbb{R}$, je *periodická*, pokud $A = \{a + p; a \in A\}$ pro nějakou *periodu* $p \in X$. Zřejmě X a $\emptyset$ jsou periodické (s libovolnou periodou); naopak žádná neprázdná konečná množina nemůže být periodická (s periodou $p \neq 0$). Dokažte, že pro jakékoli $p \in X$ tvoří systém všech periodických množin s periodou p Booleovu algebru vůči množinovým operacím. Popište tuto algebru explicitně pro $p = 0, 1, 2, 3$.

4.1.10 Cvičení. Přirozené číslo $m > 1$ je *square-free*, pokud se v jeho prvočíselném rozkladu žádné prvočíslo nevyskytuje s vyšší než první mocninou. To je právě když $\mathbb{B} = \{x \in \mathbb{N}; x \text{ dělí } m\}$ tvoří Booleovu algebru (*algebru dělitelů*), kde $\mathbf{0}$ je číslo 1, $\mathbf{1}$ je číslo m, komplement $-x$ je podíl m/x, $x \vee y$ je nejmenší společný násobek a $x \wedge y$ je největší společný dělitel.

4.1.11 Cvičení (Booleovské grupy)**.** Je-li $(\mathbb{B}, \wedge, \vee, -, \mathbf{0}, \mathbf{1})$ Booleova algebra, položme $x \triangle y = (x \wedge -y) \vee (-x \wedge y)$. Ukažte, že $(\mathbb{B}, \triangle, \mathbf{0})$ je komutativní grupa, ve které každý prvek je svým vlastním inverzem.

4.1.12 Cvičení (Booleovské okruhy)**.** Buď $(R, +, -, *, 0, 1)$ okruh. Prvek $x \in R$ je *idempotentní*, pokud $x \cdot x = x$. Okruh R je *Booleovský*, pokud všechny jeho prvky jsou idempotentní. (a) Buď $(R, +, -, *, 0, 1)$ Booleovský okruh. Pak R spolu s operacemi $x \wedge y = x * y$, $x \vee y = x + y + x * y$, $-x = 1 + x$ tvoří Booleovu algebru. (b) Je-li $(\mathbb{B}, \wedge, \vee, -, \mathbf{0}, \mathbf{1})$ Booleova algebra, je B spolu s operacemi $x + y = x \triangle y$ a $x * y = x \wedge y$ Booleovský okruh. (c) Každý Booleovský okruh je komutativní a má charakteristiku 2.

4.2 Uspořádání

Booleova algebra nese kromě algebraické struktury zároveň jisté kanonické uspořádání, přičemž jednu strukturu lze rekonstruovat z druhé. Na Booleovu algebru tedy můžeme ekvivalentně pohlížet jako speciální případ uspořádané množiny.

4.2.1 Definice. Pro $x, y \in \mathbb{B}$ položme $x \leq y$ právě tehdy, když $x \wedge y = x$.

Definice opět zobecňuje chování množin: pro množiny A, B je $A \subseteq B$ právě když $A \cap B = A$, což je právě když $A \cup B = B$. Ukážeme nejprve, že zavedená relace je vskutku uspořádáním množiny $\mathbb{B}$, prvek $\mathbf{0}$ je nejmenší a $\mathbf{1}$ je největší, spojení $x \vee y$ je supremem a průsek $x \wedge y$ je infimem dvouprvkové množiny $\{x, y\}$.

4.2.2 Lemma. *Pro každé prvky x, y, z každé Booleovy algebry platí*

(a) $x \leq x$*;*

(b) *je-li $x \leq y$ a $y \leq z$, je $x \leq z$;*

(c) *je-li $x \leq y$ a $y \leq x$, je $x = y$;*

(d) $\mathbf{0} \leq x \leq \mathbf{1}$*;*

(e) *je-li $x \leq z$ a $y \leq z$, je $x \vee y \leq z$;*

(f) *je-li $x \geq z$ a $y \geq z$, je $x \wedge y \geq z$.*

Důkaz. (b) Je-li $x \wedge y = x$ a $y \wedge z = y$, je $x \wedge z = (x \wedge y) \wedge z = x \wedge (y \wedge z) = x \wedge y = x$. (e) Je-li $x \leq z$ a $y \leq z$, je $x \vee z = z$ a $y \vee z = z$; tedy také $x \vee y \vee z = x \vee z = z$. □

Vlastnosti (e) a (f) výše se snadno rozšíří na libovolný konečný počet prvků, takže $x_1 \vee x_2 \vee \ldots \vee x_n$ je supremum a $x_1 \wedge x_2 \wedge \ldots \wedge x_n$ je infimum konečné množiny $\{x_1, \ldots, x_n\} \subseteq \mathbb{B}$ v uspořádání $(\mathbb{B}, \leq)$. Má-li $X \subseteq \mathbb{B}$ supremum resp. infimum v uspořádání $(\mathbb{B}, \leq)$, budeme je značit $\bigvee X$ resp. $\bigwedge X$.

Snadno se nahlédne, že uspořádáním každé algebry množin je inkluze, uspořádáním algebry formulí je vztah logického důsledku, a uspořádáním algebry dělitelů vztah dělitelnosti. Algebra $RO(X)$ regulárních otevřených množin je také uspořádána inkluzí, přestože není algebrou množin.

4.2.3 Definice. Řekneme, že dva nenulové prvky x, y Booleovy algebry $\mathbb{B}$ jsou navzájem *disjunktní*, a píšeme $x \perp y$, pokud $x \wedge y = \mathbf{0}$. V opačném případě řekneme, že jsou *kompatibilní*. Prvek $a \in \mathbb{B}^+$ je *atom*, pokud pod a neexistují dva navzájem disjunktní nenulové prvky. Pokud pod každým nenulovým prvkem $x \in \mathbb{B}^+$ existuje nějaký atom, řekneme, že algebra $\mathbb{B}$ je *atomární*. Pokud naopak žádné atomy nemá, řekneme, že je *bezatomární*.

4.2.4 Příklad. Každá konečná algebra je atomární, specielně 2^n je atomární. Potenční algebra $P(\mathbb{N})$ a intervalová algebra uspořádání $(\mathbb{Z}, \leq)$ jsou rovněž atomární, jejich atomy jsou právě jednoprvkové množiny. Naopak intervalová algebra reálné ani racionální přímky žádné atomy nemá. Intervalová algebra uspořádání $(-\infty, -2) \cup \{0, 1\} \cup (2, \infty)$ má atom, ale není atomární.

4.2.5 Cvičení. Pro každé dva prvky Booleovy algebry platí (a) $x \wedge y \leq x \leq x \vee y$; (b) $x \leq y$ právě když $x - y = \mathbf{0}$ (právě když $-x \vee y = \mathbf{1}$; právě když $-y \leq -x$); (c) $x < y$ právě když $x \leq y$ a $y - x \neq \mathbf{0}$.

4.2.6 Cvičení. Atomy konečné algebry výroků jsou právě mintermy úplného disjunktivního tvaru. Nekonečná algebra výroků je naopak bezatomární.

4.2.7 Cvičení. Pro $a \in \mathbb{B}^+$ jsou následující podmínky ekvivalentní: a je atom; a je minimálním prvkem $(\mathbb{B}^+, \leq)$; pro každé $x \in \mathbb{B}$ je buďto $a \leq x$ nebo $a \leq -x$; je-li $a \leq x \vee y$, je $a \leq x$ nebo $a \leq y$; je-li $a = x \vee y$, je $a = x$ nebo $a = y$.

4.2.8 Cvičení. Pro Booleovu algebru $\mathbb{B}$ jsou následující podmínky ekvivalentní: $\mathbb{B}$ je atomární; $\mathbf{1} = \bigvee \{a \in \mathbb{B}^+; a \text{ je atom}\}$; každé $x = \bigvee \{a \leq x; a \text{ je atom}\}$; $x \leq y$ právě když každý atom ležící pod x leží i pod y.

4.2.9 Cvičení. Uspořádaná množina $(X, \leq)$ je *svaz*, pokud každá dvouprvková podmnožina má supremum a infimum. Podle předchozího je tedy každá Booleova algebra svaz s nejmenším a největším prvkem. Je-li naopak $(X, \leq)$ svaz, pak pro $x, y \in X$ položme $x \vee y = \sup\{x, y\}$ a $x \wedge y = \inf\{x, y\}$. Pro uspořádání Booleovy algebry $(\mathbb{B}, \leq)$ se pak navíc jedná o *distributivní* a *komplementární* svaz: operace $\wedge$ a $\vee$ jsou navzájem distributivní, a ke každému prvku $x \in \mathbb{B}$ existuje *komplement*, tj. prvek $y \in \mathbb{B}$ takový, že $x \wedge y = \mathbf{0}$ a $x \vee y = \mathbf{1}$.

(a) Ukažte, že v distributivním svazu jsou komplementy nutně jednoznačné. (b) Ukažte, že každý komplementární distributivní svaz s nejmenším a největším prvkem nese strukturu Booleovy algebry, definujeme-li operace $\wedge$ a $\vee$ jako výše. (c) Uvažte algebru dělitelů z 4.1.10 jakožto uspořádání. Zjistěte, která z výše uvedených svazových vlastností selže, pokud uvažované číslo *není* square-free, tedy pokud jeho prvočíselný rozklad obsahuje mocninu nějakého prvočísla.

4.3 Podalgebry

4.3.1 Definice. Buď $(\mathbb{B}, \wedge, \vee, -, \mathbf{0}, \mathbf{1})$ Booleova algebra. Řekneme, že podmnožina $\mathbb{A} \subseteq \mathbb{B}$ je její *podalgebrou*, pokud je uzavřená na Booleovské operace. To znamená, že $\mathbf{0}, \mathbf{1} \in \mathbb{A}$, a pro každé $x, y \in \mathbb{A}$ je také $-x, x \wedge y, x \vee y \in \mathbb{A}$.

Každá Booleova algebra má triviální podalgebru $\{\mathbf{0}, \mathbf{1}\}$ a zároveň je sama svou podalgebrou. Některé Booleovské operace lze vyjádřit pomocí jiných,

takže v definici by stačilo požadovat uzavřenost na $-$ a $\wedge$, nebo duálně na $-$ a $\vee$, neboť $\mathbf{1} = x \vee -x$, $\mathbf{0} = -\mathbf{1}$ a $x \wedge y = -(-x \vee -y)$. Naopak uzavřenost na $\wedge, \vee, \mathbf{0}, \mathbf{1}$ nestačí, protože $\{\mathbf{0}, x, \mathbf{1}\} \subseteq \mathbb{B}$ není podalgebra. Laskavý čtenář si povšimne souvislosti s úplnými systémy výrokových spojek. Z uzavřenosti na operace plyne, že uspořádání podalgebry $\mathbb{A} \subseteq \mathbb{B}$ je právě uspořádání algebry $\mathbb{B}$ zúžené na $\mathbb{A}$.

Algebry množin jsou z definice právě podalgebry potenčních algeber; algebra konečných množin $A \subseteq \mathbb{N}$ a jejich doplňků je podalgebrou $P(\mathbb{N})$, algebra spočetných množin $A \subseteq \mathbb{R}$ a jejich doplňků je podalgebrou $P(\mathbb{R})$, intervalová algebra $Int(X, \leq)$ je podalgebrou $P(X)$, atd. Algebra $CO(X)$ obojetných množin prostoru X je podalgebrou algebry $RO(X)$ regulárních otevřených množin, a na této podalgebře splývají operace algebry $RO(X)$ s množinovými operacemi.

Jsou-li algebry $\mathbb{A}_i$ pro $i \in I$ podalgebrami nějaké předem dané algebry $\mathbb{B}$, pak také $\bigcap_{i \in I} \mathbb{A}_i$ je podalgebrou $\mathbb{B}$. Jinými slovy, průnik libovolného systému podalgeber tvoří opět podalgebru. To nás opravňuje k následující definici.

4.3.2 Definice. Buď $\mathbb{B}$ Booleova algebra a X její podmnožina. Pak podalgebra $\langle X \rangle = \bigcap \{\mathbb{A} \supseteq X; \mathbb{A} \text{ je podalgebra } \mathbb{B}\}$ je *generovaná* množinou X.

Algebra $\langle X \rangle$ je nejmenší podalgebra algebry $\mathbb{B}$, která obsahuje X jako podmnožinu: je průnikem všech takových. Například algebra konečných množin a jejich doplňků je právě podalgebra $P(\mathbb{N})$ generovaná všemi jednobodovými podmnožinami, a intervalová algebra $Int(X, \leq)$ je právě podalgebra potenční algebry $P(X)$ generovaná všemi polouzavřenými intervaly. Generování podalgebry je zřejmě monotónní operace: pro $X \subseteq Y \subseteq \mathbb{B}$ je $\langle X \rangle \subseteq \langle Y \rangle$.

Normální tvar Je-li $X \subseteq \mathbb{B}$, pak v *normálním tvaru* nad X jsou právě prvky algebry $\mathbb{B}$ tvaru $(x_1^1 \wedge \ldots \wedge x_1^{k_1}) \vee (x_2^1 \wedge \ldots \wedge x_2^{k_2}) \vee \ldots \vee (x_n^1 \wedge \ldots \wedge x_n^{k_n})$, kde pro každé x_i^j je $x_i^j \in X$ nebo $-x_i^j \in X$. Jinými slovy, v normálním tvaru nad X jsou právě konečná spojení konečných průseků prvků z X nebo jejich doplňků. Speciálně pro algebru výroků se jedná o disjunktivní tvar formulí nad X.

4.3.3 Věta. *Buď $\mathbb{B}$ Booleova algebra a $X \subseteq \mathbb{B}$ její podmnožina. Pak podalgebru $\langle X \rangle$ generovanou množinou X tvoří právě prvky v normálním tvaru nad X.*

Důkaz. Označme jako $\mathbb{A} \subseteq \mathbb{B}$ množinu všech prvků v normálním tvaru nad X. Pak je $X \subseteq \mathbb{A}$, přitom každý prvek v normálním tvaru nad X nutně leží v každé podalgebře obsahující X. Stačí tedy ukázat, že $\mathbb{A} \subseteq \mathbb{B}$ je podalgebra. Z definice je $\mathbb{A}$ uzavřená na spojení. Stejně tak je uzavřená na doplňky: je-li prvek algebry $\mathbb{B}$ v normálním tvaru nad X, pak i jeho doplněk lze pomocí deMorganových formulí a distribučního zákona zapsat v normálním tvaru. □

Duálně můžeme nyní ukázat, že prvky podalgebry $\langle X \rangle$ jsou právě tvaru $(x_1^1 \vee \ldots \vee x_1^{k_1}) \wedge (x_2^1 \vee \ldots \vee x_2^{k_2}) \wedge \ldots \wedge (x_n^1 \vee \ldots \vee x_n^{k_n})$, kde pro každé x_i^j je $x_i^j \in X$ nebo $-x_i^j \in X$. V algebře výroků pak jde o konjunktivní tvar. Vidíme, že právě dokázaná věta zobecňuje větu o normálním tvaru výrokových formulí.

Věta dává explicitní *vnitřní popis* prvků generované algebry, která je definována abstraktně jako průnik podalgeber. Jako důsledek získáváme, že podalgebra generovaná konečnou množinou mohutnosti $|X| = n$ má nejvýše 2^{2^n}

prvků a je tedy konečná.[2] V oddíle o volných algebrách uvidíme, že množina mohutnosti $|X| = n$ generuje algebru maximální možné mohutnosti 2^{2^n} právě když je nezávislá. Pro množinu nekonečné mohutnosti $|X|$ je $|\langle X\rangle| = |X|$, takže nekonečná algebra $\mathbb{B}$ má podalgebry všech nekonečných mohutností $\leq |\mathbb{B}|$.

4.3.4 Cvičení. Uvažte množiny $2\mathbb{N} = \{2n; n \in \mathbb{N}\}, 3\mathbb{N} = \{3n; n \in \mathbb{N}\}$ a další jakožto prvky potenční algebry $P(\mathbb{N})$. Popište explicitně prvky generovaných podalgeber $\langle\{2\mathbb{N}\}\rangle$, $\langle\{2\mathbb{N}, 3\mathbb{N}\}\rangle$, $\langle\{2\mathbb{N}, 4\mathbb{N}\}\rangle$, $\langle\{2\mathbb{N}, 3\mathbb{N}, 4\mathbb{N}\}\rangle$.

4.3.5 Cvičení. Je-li $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{B}$ konečná, jsou atomy konečné algebry $\langle X\rangle$ právě tvaru $y_1 \wedge \ldots \wedge y_n$, kde každé y_i je buďto $x_i \in X$ nebo $-x_i$.

4.3.6 Cvičení. Buď $\mathbb{B}$ Booleova algebra, buď $\mathbb{A}$ její podalgebra, a buď $b \in \mathbb{B}\setminus\mathbb{A}$. Pak prvky algebry $\langle\mathbb{A} \cup \{b\}\rangle$ jsou právě tvaru $(a_1 \wedge b) \vee (a_2 - b)$, kde $a_1, a_2 \in \mathbb{A}$.

4.4 Morfismy

4.4.1 Definice. Buďte $\mathbb{A}$ a $\mathbb{B}$ Booleovy algebry. Pak zobrazení $f : \mathbb{A} \to \mathbb{B}$ je *homomorfismus*, pokud $f(\mathbf{0}_\mathbb{A}) = \mathbf{0}_\mathbb{B}$, $f(\mathbf{1}_\mathbb{A}) = \mathbf{1}_\mathbb{B}$, a pro každé $x, y \in \mathbb{A}$ je $f(-x) = -f(x)$, $f(x \vee y) = f(x) \vee f(y)$ a $f(x \wedge y) = f(x) \wedge f(y)$.

Jinými slovy, homomorfismus je zobrazení, které zachovává Booleovské operace. Jelikož se jedná o dvě různé algebry, měli bychom přesněji psát $f(\mathbf{0}_\mathbb{A}) = \mathbf{0}_\mathbb{B}$, $f(x \wedge^\mathbb{A} y) = f(x) \wedge^\mathbb{B} f(y)$ a podobně; nebude-li však hrozit nedorozumění, nebudeme takové značení používat.

4.4.2 Příklad. Pravdivostní ohodnocení výrokových formulí nad výroky z $\mathcal{A}$ je právě homomorfismus z algebry $\mathbb{B}(\mathcal{A})$ do algebry $2 = \{0, 1\}$. Vskutku, respektovat operace na $\mathbb{B}(\mathcal{A})$, totiž výrokové spojky, znamená právě přiřazovat negacím, konjunkcím a disjunkcím patřičné hodnoty[3] z $\{0, 1\}$.

4.4.3 Lemma. *Pro zobrazení $f : \mathbb{A} \to \mathbb{B}$ jsou následující podmínky ekvivalentní:*

(a) *f je homomorfismus;*

(b) *pro každé $x, y \in \mathbb{A}$ je $f(-x) = -f(x)$ a $f(x \vee y) = f(x) \vee f(y)$;*

(c) *pro každé $x, y \in \mathbb{A}$ je $f(-x) = -f(x)$ a $f(x \wedge y) = f(x) \wedge f(y)$;*

(d) *$f(\mathbf{1}) = \mathbf{1}$, $f(x \vee y) = f(x) \vee f(y)$, a pro $x \perp y$ je $f(x) \perp f(y)$.*

Důkaz. Implikace z (a) do (b) platí z definice. Tvrzení (b) a (c) jsou ekvivalentní díky de Morganovým formulím. Pokud platí (b), máme $f(\mathbf{1}) = f(x \vee -x) = f(x) \vee -f(x) = \mathbf{1}$, a podobně z (c) plyne $f(\mathbf{0}) = \mathbf{0}$. Tedy (a),(b),(c) jsou ekvivalentní. Implikace z (a) do (d) platí z definice. Pokud platí (d), pak pro každé $x \in \mathbb{B}$ máme $f(x) \wedge f(-x) = \mathbf{0}$ a $f(x) \vee f(-x) = f(x \vee -x) = f(\mathbf{1}) = \mathbf{1}$, takže $f(x)$ je právě komplement $f(-x)$ a máme (b). □

[2] To není samozřejmé: například nekonečnou grupu $(\mathbb{Z}, +)$ generuje jediný prvek $1 \in \mathbb{Z}$.

[3] Nabízí se prozkoumat zobecnění výrokové logiky, ve kterém pravdivostními ohodnoceními jsou homomorfismy z $\mathbb{B}(\mathcal{A})$ do *nějaké* Booleovy algebry, jejíž prvky budou hrát roli pravdivostních hodnot. Klasická výroková logika se v tomto světle jeví jako nejjednodušší možná.

4.4.4 Definice. Prostý homomorfismus z algebry $\mathbb{A}$ na algebru $\mathbb{B}$ je *isomorfismus* mezi algebrami $\mathbb{A}$ a $\mathbb{B}$. Pokud takový isomorfismus existuje, řekneme, že algebry $\mathbb{A}$ a $\mathbb{B}$ jsou navzájem *isomorfní*, a píšeme $\mathbb{A} \simeq \mathbb{B}$. Isomorfismus mezi algebrou $\mathbb{A}$ a nějakou podalgebrou algebry $\mathbb{B}$ je *isomorfní vnoření* $\mathbb{A}$ do $\mathbb{B}$.

Mezi isomorfními algebrami není žádný podstatný rozdíl, jejich Booleovská struktura je totožná. Každá algebra je isomorfní sama se sebou, inverzní zobrazení k isomorfismu je rovněž isomorfismus, a složení isomorfismů je isomorfismus. Býti isomorfní je tedy ekvivalence na třídě Booleových algeber.

4.4.5 Příklad. Potenční algebra $P(X)$ je isomorfní s algebrou 2^X. Stačí množině $A \subseteq X$ přiřadit její *charakteristickou funkci* $\chi_A \in 2^X$ definovanou následovně: $\chi_A(x) = 1$ pokud $x \in A$, $\chi_A(x) = 0$ pokud $x \notin A$.

4.4.6 Příklad. Rozklad množiny $\mathbb{N}$ přirozených čísel na $\{X_k \subseteq \mathbb{N}; k \in \mathbb{N}\}$ určuje isomorfní vnoření algebry $P(\mathbb{N})$ do sebe: pro $A \subseteq \mathbb{N}$ položme $f(A) = \bigcup_{k\in A} X_k$. Algebra $P(\mathbb{N})$ obsahuje $\mathfrak{c}$ navzájem různých kopií sebe sama.

4.4.7 Věta. *Každá konečná algebra je isomorfní s 2^n pro nějaké $n \in \mathbb{N}$.*

Důkaz. Konečná algebra $\mathbb{B}$ je atomární, označme jako A množinu jejích atomů. Pro $x \in \mathbb{B}$ položme $f(x) = \{a \in A; a \leq x\}$. Ukážeme, že f je isomorfismus.

Pro $x \neq y$ můžeme předpokládat, že $x \not\leq y$ (nebo $y \not\leq x$), takže $x \wedge -y \neq \mathbf{0}$. Zvolme libovolný atom a pod nenulovým prvkem $x \wedge -y$; pak je $a \in f(x) \setminus f(y)$, tedy $f(x) \neq f(y)$ a f je prosté. Je-li $X = \{a_1, \ldots, a_n\} \subseteq A$ nějaká množina atomů, pak pro prvek $x = a_1 \vee \ldots \vee a_n \in \mathbb{B}$ je $f(x) = X$, tedy f je *na*.

Pro každé $x \in \mathbb{B}$ a každý atom $a \in A$ je $a \leq -x$ právě když $a \not\leq x$, takže $f(-x) = A \setminus f(x)$ a zobrazení f zachovává komplementy. Pro $x, y \in \mathbb{B}$ a každý atom a je $a \leq x \vee y$ právě když $a \leq x$ nebo $a \leq y$, takže $f(x \vee y) = f(x) \cup f(y)$ a zobrazení f zachovává spojení. Tedy $f : \mathbb{B} \to P(A) \simeq 2^{|A|}$ je isomorfismus. □

4.4.8 Důsledek. *Pokud nějaká rovnost Booleovských termů platí v triviální algebře, pak již platí v každé Booleově algebře.*

Důkaz. Buď $s(x_1, \ldots, x_k) = t(x_1, \ldots, x_k)$ formule splněná v triviální algebře. Potom je splněna v každé konečné algebře $\mathbb{A}$, neboť pro konečnou algebru je $\mathbb{A} \simeq 2^n$, a operace v 2^n jsou definovány po složkách. Je-li nakonec $\mathbb{B}$ libovolná Booleova algebra a $b_1, \ldots, b_k \in \mathbb{B}$, máme ukázat, že v $\mathbb{B}$ platí $s(b_1, \ldots, b_k) = t(b_1, \ldots, b_k)$. K tomu stačí uvážit konečnou podalgebru $\mathbb{A} = \langle b_1, \ldots, b_k \rangle \subseteq \mathbb{B}$: hodnota termů s, t v $\mathbb{A}$ je stejná jako v $\mathbb{B}$, a v konečné algebře $\mathbb{A}$ se rovnají. □

Při studiu homomorfismů je přirozené se ptát, která zobrazení mezi algebrami lze rozšířit do homomorfismu. Úplnou odpověď dává následující věta. Pro jednoduchost zápisu zavedeme pro $x \in A$ značení $(+1)x = x$ a $(-1)x = -x$. Pro $x_1, \ldots, x_m \in \mathbb{A}$ a posloupnost $e = (e_i)_{i\leq m} \in \{+1, -1\}^m$ pak můžeme průsek prvků $x_1, \ldots, x_m$ nebo jejich doplňků psát krátce jako $\bigwedge_{i\leq m} e_i x_i$.

4.4.9 Věta (Sikorski)**.** *Buď $\mathbb{A}$ Booleova algebra generovaná množinou $X \subseteq \mathbb{A}$. Zobrazení $z : X \to \mathbb{B}$ do Booleovy algebry $\mathbb{B}$ má rozšíření do homomorfismu $f : \mathbb{A} \to \mathbb{B}$ právě když pro každé $x_1, \ldots, x_m \in X$ a každé $e \in \{+1, -1\}^m$ platí*

$$\bigwedge_{i\leq m} e_i x_i = \mathbf{0} \to \bigwedge_{i\leq m} e_i z(x_i) = \mathbf{0}.$$

Důkaz. Jedna implikace je triviální: homomorfismus zachová doplňky, průseky a nulu. V opačném směru máme najít homomorfní rozšíření $f : \mathbb{A} \to \mathbb{B}$. Podle věty o normálním tvaru lze každé $a \in \mathbb{A} = \langle X \rangle$ zapsat jako $a = \bigvee_{i \leq m} \bigwedge_{j \leq p_i} e_{ij} x_{ij}$ pro nějaká $x_{ij} \in X$. Jako obraz takového prvku a potom přichází v úvahu jen $f(a) = \bigvee_{i \leq m} \bigwedge_{j \leq p_i} e_{ij} z(x_{ij})$. Ukážeme, že to je hledané homomorfní rozšíření.

Daný prvek $a \in \mathbb{A}$ lze zapsat v normálním tvaru nad X různými způsoby. Musíme tedy předně ukázat, že předpis pro $f(a)$ nezávisí na volbě normálního tvaru. Je-li $b = \bigvee_{k \leq n} \bigwedge_{l \leq q_k} \varepsilon_{kl} y_{kl}$ jakékoli jiné vyjádření téhož prvku $a = b$, máme ukázat, že $f(a) = f(b)$. Ukážeme jen $f(a) - f(b) = \mathbf{0}$, zbytek je podobný.

Máme $a - b = \bigvee_{i \leq m} \bigwedge_j e_{ij} x_{ij} - \bigvee_{k \leq n} \bigwedge_l \varepsilon_{kl} y_{kl} = \mathbf{0}$. Pomocí distribučního zákona a deMorganových formulí pro $-b$ můžeme ekvivalentně psát

$$\bigvee_{(l_k)} \bigvee_{i \leq m} \bigwedge_{j \leq p_i} \bigwedge_{l \leq q_k} (e_{ij} x_{ij} - \varepsilon_{kl_k} y_{kl_k}) = \mathbf{0},$$

kde $\bigvee_{(l_k)}$ je spojení přes všechny posloupnosti[4] indexů $(l_k)_{k \leq n}$ takové, že $l_k \leq q_k$ pro každé $k \leq n$. Jelikož je toto spojení nulové, jsou nulové i všechny průseky $\bigwedge_j \bigwedge_l (e_{ij} x_{ij} - \varepsilon_{kl_k} y_{kl_k})$, a podle předpokladu jsou tedy nulové i všechny průseky $\bigwedge_j \bigwedge_l (e_{ij} z(x_{ij}) - \varepsilon_{kl_k} z(y_{kl_k}))$. Potom je ale také

$$\bigvee_{(l_k)} \bigvee_{i \leq m} \bigwedge_{j \leq p_i} \bigwedge_{l \leq q_k} (e_{ij} z(x_{ij}) - \varepsilon_{kl_k} z(y_{kl_k})) = \mathbf{0},$$

neboli $\bigvee_{i \leq m} \bigwedge_j e_{ij} z(x_{ij}) - \bigvee_{k \leq n} \bigwedge_l \varepsilon_{kl} z(y_{kl}) = f(a) - f(b) = \mathbf{0}$.

Ukázali jsme, že předpis pro $f(a)$ nezávisí na volbě normálního tvaru, takže korektně definuje zobrazení $f : \mathbb{A} \to \mathbb{B}$. Jelikož f z definice zachovává normální tvary, snadno se nahlédne, že zachovává doplňky, průseky a spojení, takže f je homomorfismus. Prvky $x \in X$ jsou samy v normálním tvaru nad X, takže je $f(x) = z(x)$ pro každé $x \in X$. Rozšíření $f : \mathbb{A} \to \mathbb{B}$ je definováno na celé algebře, neboť X generuje $\mathbb{A}$. Ze stejného důvodu je toto rozšíření jediné. □

4.4.10 Cvičení. (a) Dejte příklad zobrazení $f : \mathbb{A} \to \mathbb{B}$ mezi Booleovými algebrami, které zachovává $\mathbf{0}, \mathbf{1}, \wedge, \vee$, ale není homomorfismem. (b) Pro zobrazení $z : X \to Y$ a $A \subseteq X$, $B \subseteq Y$ položme $f(A) = z[A] \subseteq Y$ a $g(B) = z^{-1}[B] \subseteq X$. Jsou $f : P(X) \to P(Y)$ a $g : P(Y) \to P(X)$ homomorfismy?

4.4.11 Cvičení. Buď $f : \mathbb{A} \to \mathbb{B}$ homomorfismus. (a) $f[\mathbb{A}] \subseteq \mathbb{B}$ je podalgebra. (b) f je prostý právě když pro $x > \mathbf{0}$ je $f(x) > \mathbf{0}$. (c) Pokud se f shoduje s morfismem g na nějaké množině generátorů $X \subseteq \mathbb{A}$, je $f = g$.

4.4.12 Cvičení. (a) Algebra dělitelů z 4.1.10 je konečná; popište nějaký její isomorfismus s algebrou tvaru 2^n. (b) Konečné algebry jsou isomorfní právě tehdy, když mají stejný počet atomů. Kolik různých isomorfismů mezi nimi potom existuje? (c) Algebra periodických množin $A \subseteq \mathbb{Z}$ s nenulovou periodou $p \in \mathbb{Z}$ je isomorfní s konečnou algebrou 2^p. Algebra periodických množin $A \subseteq \mathbb{Q}$

[4] Tyto posloupnosti jsou prostě indexy průseků normálního tvaru pro $-b$. Pro názornost uvažme $b = (y_{11} \wedge -y_{12}) \vee (y_{21} \wedge -y_{22} \wedge -y_{23})$. Potom je $-b = (-y_{11} \vee y_{12}) \wedge (-y_{21} \vee y_{22} \vee y_{23}) = (-y_{11} \wedge -y_{21}) \vee (-y_{11} \wedge y_{22}) \vee (-y_{11} \wedge y_{23}) \vee (y_{12} \wedge -y_{21}) \vee (y_{12} \wedge y_{22}) \vee (y_{12} \wedge y_{23})$, což je právě spojení průseků $\bigwedge_{k \leq 2} -\varepsilon_{kl_k} y_{kl_k}$ přes posloupnosti $(11), (12), (13), (21), (22), (23)$.

($A \subseteq \mathbb{R}$) s nenulovou periodou $p \in \mathbb{Q}$ ($p \in \mathbb{R}$) je isomorfní s algebrou konečných množin racionálních (reálných) čísel a jejich doplňků.

4.4.13 Cvičení. (a) Algebry $P(X)$ a $P(Y)$ jsou isomorfní právě když $|X| = |Y|$. (b) Algebra konečných $A \subseteq \mathbb{N}$ a jejich doplňků není isomorfní s žádnou $P(X)$. Atomární algebry se stejným počtem atomů tedy obecně nemusí být isomorfní.

4.4.14 Cvičení. Zobrazení $f : \mathbb{A} \to \mathbb{B}$ je isomorfismus algeber právě tehdy, když je isomorfismem uspořádání $(\mathbb{A}, \leq)$ a $(\mathbb{B}, \leq)$. Homomorfismus $f : \mathbb{A} \to \mathbb{B}$ je monotónní zobrazení z $(\mathbb{A}, \leq)$ do $(\mathbb{B}, \leq)$, ale nikoli nutně naopak.

4.4.15 Cvičení. Napište sentenci jazyka Booleových algeber, která platí právě a jen v algebře 2^2 (a jejích isomorfních kopiích). Podobně se nahlédne, že takto lze charakterizovat každou konečnou Booleovu algebru.

4.5 Ideály a filtry

Pojem ideálu a filtru na Booleově algebře zobecňuje pojem ideálu a filtru na množině (viz 3.9). Předvedeme souvislost ideálů na algebře s jejími homomorfními obrazy a souvislost filtrů na algebře jazyka s predikátovou logikou.

4.5.1 Definice. Podmnožina $\emptyset \neq \mathcal{I} \subset \mathbb{B}$ Booleovy algebry $\mathbb{B}$ je *ideál*, pokud pro $x, y \in \mathcal{I}$ je také $x \vee y \in \mathcal{I}$, a pro $x \leq y \in \mathcal{I}$ je také $x \in \mathcal{I}$. Podmnožina $\emptyset \neq \mathcal{F} \subset \mathbb{B}$ je *filtr*, pokud pro $x, y \in \mathcal{F}$ je $x \wedge y \in \mathcal{F}$, a pro $y \geq x \in \mathcal{F}$ je $y \in \mathcal{F}$.

Jinými slovy, ideál je dolů uzavřená, nahoru usměrněná množina, a filtr je nahoru uzavřená, dolů usměrněná množina. Zřejmě $\mathbf{0} \in \mathcal{I}$ a $\mathbf{1} \notin \mathcal{I}$ pro každý ideál $\mathcal{I}$ a naopak $\mathbf{0} \notin \mathcal{F}$ a $\mathbf{1} \in \mathcal{F}$ pro každý filtr $\mathcal{F}$.

4.5.2 Cvičení. (a) $\mathcal{I}$ je ideál právě když $\mathcal{I}^* = \{-x; x \in \mathcal{I}\}$ je filtr. (b) $\mathcal{F}$ je filtr právě když $\mathcal{F}_* = \{-x; x \in \mathcal{F}\}$ je ideál. (c) Pro každý ideál $\mathcal{I}$ na $\mathbb{B}$ je $\mathcal{I} \cup \mathcal{I}^*$ podalgebra $\mathbb{B}$; podobně pro filtry.

4.5.3 Příklad. Na každé algebře máme *triviální ideál* $\{\mathbf{0}\}$ a *triviání filtr* $\{\mathbf{1}\}$. Pro $b \neq \mathbf{1}$ je $\mathcal{I}_b = \{x \in \mathbb{B}; x \leq b\}$ ideál a pro $b \neq \mathbf{0}$ je $\mathcal{F}_b = \{x \in \mathbb{B}; x \geq b\}$ filtr. Takové ideály a filtry nazýváme *hlavní*. Ideál $Fin = \{A \subseteq \mathbb{N}; A \text{ konečná}\}$ na $P(\mathbb{N})$ není hlavní; nazývá se *Fréchetův ideál*. K němu duální *Fréchetův filtr* sestává z množin s konečným doplňkem. Podalgebra $Fin \cup Fin^*$ algebry $P(\mathbb{N})$ je příkladem spočetné atomární algebry.

Příklady ideálů a filtrů na potenčních algebrách, které se přirozeně vyskytují v různých odvětvích matematiky, uvádíme v 3.9.

Snadno ověříme, že průnik každého systému ideálů je opět ideál, a průnik každého systému filtrů je filtr. To nás opravňuje k následující definici.

4.5.4 Definice. Podmnožina $X \subseteq \mathbb{B}$ Booleovy algebry je *centrovaná*, pokud pro každých konečně mnoho $x_1, \ldots, x_n \in X$ je $x_1 \wedge \ldots \wedge x_n > \mathbf{0}$. Je-li $X \subseteq \mathbb{B}$ centrovaná, buď $\mathcal{F}(X)$ průnik všech filtrů $\mathcal{F} \subseteq \mathbb{B}$ takových, že $X \subseteq \mathcal{F}$. Říkáme, že množina X *generuje* filtr $\mathcal{F}(X)$. O podmnožině X filtru $\mathcal{F}$ řekneme, že je *bází* filtru $\mathcal{F}$, pokud pod každým prvkem $y \in \mathcal{F}$ leží nějaké $x \in X$.

Každá podmnožina filtru je centrovaná, a naopak každá centrovaná podmnožina generuje filtr. Je-li $X \subseteq \mathbb{B}$ centrovaná, je $\mathcal{F}(X)$ nejmenší filtr na $\mathbb{B}$ obsahující X jako podmnožinu. Hlavní filtry jsou právě filtry s jednoprvkovou bazí. Přitom filtr $\mathcal{F}$ generovaný konečnou množinou $\{x_1, \ldots, x_n\} \subseteq \mathcal{F}$ má jednoprvkovou bazi $x_1 \wedge \ldots \wedge x_n \in \mathcal{F}$. Podmnožina X filtru $\mathcal{F}$ je generující právě tehdy, když pro každé $y \in \mathcal{F}$ existuje konečně mnoho $x_1, \ldots, x_n \in X$ takových, že $x_1 \wedge \ldots \wedge x_n \leq y$. Jinými slovy, množina $X \subseteq \mathcal{F}$ generuje filtr $\mathcal{F}$ právě tehdy, když množina všech konečných průniků z X je bazí $\mathcal{F}$. Fréchetův filtr není generován žádnou konečnou množinou.

4.5.5 Příklad. Buď T výroková teorie v jazyce $\mathcal{A}$. Potom T je bezesporná právě tehdy, když $\{[\varphi]; \varphi \in T\}$ je centrovaná množina v algebře výroků $\mathbb{B}(\mathcal{A})$. Konjunkce formulí z T pak tvoří bazi filtru $Thm(T) = \{[\varphi]; T \vdash \varphi\}$ na $\mathbb{B}(\mathcal{A})$.

Jako speciální případ máme triviální filtr $Thm(\emptyset) = \{\mathbf{1}\}$: prázdná výroková teorie (totiž výroková logika) je bezesporná a dokazuje právě všechny tautologie.

4.5.6 Příklad. Buď T teorie v jazyce $\mathcal{L}$ predikátové logiky. Potom T je bezesporná právě tehdy, když $\{[\varphi]; \varphi \in T\}$ je centrovaná množina v Lindenbaumově algebře $\mathbb{B}(\mathcal{L})$. Množina T pak generuje filtr $Thm(T) = \{[\varphi]; T \vdash \varphi\}$ na $\mathbb{B}(\mathcal{L})$. Tento filtr je hlavní právě tehdy, když je teorie T konečně axiomatizovatelná. Přitom $Thm(T) = T$ právě tehdy, když je T uzavřená na důsledky.

Popíšeme nyní vztah mezi ideály na algebře a jejími homomorfními obrazy. Nejprve potřebujeme zavést pojem ekvivalence podle ideálu (filtru).

4.5.7 Definice. Pro prvky $x, y \in \mathbb{A}$ položme $x \triangle y = (x - y) \vee (y - x)$. Prvek $x \triangle y$ je *symetrická diference* prvků x, y. Je-li potom $\mathcal{I}$ ideál na algebře $\mathbb{A}$, řekneme, že prvky $x, y \in \mathbb{A}$ jsou *ekvivalentní vůči ideálu* $\mathcal{I}$, pokud je $x \triangle y \in \mathcal{I}$. V tom případě píšeme $x \equiv_{\mathcal{I}} y$.

Duálně můžeme zavést ekvivalenci $x \equiv_{\mathcal{F}} y$ podle filtru $\mathcal{F}$ jako ekvivalenci podle $\mathcal{F}_*$. Potom je $x \equiv_{\mathcal{F}} y$ právě když $-(x \triangle y) = (x \wedge y) \vee -(x \vee y) \in \mathcal{F}$. Smysl těchto definic je dosti názorný: ekvivalentní jsou takové prvky, jejichž diference je malá, resp. jejichž ekvivalence je velká.

Snadno se ověří, že relace $\equiv_{\mathcal{I}}$ je ekvivalence na $\mathbb{A}$. Navíc $\equiv_{\mathcal{I}}$ respektuje Booleovské operace: pro $x_1 \equiv y_1$ a $x_2 \equiv y_2$ je také $(x_1 \wedge x_2) \equiv (y_1 \wedge y_2)$, $(x_1 \vee x_2) \equiv (y_1 \vee y_2)$ a $(-x_1) \equiv (-y_1)$. Ekvivalence respektující algebraickou strukturu se nazývá *kongruence*.

Pro prvek $a \in \mathbb{A}$ označme jako $[a]_{\mathcal{I}}$ nebo stručněji $[a]$ jeho ekvivalenční třídu vůči relaci $\equiv_{\mathcal{I}}$. Specielně $[\mathbf{0}] = \mathcal{I}$ a $[\mathbf{1}] = \mathcal{I}^*$. Množina $\mathbb{A}/\mathcal{I}$ těchto ekvivalenčních tříd je potom *kvocient* algebry $\mathbb{A}$ podle ideálu $\mathcal{I}$. Předchozí pozorování nás opravňuje k následující definici:

4.5.8 Definice. Buď $\mathcal{I}$ ideál na algebře $\mathbb{A}$. Kvocient $\mathbb{A}/\mathcal{I}$ opatřený Booleovskými operacemi $\mathbf{0} = \mathcal{I}$, $\mathbf{1} = \mathcal{I}^*$, $-[a] = [-a]$, $[a] \wedge [b] = [a \wedge b]$, $[a] \vee [b] = [a \vee b]$ pak nazýváme *kvocientní algebrou*. Zobrazení $\pi_{\mathcal{I}} : \mathbb{A} \to \mathbb{A}/\mathcal{I}$, které prvku $a \in \mathbb{A}$ přiřazuje ekvivalenční třídu $[a]$, potom nazýváme *kanonický homomorfismus* algebry $\mathbb{A}$ na kvocient $\mathbb{A}/\mathcal{I}$.

Definice Booleovských operací na kvocientu je korektní, tj. nezávisí na použitých reprezentatech ekvivalenčních třích, jelikož $\equiv_{\mathcal{I}}$ je kongruence. Stejně snadno se ověří, že $\pi_{\mathcal{I}}$ je skutečně homomorfismus.

Kvocient podle filtru můžeme zavést jako kvocient podle duálního ideálu. Například Lindenbaumova algebra $\mathbb{B}(T)$ teorie T je potom právě kvocient algebry $\mathbb{B}(\mathcal{L})$ jazyka $\mathcal{L}$ podle filtru $Thm(T)$.

4.5.9 Věta (o homomorfismu). *Buď $f : \mathbb{A} \to \mathbb{B}$ homomorfismus algebry $\mathbb{A}$ na algebru $\mathbb{B}$. Potom $\mathcal{I} = \{x \in \mathbb{B}; f(x) = \mathbf{0}\}$ je ideál na algebře $\mathbb{A}$, a kvocient $\mathbb{A}/\mathcal{I}$ je isomorfní s $\mathbb{B}$. Tedy homomorfní obrazy algebry $\mathbb{A}$ jsou právě její kvocienty.*

Důkaz. Snadno se ověří, že $\mathcal{I}$ je ideál, a máme tedy kvocient $\mathbb{A}/\mathcal{I}$. Pro $[x] \in \mathbb{A}/\mathcal{I}$ položme $g([x]) = f(x) \in \mathbb{B}$. Ukážeme, že $g : \mathbb{A}/\mathcal{I} \to \mathbb{B}$ je isomorfismus. Předně, definice $g([x])$ nezávisí na reprezentantu $x \in \mathbb{A}$, protože pro $[x] = [y]$ je $x \triangle y \in \mathcal{I}$, takže $f(x) \triangle f(y) = f(x \triangle y) = \mathbf{0}$, neboli $f(x) = f(y)$. Pro $x \in \mathbb{A}$ je $g(-[x]) = g([-x]) = f(-x) = -f(x) = -g([x])$, takže g zachovává komplementy, a pro $x, y \in \mathbb{A}$ je $g([x] \wedge [y]) = g([x \wedge y]) = f(x \wedge y) = f(x) \wedge f(y) = g([x]) \wedge g([y])$, takže g zachovává průseky. Přitom g je prosté, protože $[x] \neq [y]$ znamená právě tolik co $f(x) \neq f(y)$. Zároveň g je na, protože f je na. Tedy g je isomorfismus. □

4.5.10 Cvičení. Podmínky z definice ideálu (filtru) lze nahradit jedinou podmínkou: $x \vee y \in \mathcal{I}$ právě když $x, y \in \mathcal{I}$ ($x \wedge y \in \mathcal{F}$ právě když $x, y \in \mathcal{F}$).

4.5.11 Cvičení. Je-li $f : \mathbb{A} \to \mathbb{B}$ homomorfismus a $X \subset \mathbb{B}$ je ideál (filtr) pak $f^{-1}[X] \subset \mathbb{A}$ je ideál (filtr). Specielně ideál $f^{-1}[\mathbf{0}] = \ker(f)$ je *kernel* homomorfismu f. Přitom $\ker(f) = \{\mathbf{0}\}$ právě když homomorfismus f je prostý.

4.6 Ultrafiltry

4.6.1 Definice. Filtr $\mathcal{F}$ na algebře $\mathbb{B}$ je *ultrafiltr*, pokud je maximální, tj. pokud pro každý filtr $\mathcal{G}$ splňující $\mathcal{F} \subseteq \mathcal{G} \subset \mathbb{B}$ je $\mathcal{F} = \mathcal{G}$. Jinými slovy, ultrafitr je takový filtr, který již nelze rozšířit do většího filtru.

4.6.2 Lemma. *Pro filtr $\mathcal{F} \subseteq \mathbb{B}$ jsou následující podmínky ekvivalentní:*

(i) *$\mathcal{F}$ je ultrafiltr;*

(ii) *$x \notin \mathcal{F}$ právě když $-x \in \mathcal{F}$;*

(iii) *$\mathbb{B} \setminus \mathcal{F}$ je právě ideál $\mathcal{F}_*$, takže $\mathbb{B} = \mathcal{F} \cup \mathcal{F}_*$;*

(iv) *$x \vee y \in \mathcal{F}$ právě když $x \in \mathcal{F}$ nebo $y \in \mathcal{F}$.*

Důkaz. (i $\to$ ii) Buď $\mathcal{F}$ maximální filtr a $x \notin \mathcal{F}$. Je-li množina $\mathcal{F} \cup \{x\}$ centrovaná, pak generuje filtr, který rozšiřuje $\mathcal{F}$, což není možné. Tedy $x \wedge a = \mathbf{0}$ pro nějaké $a \in \mathcal{F}$, načež $a \leq -x \in \mathcal{F}$. (iii $\to$ iv) Pokud $x \vee y \in \mathcal{F}$, ale $x, y \notin \mathcal{F}$, pak podle (iii) je $-x, -y \in \mathcal{F}$, tedy také $-x \wedge -y = -(x \vee y) \in \mathcal{F}$, což není možné. (iv $\to$ i) Pokud filtr $\mathcal{G}$ rozšiřuje $\mathcal{F}$, buď $x \in \mathcal{G} \setminus \mathcal{F}$. Máme $\mathbf{1} = x \vee -x \in \mathcal{F}$, přitom $x \notin \mathcal{F}$, takže podle (iv) je $-x \in \mathcal{F} \subset \mathcal{G}$, což není možné. □

4.6.3 Cvičení. Hlavní filtr $\mathcal{F}_a$ je ultrafiltr právě tehdy, když $a \in \mathbb{B}^+$ je atom. Ultrafiltry tohoto tvaru se nazývají *triviální*. Pojem ultrafiltru tedy zobecňuje pojem atomu. Na konečné algebře existují jen triviální ultrafiltry.

4.6.4 Cvičení. Fin^* není ultrafiltr na $P(\mathbb{N})$, ale na algebře $Fin \cup Fin^*$ je netriviálním ultrafiltrem. Ultrafiltr na $P(\mathbb{N})$ je netriviální pokud rozšiřuje Fin^*.

4.6.5 Cvičení. Řekneme, že prvek $x \in \mathbb{B}$ je *kompatibilní* s filtrem $\mathcal{F}$, pokud pro každé $a \in \mathcal{F}$ je $x \wedge a \neq \mathbf{0}$. To je právě tehdy, když $\mathcal{F} \cup \{x\} \subseteq \mathbb{B}$ je centrovaná. Filtr $\mathcal{F}$ je maximální právě tehdy, když obsahuje každý kompatibilní prvek.

4.6.6 Cvičení. (a) Pro podmnožinu $\mathcal{U}$ algebry $\mathbb{B}$ položme $f(x) = \mathbf{0}$ pro $x \in \mathcal{U}$ a $f(x) = \mathbf{1}$ pro $x \notin \mathcal{U}$. Tím je definováno zobrazení $f : \mathbb{B} \to \{0, 1\}$. Ukažte, že f je homomorfismus právě když $\mathcal{U}$ je ultrafiltr. (b) Pravdivostní ohodnocení výrokových formulí v jazyce $\mathcal{A}$ je právě homomorfismus z algebry výroků $\mathbb{B}(\mathcal{A})$ do algebry $\{0, 1\}$. Tedy pravdivostní ohodnocení odpovídají ultrafiltrům na $\mathbb{B}(\mathcal{A})$.

4.6.7 Příklad. Podle 4.5.5 generuje bezesporná teorie T filtr $Thm(T)$ na Lindenbaumově algebře $\mathbb{B}(\mathcal{L})$. Je-li T úplná, je pro každou uzavřenou formuli φ jazyka $\mathcal{L}$ buďto $T \vdash \varphi$ nebo $T \vdash \neg\varphi$. To znamená právě tolik, že $Thm(T)$ je ultrafiltr. Úplné teorie v jazyce $\mathcal{L}$ odpovídají ultrafiltrům na algebře $\mathbb{B}(\mathcal{L})$.

V důkaze věty 1.4.18 o kompaktnosti výrokové logiky jsme rozšířili konečně splnitelnou teorii, tedy centrovaný systém na algebře $\mathbb{B}(\mathcal{A})$, do ultrafiltru; to jest, našli jsme splňující ohodnocení. Podobně v predikátové logice jsme v důkaze Lindenbaumovy věty 2.4.16 rozšířili bezespornou teorii do úplné bezesporné, tj. do ultrafiltru na $\mathbb{B}(\mathcal{L})$. V kontextu ultrafiltrů na Booleových algebrách vidíme, že oba výsledky jsou speciálním případem následující věty.

4.6.8 Věta. *Každý centrovaný systém na algebře $\mathbb{B}$ lze rozšířit do ultrafiltru. Specielně každý nenulový prvek $x \in \mathbb{B}^+$ leží v nějakém ultrafiltru.*

Důkaz. Daný centrovaný systém se rozšiřuje do filtru. Systém $\mathcal{X}$ všech filtrů na algebře $\mathbb{B}$ je částečně uspořádán inkluzí. Přitom v uspořádání $(\mathcal{X}, \subseteq)$ má každý řetězec horní mez: je-li $\mathcal{C} \subseteq \mathcal{X}$ řetězec filtrů na $\mathbb{B}$, pak i $\bigcup \mathcal{C}$ je filtr na $\mathbb{B}$, jak se snadno ověří. Uspořádání $(\mathcal{X}, \subseteq)$ tedy splňuje předpoklady Zornova lemmatu, a nad každým filtrem existuje maximální filtr, tj. ultrafiltr. □

4.6.9 Věta (Stone). *Každá Booleova algebra je isomorfní s algebrou množin.*

Důkaz. Pro algebru $\mathbb{B}$ označme $\mathcal{S}(\mathbb{B}) = \{\mathcal{U} \subseteq \mathbb{B}; \mathcal{U} \text{ je ultrafiltr}\}$. Pro $x \in \mathbb{B}$ položme $f(x) = \{\mathcal{U} \in \mathcal{S}(\mathbb{B}); x \in \mathcal{U}\}$. Ukážeme, že f je vnoření $\mathbb{B}$ do $P(\mathcal{S}(\mathbb{B}))$.

Je-li $x \neq y$, můžeme předpokládat, že $x \not\leq y$, takže $x \wedge -y > \mathbf{0}$. Pro libovolný ultrafiltr $\mathcal{U}$ obsahující $x \wedge -y$ je potom $x \in \mathcal{U}$ a $y \notin \mathcal{U}$, takže $f(x) \neq f(y)$ a zobrazení f je prosté. Pro $x, y \in \mathbb{B}$ a $\mathcal{U} \in \mathcal{S}(\mathbb{B})$ je $x \wedge y \in \mathcal{U}$ právě když $x \in \mathcal{U}$ a $y \in \mathcal{U}$, jelikož $\mathcal{U}$ je filtr, takže $f(x \wedge y) = f(x) \cap f(y)$ a zobrazení f zachovává průseky. Pro každé $x \in \mathbb{B}$ a $\mathcal{U} \in \mathcal{S}(\mathbb{B})$ je $-x \in \mathcal{U}$ právě když $x \notin \mathcal{U}$, jelikož $\mathcal{U}$ je ultrafiltr, takže $f(-x) = \mathcal{S}(\mathbb{B}) \setminus f(x)$ a zobrazení f zachovává komplementy. □

Důkaz Stoneovy věty připomíná důkaz věty 4.4.7, podstatná změna je v tom, že pojem atomu nahrazuje pojmem ultrafiltru. Booleova algebra nemusí mít atomy, ale vždy má dostatek ultrafiltrů.

4.7 Volné algebry

4.7.1 Definice. Booleova algebra $\mathbb{A}$ je *volná nad* $X \subseteq \mathbb{A}$, pokud se každé zobrazení $z : X \to \mathbb{B}$ jediným způsobem rozšiřuje do homomorfismu $f : \mathbb{A} \to \mathbb{B}$.

Technická podmínka uvedená v definici zachycuje intuitivní představu o volnosti: prvky z X nejsou svázány žádnými speciálními vztahy. Každou Booleovskou rovnost, kterou splňují $x_1, \ldots, x_n \in X$, splňují i $f(x_1), \ldots, f(x_n) \in \mathbb{B}$ v libovolné jiné algebře. Žádné prvky z X tedy například nejsou navzájem disjunktní ani porovnatelné, žádný z nich není **0** ani **1**, atd.

4.7.2 Lemma. *Je-li $\mathbb{A}$ je volná nad X, pak X generuje $\mathbb{A}$.*

Důkaz. Je-li $\langle X\rangle \subset \mathbb{A}$ vlastní podalgebra, pak podle **??** existují různé ultrafiltry $\mathcal{U}, \mathcal{V}$ na $\mathbb{A}$ tak, že $\mathcal{U} \cap \langle X\rangle = \mathcal{V} \cap \langle X\rangle$. Homomorfismy $\chi_{\mathcal{U}}, \chi_{\mathcal{V}} : \mathbb{A} \to 2$ pak dvěma různými způsoby rozšiřují totéž zobrazení z X do 2, což je spor. □

4.7.3 Lemma. *Buď $\mathbb{A}$ algebra volně generovaná množinou X a buď $\mathbb{B}$ volně generovaná množinou Y. Potom je-li $|X| = |Y|$, jsou algebry $\mathbb{A}$ a $\mathbb{B}$ isomorfní.*

Důkaz. Bijekce $z : X \to Y$ se jednoznačně rozšiřuje do morfismu $f : \mathbb{A} \to \mathbb{B}$, a bijekce $z^{-1} : Y \to X$ se jednoznačně rozšiřuje do $g : \mathbb{B} \to \mathbb{A}$. Homomorfismus $g \circ f$ a identita na $\mathbb{A}$ se tedy shodují na generátorech, takže $g \circ f$ je identita na $\mathbb{A}$; podobně $f \circ g$ je identita na $\mathbb{B}$. Tedy f a g jsou isomorfismy. □

Volná algebra nad množinou dané mohutnosti je tedy určena jednoznačně až na isomorfismus. Volnou algebru nad množinou X mohutnosti κ budeme značit $Fr(X)$ nebo $Fr(\kappa)$. Zatím jsme ovšem neukázali, že existuje.

4.7.4 Definice. Podmnožina X Booleovy algebry $\mathbb{A}$ je *nezávislá*, pokud pro každých konečně mnoho navzájem různých prvků $x_1, \ldots x_m, y_1, \ldots y_n \in X$ je $x_1 \wedge \ldots \wedge x_m \wedge -y_1 \wedge \ldots \wedge -y_n \neq \mathbf{0}$. Pokud zároveň X generuje algebru $\mathbb{A}$, jsou prvky z X její *nezávislé generátory*.

4.7.5 Věta. *Algebra $\mathbb{A}$ je volná nad X právě když X nezávisle generuje $\mathbb{A}$.*

Důkaz. Víme, že X generuje $\mathbb{A}$. Zároveň je nezávislá: kdyby pro nějakých konečně mnoho $x_1, \ldots x_m, y_1, \ldots y_n \in X$ bylo $x_1 \wedge \ldots \wedge x_m \wedge -y_1 \ldots \wedge -y_n = \mathbf{0}$, položíme $z(x_i) = \mathbf{1}$ a $z(y_j) = \mathbf{0}$; zobrazení $z : X \to 2$ pak nemá rozšíření. Pokud naopak X nezávisle generuje $\mathbb{A}$ a $z : X \to 2$ je nějaké zobrazení, získáme homomorfní rozšíření pomocí Sikorskiho věty 4.4.9. Toto rozšíření je jednoznačné, protože je předem dáno na množině generátorů. □

4.7.6 Věta. *Pro každou mohutnost κ existuje volná algebra $Fr(\kappa)$.*

Důkaz. Je-li $\mathcal{A}$ množina prvotních formulí výrokové logiky, pak algebra $\mathbb{B}(\mathcal{A})$ výrokových formulí je volná nad $\mathcal{A}$: prvotní formule z $\mathcal{A}$ generují $\mathbb{B}(\mathcal{A})$, a snadno se nahlédne, že jsou nezávislé. Stačí tedy vzít $\mathcal{A}$ mohutnosti κ. □

Pomocí elementárních prostředků výrokové logiky máme tedy k dispozici konkrétní reprezentaci volné algebry $Fr(\kappa)$. Specielně konečné volné algebry jsou právě tvaru 2^{2^n}, a konečná podmnožina $X \subseteq \mathbb{A}$ je nezávislá právě tehdy, když generuje podalgebru maximální možné mohutnosti $2^{2^{|X|}}$. Pro čtenáře obeznámeného se základy topologie dodejme, že $Fr(\kappa)$ lze reprezentovat i jako algebru $CO(2^\kappa)$ obojetných množin Cantorova prostoru.

4.7.7 Důsledek. *Každá Booleova algebra je obrazem volné Booleovy algebry.*

Důkaz. Pro danou algebru $\mathbb{A}$ stačí vzít volnou algebru nad množinou X mohutnosti $|X| \geq |\mathbb{A}|$. Libovolné zobrazení z X na $\mathbb{A}$ se pak z definice rozšiřuje do homomorfismu $Fr(X)$ na $\mathbb{A}$. □

4.7.8 Důsledek. *Každá Booleova algebra je tvaru $\mathbb{B}(T)$ pro vhodnou teorii T.*

Důkaz. Daná algebra $\mathbb{A}$ je obrazem volné algebry $\mathbb{B}(\mathcal{A})$ při nějakém homomorfismu $f : \mathbb{B}(\mathcal{A}) \to \mathbb{A}$. Podle věty o homomorfismu je tedy $\mathbb{A}$ isomorfní s kvocientem algebry $\mathbb{B}(\mathcal{A})$ podle filtru $T = \{\varphi; f([\varphi]) = \mathbf{1}\}$, což je algebra $\mathbb{B}(T)$. □

4.8 Distributivita

Axiomy Booleovy algebry zajišťují platnost distributivních zákonů, tj. tvrzení $x \wedge (y \vee z) = (x \vee y) \wedge (x \vee z)$ a duálně $x \vee (y \wedge z) = (x \wedge y) \vee (x \wedge z)$. Indukcí se snadno dokáže analogické tvrzení pro libovolné konečné spojení konečných průseků (resp. konečný průsek konečných spojení). Ptáme se nyní, jaké distributivní zákony platí pro nekonečná spojení a průseky.

4.8.1 Lemma. *Buď $\mathbb{A}$ Booleova algebra. Potom pro každá $a, a_i, b_j \in \mathbb{A}$ platí*

(i) $a \wedge \bigvee \{b_j; j \in J\} = \bigvee \{a \wedge b_j; j \in J\}$,
$a \vee \bigwedge \{b_j; j \in J\} = \bigwedge \{a \vee b_j; j \in J\}$

(ii) $\bigvee \{a_i; i \in I\} \wedge \bigvee \{b_j; j \in J\} = \bigvee \{a_i \wedge b_j; i \in I, j \in J\}$,
$\bigwedge \{a_i; i \in I\} \vee \bigwedge \{b_j; j \in J\} = \bigwedge \{a_i \vee b_j; i \in I, j \in J\}$

Prvním zajímavým případem je tedy průsek spočetně mnoha spočetných spojení, nebo duálně spojení spočetně mnoha spočetných průseků.

4.8.2 Definice. Řekneme, že algebra $\mathbb{A}$ je (ω, ω)-distributivní, pokud pro každý systém $\{a^i_j; i, j \in \omega\} \subseteq \mathbb{A}$ je $\bigwedge_i \bigvee_j a^i_j = \bigvee_{f \in \omega^\omega} \bigwedge_i a^i_{f(i)}$.

Podmínka uvedená v definici je přirozeným zobecněním konečných distributivních zákonů. Na levé straně stojí průsek jednotlivých spojení, na pravé pak spojení jednotlivých průseků přes všechny funkce,[5] které do průseku vybírají pro dané $i \in \omega$ vždy jeden prvek $a^i_{f(i)}$ ze všech a^i_j.

4.9 Úplné algebry

4.9.1 Lemma. *Buď $\mathbb{A}$ κ-úplná Booleova algebra, buď $\mathcal{I}$ κ-ideál na $\mathbb{A}$. Potom kvocientní algebra $\mathbb{A}/\mathcal{I}$ je opět κ-úplná.*

Podle Stoneovy věty je každá Booleova algebra isomorfní s algebrou množin. Úplná atomární algebra je isomorfní s potenční algebrou. Nabízí se otázka, jestli lze toto tvrzení zobecnit na κ-úplné algebry. Například, je každá σ-úplná algebra σ-isomorfní s nějakou σ-algebrou množin?

4.9.2 Věta (Loomis-Sikorski)**.** *Buď $\mathbb{A}$ σ-úplná Booleova algebra. Potom existuje σ-algebra množin $\mathbb{B}$, σ-ideál $\mathcal{I}$ na $\mathbb{B}$, a σ-úplný isomorfismus mezi $\mathbb{A}$ a $\mathbb{B}/\mathcal{I}$.*

[5] Srv. též poznámku v důkaze věty 4.4.9.

Důkaz. Označme jako $\mathcal{S}$ množinu všech $S \subseteq \mathbb{A}$ takových, že pro každé $a \in \mathbb{A}$ je $a \in S$ nebo $-a \in S$, ale ne oboje. Pro $a \in \mathbb{A}$ položme $s(a) = \{S \in \mathcal{S}; a \in S\}$. Buď $\mathbb{B} \subseteq P(\mathcal{S})$ nejmenší σ-algebra množin obsahující všechny $s(a)$, a buď $\mathcal{I} \subseteq \mathbb{B}$ nejmenší σ-ideál obsahující všechny $\bigcap_{a\in X} s(a)$, kde $X \subseteq \mathbb{A}$ je nanejvýš spočetná množina s nulovým průsekem. Kvocient $\mathbb{B}/\mathcal{I}$ je pak σ-úplná algebra podle 4.9.1. Ukážeme, že zobrazení $f(a) = [s(a)]_{\mathcal{I}}$ je σ-úplný isomorfismus mezi $\mathbb{A}$ a $\mathbb{B}/\mathcal{I}$.

Zobrazení $f : \mathbb{A} \to \mathbb{B}/\mathcal{I}$ očividně zachovává komplementy. Je-li $a = \bigvee a_i$, je každé $a_i \leq a$, neboli $a_i \wedge -a = \mathbf{0}$, takže máme $s(a_i) \setminus s(a) \in \mathcal{I}$, neboli $f(a_i) = [s(a_i)] \leq [s(a)] = f(a)$. Zároveň je $a \wedge -\bigvee a_i = a \wedge \bigwedge -a_i = \mathbf{0}$, takže máme $s(a) \cap \bigcap -s(a_i) = s(a) \cap -\bigcup s(a_i) \in \mathcal{I}$, neboli $f(a) = [s(a)] \leq [\bigcup s(a_i)] = \bigvee [s(a_i)] = \bigvee f(a_i)$. Zobrazení f tedy zachovává spočetná spojení a je σ-úplným homomorfismem. To znamená, že obraz $f[A]$ je σ-úplná podalgebra; přitom $s(a)$ generují $\mathbb{B}$, takže f je zobrazení na $\mathbb{B}/\mathcal{I}$. Zbývá ukázat, že je prosté.

Ukážeme, že $f(a) = \mathbf{0}$ jen pro $a = \mathbf{0}$. Je-li $f(a) = [s(a)] = \mathbf{0}$, je $s(a) \in \mathcal{I}$. To znamená, že množina $s(a)$ je pokryta nějakými spočetně mnoha generátory σ-ideálu $\mathcal{I}$, neboli $s(a) \subseteq \bigcup_m \bigcap_k s(a_k^m)$, kde pro každé m je $\bigwedge_k a_k^m = \mathbf{0}$. Stačí tedy ukázat, že v takovém případě je $a \leq \bigvee_m \bigwedge_k a_k^m$. Ukážeme místo toho duálně, že pro $\bigcap_m \bigcup_k s(a_k^m) \subseteq s(a)$ je $\bigwedge_m \bigvee_k a_k^m \leq a$.

Algebra množin $\mathbb{B}$ je plně distributivní, takže $\bigcap_m \bigcup_k s(a_k^m) \subseteq s(a)$ ekvivalentně znamená, že je $\bigcap_m s(a^m_{k(m)}) \subseteq s(a)$ pro každou $k : \omega \to \omega$. To se ale pro danou $k : \omega \to \omega$ může stát jen tak, že posloupnost $a^m_{k(m)}$ buďto obsahuje a, nebo obsahuje nějaký prvek i jeho komplement. Jinak totiž existuje $S \in \mathcal{S}$ obsahující všechny $a^m_{k(m)}$ a zároveň $-a$, což je spor.

V tom případě je ale $\bigwedge_m \bigvee_k a_k^m \leq a$. Pokud ne, buď $b = \bigwedge_m \bigvee_k a_k^m - a \neq \mathbf{0}$. Máme $b \leq \bigwedge_m \bigvee_k a_k^m$, takže je $b \leq \bigvee_k a_k^m$ pro každé $m \in \omega$. Specielně pro $m = 1$ tedy existuje nějaké $k(1) \in \omega$ tak, že $b \wedge a^1_{k(1)} \neq \mathbf{0}$; pro $m = 2$ potom existuje nějaké $k(2) \in \omega$ tak, že $b \wedge a^1_{k(1)} \wedge a^2_{k(2)} \neq \mathbf{0}$. Indukcí tak získáme indexy $k(m)$ takové, že $b \wedge \bigwedge_{m\leq n} a^m_{k(m)} \neq \mathbf{0}$ pro každé n. Tím je určena funkce $k : \omega \to \omega$ taková, že posloupnost $a^m_{k(m)}$ neobsahuje ani komplementární prvky, ani a. □

Kapitola 5

Vyčíslitelnost

Od pradávna používají se v matematice různé *algoritmy*, počínaje staroindickými postupy pro základní číselné operace. Pro různé aritmetické úlohy, jako například ověřit prvočíselnost nebo najít nejmenší společný násobek, známe mechanické postupy, které každou instanci dané úlohy přivedou po konečně mnoha krocích ke správné odpovědi.

Teprve na počátku dvacátého století se však do popředí dostala otázka, co přesně se algoritmem vůbec myslí. Hlavní podnět vzešel z matematické logiky, když vzniklo podezření, že některé matematické otázky algoritmické řešení nemají: chceme-li ukázat, že nějaká úloha algoritmické řešení nemá, musíme předně vyjasnit, co za algoritmus považujeme. Kromě samotné teorie vyčíslitelnosti je zde hlavním výsledkem věta o algoritmické nerozhodnutelnosti otázky po dokazatelnosti (*Entscheidungsproblem*). Již v Hilbertově seznamu problémů (1900) ale figuruje problém diofantických rovnic, totiž úloha rozhodnout, zda má daný polynom s celočíselnými koeficienty nějaké celočíselné kořeny; mnohem později (1970) se ukázalo, že ani tato úloha nemá algoritmické řešení.

Některé otázky v logice (a vůbec v matematice, potažmo informatice) lze naopak řešit zcela mechanicky: například zjistit, zda jedna formule je instancí druhé, zda je daná výroková formule tautologická, nebo zda je nějaká daná posloupnost formulí korektním důkazem. V této kapitole chceme mimo jiné prozkoumat právě hranici algoritmické rozhodnutelnosti.

Během třicátých let se pak teorie algoritmů, nazývaná též teorie rekurze či teorie vyčíslitelnosti, etablovala jako samostatný obor. Základní otázkou zde je, které funkce máme považovat za *efektivně vyčíslitelné*, tj. takové, u kterých můžeme výpočet funkčních hodnot svěřit nějakému stroji; podobně pak, pro které úlohy existuje *rozhodovací algoritmus*, který na každou instanci v konečném čase správně odpoví. Taková otázka má dokonce filozofický rozměr: do jaké míry může být lidský rozum nahrazen strojem?

Popíšeme dvě hlavní formalizace efektivní vyčíslitelnosti, totiž Turingovy stroje a rekurzivní funkce, a předvedeme, že jejich výpočetní síla je stejná. Postupně se ukázalo, že i další pokusy o formalizaci pojmu algoritmu jsou s nimi ekvivalentní. To vedlo ke všeobecnému přijetí *Churchovy teze*, podle které pojem rekurzivní funkce zachycuje „správnou“ představu algoritmu, vyjádřenou různými, avšak navzájem ekvivalentními způsoby.

Omezíme se předně na efektivní vyčíslování funkcí, neboť na tento případ

lze převést všechny ostatní. Funkcí přitom rozumíme v této kapitole zobrazení f definované na nějaké podmnožině $dom(f) \subseteq \mathbb{N}^k$, pro nějaké $k \in \mathbb{N}$, s hodnotami v $\mathbb{N}$. To znamená, že argumenty i hodnotami uvažovaných funkcí jsou přirozená čísla, a funkce nejsou nutně *totální*, tj. definované na celém $\mathbb{N}^k$. Pro zkrácení zápisu budeme někdy k-tici $(x_1, \ldots, x_k) \in \mathbb{N}^k$ značit stručně $\overline{x}$ a hodnotu $f(x_1, \ldots, x_k) \in \mathbb{N}$ funkce f na těchto argumentech jako $f(\overline{x})$. Podobně pro k-ární relaci R budeme někdy psát krátce $\overline{x} \in R$ nebo $R(\overline{x})$.

Zastavme se nejprve u samotné představy algoritmu jakožto mechanické procedury, která takovou funkci vyčísluje. Taková procedura by zřejmě měla sestávat z konečně mnoha instrukcí, kterým není třeba rozumět, stačí je přesně dodržet; může je provádět neúnavný, pečlivý úředník, nebo ještě lépe stroj. K dispozici má přitom neomezeně mnoho místa na poznámky. Je-li uvažovaná funkce f pro nějaké dané $\overline{x} \in \mathbb{N}^k$ definována, potom procedura zahájená se vstupem $\overline{x} \in \mathbb{N}^k$ po provedení konečně mnoha instrukcí skončí, a jako svůj výstup oznámí číslo $f(\overline{x})$. Pokud naopak funkce f v bodě $\overline{x} \in \mathbb{N}^k$ definována není, požadujeme, aby procedura žádnou hodnotu nevrátila. Může se například dostat do nekonečné smyčky, rozhodně ale nesmí skončit s nějakou předstíranou hodnotou funkce f v bodě $\overline{x} \notin dom(f)$.

Taková představa algoritmu je na jednu stranu podstatně omezená tím, že je v několika ohledech *finitární*: instrukcí je konečně mnoho, vstupem jsou čísla, tedy objekty konečné povahy, a samotný výpočet též běží jen konečně dlouho; to jsou však přirozená omezení. Na druhou stranu je taková představa idealizovaná: pokud si stroj během výpočtu potřebuje zvlášť poznamenat každý atom ve vesmíru, má k tomu dostatek místa; pokud výpočet skončí spolu se zánikem Slunce, doběhl v konečném čase. Teorie rekurze se nezatěžuje omezeními fyzických počítačů, tedy časovými a paměťovými nároky, či snad takovými podružnostmi jako velikost a stáří vesmíru, a ptá se obecně, co je vůbec možné *vypočítat*. Hlavní formulace vyčíslitelnosti ostatně historicky předcházejí vzniku strojů, které by něco skutečně vyčíslovaly, o dnešním průmyslu nemluvě.

Informatik tak může považovat teorii rekurze za nejsvobodnější možné programování, omezené jen hranicemi toho, co vůbec přichází teoreticky v úvahu.

5.1 Turingovy stroje

Turingův stroj[1] ztělesňuje výše popsanou představu následovně. Pracuje na neomezené *pásce*, rozdělené na diskrétní pole; každé z těchto polí obsahuje buďto znak `0` nebo znak `1`. Stroj v každém okamžiku čte nebo píše na jedno z polí na pásce; dokáže přečíst znak, který se na tomto poli nachází, a dokáže jej přepsat jiným znakem. Během jedné instrukce se může pohnout vždy nanejvýš o jedno pole. V každém okamžiku se nachází v jednom z konečně mnoha předem daných vnitřních *stavů*. Chod stroje se řídí sadou konečně mnoha předem daných *instrukcí* tvaru `s r w m t`, kde `s,t` jsou vnitřní stavy, `r,w` jsou symboly `0` nebo `1`, a `m` je jeden ze symbolů `L,N,R`. Taková instrukce říká: pokud jsi ve stavu `s` a na pásce čteš symbol `r`, zapiš na pásku symbol `w`, posuň se na pásce doleva (`L`), nikam (`N`), nebo doprava (`R`), a přepni se do stavu `t`. Stroj začíná ve stavu, kterým začíná první instrukce. Pokud pro nějaký stav a vstup instrukce chybí, stroj se zastaví.

[1] A. M. Turing, *On Computable Numbers*, Proc. London Math. Soc. s2-42:1, 230–265

Takový stroj vyčísluje funkci $f : \mathbb{N}^k \to \mathbb{N}$ v následujícím smyslu: je-li na pásce před spuštěním stroje zapsána k-tice $\overline{x} \in \mathbb{N}^k$, pak v případě $\overline{x} \in dom(f)$ se stroj po konečně mnoha krocích zastaví a na pásce zanechá zapsanou hodnotu $f(\overline{x})$; v případě $\overline{x} \notin dom(f)$ se stroj nikdy nezastaví.

Je otázka, jakým způsobem má být na pásce, která smí obsahovat jen znaky 0 a 1, zapsáno přirozené číslo, respektive k-tice přirozených čísel. Nabízí se zapisovat čísla[2] v unární notaci: číslo x se zapíše jako sekvence $x + 1$ po sobě jdoucích jedniček; takovou sekvenci budeme v dalším značit 1^{x+1}. Tedy například páska ...0001111000... zachycuje číslo 3. Podobně k-tice čísel $x_1, \ldots, x_k$ se zapíše jako k-tice takových posloupností, oddělených navzájem nulou. Tedy například ...0001111010111000... zachycuje trojici $(3, 0, 2)$.

Přijmeme také následující konvenci: na začátku výpočtu stojí hlava na první jedničce takového zápisu a po skončení výpočtu (tedy pokud stroj vůbec zastaví) stojí hlava na první jedničce výsledku.[3]

5.1.1 Příklad. Je zvykem popisovat Turingův stroj právě jen jako sadu jeho instrukcí. Stavům lze přitom dávat různá sugestivní jména, jde ale jen o to, aby byly navzájem rozlišitelné. Následující stroj vyčísluje, při značení zavedeném výše, funkci *signum*, která kladným číslům přiřazuje jedničku a nule nulu. Vpravo od každé instrukce uvádíme stručný komentář.

```
A 1 1 R B // z první jedničky ukroč doprava
B 0 0 L Z // je-li tam nula, je argumentem i výsledkem nula
B 1 1 L P // je-li tam další jednička, je argument pozitivní
P 1 1 L P // jdi přes pozitivní argument zpět doleva
P 0 0 L S // překroč oddělující nulu a začni psát výsledek
S 0 1 L T // napiš první jedničku výsledku a ukroč vlevo
T 0 1 N Z // napiš druhou jedničku výsledku a zastav
```

Například na vstupu 2 (totiž 01110) se tento stroj zachová následovně. Vlevo uvádíme stav pásky s vyznačenou hlavou, vpravo použitou instrukci.

```
0000111000 // A11RB
0000111000 // B11LP
0000111000 // P11LP
0000111000 // P00LS
0000111000 // S01LT
0010111000 // T01NZ
0110111000
```

Všimněme si několika technických omezení v definici Turingova stroje. Na pásce připouštíme jen znaky 0 a 1; ekvivalentně bychom mohli říci, že každé pole na pásce buďto je nebo není prázdné. Za abecedu znaků na pásce bychom ve skutečnosti mohli přijmout jakoukoli konečnou možinu: každou takovou

[2] Jedná se o zásadní problém: předáváme nikoli samotné *číslo*, totiž matematický objekt, nýbrž nějaký jeho *zápis*, tedy výraz sestavený ze symbolů nějakého formálního jazyka, což není totéž. Uvědomme si, že ve stejné situaci jsme při rutinním sčítání: místo čísel pracujeme s jejich dekadickými zápisy. To nás nemusí nutně trápit, jinak než jazykem se beztak vyjádřit neumíme. Volba konkrétní podoby takového zápisu je naopak otázka čistě technická.

[3] Taková konvence je samozřejmě arbitrární: mohli bychom vyžadovat čtení zprava, čistit na pásce všechny poznámky z mezivýpočtů, apod. Pro naše účely je však přijatá konvence čtení vstupů a psaní výstupů stejně dobrá jako každá jiná.

můžeme za cenu další práce pomocí dvou znaků binárně kódovat. V instrukcích umožňujeme pohyb vlevo, vpravo, či nikam; za cenu dalších stavů bychom mohli vždy požadovat pohyb vlevo či vpravo. Stejně tak bychom mohli připouštět buďto jen pohyb nebo jen čtení či psaní, opět za cenu dalších instrukcí. Můžeme též předpokládat, že jeden stav je vždy vyhrazen jen pro ukončení běhu stroje. Těmito modifikacemi se výpočetní síla Turingových strojů nezmění.

Sadu instrukcí tvaru `s r w m t` pro Turingův stroj můžeme ekvivalentně nahlížet jako *přechodovou funkci* $(\mathtt{s},\mathtt{r}) \mapsto (\mathtt{w},\mathtt{m},\mathtt{t})$ z množiny $S \times \{0,1\}$ do množiny $\{0,1\} \times \{\mathtt{L},\mathtt{N},\mathtt{R}\} \times S$, kde S je množina vnitřních stavů; přitom stavy stačí rozlišovat přirozenými čísly. Turingův stroj pak můžeme formálně považovat přímo za takovou přechodovou funkci.

5.1.2 Definice. *Turingův stroj* je taková funkce M, ke které existuje $n \in \mathbb{N}$ splňující $dom(M) \subseteq \{0,\ldots,n\} \times \{0,1\}$ a $rng(M) \subseteq \{0,1\} \times \{L,N,R\} \times \{0,\ldots,n\}$.

Jelikož se jedná o *funkci*, je takový stroj *deterministický* — v každé situaci existuje nanejvýš jeden způsob, jak pokračovat. Jiné stroje zkoumat nebudeme. Zároveň nepožadujeme, aby funkce M byla totální, tj. všude definovaná. To odpovídá tomu, že pro některé situace stroj žádnou instrukci nemá.

Nyní bychom mohli zavést pojmy jako *obsah pásky*, *krok výpočtu* a podobně, a mluvit o strojích zcela formálně. Takový formalismus ale zavedeme teprve až to bude nutné: představa počítajícího stroje je uchopitelnější.

5.1.3 Definice. Řekneme, že Turingův stroj M *počítá funkci* f, pokud má následující vlastnost. Je-li M spuštěn s páskou obsahující $\overline{x} \in \mathbb{N}^k$, pak nastane právě jedna ze dvou možností: je-li $\overline{x} \in dom(f)$, potom se M po konečně mnoha krocích zastaví a na pásce zanechá číslo $f(\overline{x})$, dále vpravo je páska prázdná; je-li naopak $\overline{x} \notin dom(f)$, M se nikdy nezastaví. Pokud k dané funkci takový stroj existuje, řekneme, že je *Turingovsky vyčíslitelná* nebo stručněji *vyčíslitelná*.

Výše uvedený příklad tedy ukazuje, že funkce signum je vyčíslitelná. Zřejmě každá konstantní funkce je vyčíslitelná, připíše jen vpravo předem známý počet jedniček. Identitu $f(x) = x$ počítá prázdný stroj, který nemá žádné instrukce. Každá funkce s konečným definičním oborem je vyčíslitelná, instrukce vlastně jen citují konečnou tabulku hodnot. Snadno se nahlédne, že je-li f vyčíslitelná, pak i každá její konečná modifikace je vyčíslitelná.

5.1.4 Cvičení. Popište stroje[4] vyčíslující unární funkci následníka $s(x) = x+1$, konstantní funkci $C_5^3(x_1,x_2,x_3) = 5$, konečnou funkci $f(0,1) = 2, f(1,2) = 3$, binární sčítání $f(x,y) = x + y$ a projekci $\pi_2^3(x_1,x_2,x_3) = x_2$. Skoro stejně se postaví stroje pro ostatní konstanty $C_m^k(\overline{x}) = m$ a projekce $\pi_i^k(x_1,\ldots,x_k) = x_i$.

5.1.5 Příklad. Následující stroj kopíruje daný argument ještě jednou vpravo. Pracuje tak, že vlevo od argumentu položí zarážku, všechny jedničky postupně přenese vpravo, poté je opět doplní a nakonec zarážku smaže.

```
A 1 1 L A // ukroč doleva
A 0 1 R B // polož zarážku
B 0 0 R B // jdi doprava, přejdi případné dřívější nuly
B 1 0 R C // až narazíš na jedničku, zvedni ji a nes ji doprava
```

[4] Běh Turingových strojů lze emulovat např. s pomocí `https://github.com/janstary/tm`

```
C 1 1 R C // přejdi dosud nevyzvednuté jedničky
C 0 0 R D // překroč oddělovací nulu
D 1 1 R D // přejdi dříve položené jedničky
D 0 1 L E // na prvním volném místě polož nesenou jedničku
E 1 1 L E // otoč se a jdi doleva pro další jedničku
E 0 0 L F // překroč oddělující nulu
F 1 1 L G // je vlevo ještě něco?
F 0 1 L H // pokud ne, budeme končit
G 1 1 L G // pokud ano, jdi přes jedničky doleva
G 0 0 R B // u nuly se otoč vpravo, poneseš další jedničku
H 0 1 L H // pokud končíme, znovu napiš vyzvednuté jedničky
H 1 0 R Z // u zarážky se zastav, zarážku smaž a ukroč doprava
```

S použitím tohoto stroje snadno postavíme jiný stroj, který počítá funkci $2x$, funkci $3x$, nebo obecně součin. Vidíme, že stroje lze různě kombinovat, například řetězit jejich výpočty, a tím konstruovat složitější stroje pro složitější funkce.

Následující cvičení ukazují, že omezení přijatá v definici Turingova stroje nejsou podstatná: pásek i hlav může být více, abeceda může být bohatší, a pro zastavení stroje může být vyhrazen speciální stav. V dalším budeme tato pozorování využívat: s vícehlavým či vícepáskovým strojem nebo s bohatší abecedou může být snazší úlohu vyřešit (zároveň může takový stroj pracovat rychleji), přitom důkazy stačí podat jen pro stroje v původní podobě.

5.1.6 Cvičení. Napište Turingův stroj, který dané slovo v abecedě a, b, c přepíše do abecedy $0, 1$; na začátku přitom stojí na prvním písmeně, a první znak $|$ vpravo značí konec slova. Posloupnosti $00, 01, 10, 11$ kódují na pásce původní znaky $|, a, b, c$. Tedy při vstupu `abccba|` skončí stroj s páskou `01101111100100` Napište také stroj, který překládá opačně.

5.1.7 Cvičení. Ukažte na příkladě, že k Turingovu stroji, který má k dispozici dvě pásky, existuje Turingův stroj s jednou páskou, který dává na stejných vstupech stejné vystupy jako původní stroj. (Využijte např. sudá a lichá pole.)

5.1.8 Cvičení. Ukažte na příkladě, že k Turingovu stroji, který má na pásce k dispozici dvě hlavy, existuje Turingův stroj s jednou hlavou, který dává na stejných vstupech stejné vystupy jako původní stroj.

5.1.9 Cvičení. Každý Turingův stroj lze modifikovat tak, že na všech vstupech stále dává tytéž výstupy, ale navíc existuje právě jeden stav, ve kterém stroj nemá žádné instrukce, kdežto ve všech ostatní stavech instrukce má.

Naším záměrem je prozkoumat třídu Turingových strojů, potažmo vyčíslitelných funkcí, a později ukázat, že splývá s třídou částečných rekurzivních funkcí. Předně si všimněme, že zatímco množina všech funkcí má mohutnost kontinua, vyčíslitelných funkcí je jen spočetně mnoho: Turingových strojů je nanejvýš tolik, kolik je konečných výrazů ve spočetné abecedě.

5.1.10 Příklad. Předvedeme funkci $\Sigma : \mathbb{N} \to \mathbb{N}$, která není vyčíslitelná,[5] protože roste rychleji než všechny vyčíslitelné funkce. Pro dané $x \in \mathbb{N}$ uvažme

[5] T. Rado, *On Non-Computable Functions*, BSTJ (1962), 877–884

všechny Turingovy stroje, které mají právě x stavů; takových strojů je konečně mnoho. Pro každý z nich, který se zastaví při spuštění na prázdné pásce, uvažme počet jedniček, které po sobě na pásce zanechá; to je konečné číslo. Maximální takový počet označme $\Sigma(x)$. To je pro každé $x \in \mathbb{N}$ dobře definovaná hodnota, totiž maximum jisté konečné množiny čísel. Funkce $\Sigma : \mathbb{N} \to \mathbb{N}$ je tedy dobře definovaná; snadno se nahlédne, že je rostoucí. Ukážeme, že není vyčíslitelná.

Buď totiž $f : \mathbb{N} \to \mathbb{N}$ nějaká vyčíslitelná funkce. Ukážeme, že od jistého $x \in \mathbb{N}$ dále je $f(x) < \Sigma(x)$. Je-li f vyčíslitelná funkce, je vyčíslitelná i funkce $\max(f(2x), f(2x+1)) + 1$; buď M stroj, který ji vyčísluje, buď $n \in \mathbb{N}$ počet jeho stavů. Pro každé $x \in \mathbb{N}$ buď potom M_x stroj, který na prázdnou pásku napíše číslo x a dál se chová jako stroj M; tedy M_x se nad prázdnou páskou zastaví — nejprve napíše x, a nad tímto vstupem pak spočítá větší z hodnot $f(2x), f(2x+1)$. Přitom M_x má $x+n$ stavů, takže $f(2x), f(2x+1) < \Sigma(x+n)$. Pro $x \geq n$ je ovšem $\Sigma(x+n) \leq \Sigma(2x)$, neboť Σ je neklesající, takže dohromady máme $f(2x), f(2x+1) < \Sigma(2x)$ pro každé $x \geq n$.

Pro výpočet funkce Σ se přitom nabízí následující procedura: sepiš všechny stroje o x stavech (tento počet zřejmě velmi rychle roste, ale pro každé $x \in \mathbb{N}$ je konečný); vyřaď ty z nich, které nikdy nedoběhnou; ostatní spusť, a počkej, až doběhnou; přečti si maximální výsledek z těchto konečně mnoha. Problém je v druhém kroku: později uvidíme, že žádná efektivní procedura nedokáže rozhodovat o tom, zda daný stroj na daném vstupu vůbec doběhne.

5.1.11 Cvičení. Popište všechny stroje s jedním stavem a určete hodnotu $\Sigma(1)$. Najděte svědky pro $\Sigma(2) = 4, \Sigma(3) = 6, \Sigma(4) = 13$. Kolik je uchazečů na $\Sigma(5)$?

5.2 Rekurzivní funkce

V tomto oddíle představíme druhou hlavní formalizaci efektivní vyčíslitelnosti, totiž třídu rekurzivních funkcí. Její definice má jinou formu než třída Turingovsky vyčíslitelných funkcí: jisté elementární funkce, například sčítání a násobení, za efektivní prohlásíme, a budeme požadovat uzavřenost na jisté základní operace s funkcemi, například skládání. Tyto požadavky budeme postupně zesilovat a tím zachytíme čím dál širší třídu funkcí: elementární, primitivní, obecné a částečné rekurzivní funkce. O třídě částečných rekurzivních funkcí pak ukážeme, že se shoduje s třídou Turingovsky vyčíslitelných funkcí.

Elementární rekurzivní funkce Nejskromnější představa o efektivní vyčíslitelnosti bude zřejmě za vyčíslitelné považovat základní aritmetické operace sčítání a násobení; pro ty jistě existují efektivní vyčíslující procedury, známe je ze školy. Přirozená je též představa, že dokážeme efektivně rozhodnout o rovnosti, vybrat daný prvek z daných konečně mnoha (třeba pátý z devíti), že se efektivní funkce dají skládat, a že dokážeme efektivně sečíst a násobit konečnou řadu. Tato představa vede k následující definici.

5.2.1 Definice. Třída *elementárních rekurzivních funkcí* je nejmenší třída funkcí, která obsahuje binární sčítání a násobení, charakteristickou funkci rovnosti, všechny *projekce* $\pi_i^k(x_1, \ldots, x_k) = x_i$ za každé $1 \leq i \leq k \in \mathbb{N}$, a je uzavřena na:

(a) *skládání*: jsou-li $f, g_1, \ldots, g_n \in \mathcal{ER}$, kde f je n-ární funkce a g_i jsou k-ární, pak do $\mathcal{ER}$ padne i složená k-ární funkce $h(\overline{x}) = f(g_1(\overline{x}), \ldots, g_n(\overline{x}))$;

(b) *sumace*: s každou k-ární funkcí $f \in \mathcal{ER}$ padne do $\mathcal{ER}$ i k-ární funkce $\sum f$, jejíž hodnotou v bodě $x_1, \ldots, x_k$ je $\sum \{f(x_1, \ldots, x_{k-1}, y); y < x_k\}$;

(c) *multiplikace*: s každou k-ární $f \in \mathcal{ER}$ padne do $\mathcal{ER}$ i k-ární funkce $\prod f$, jejíž hodnotou v bodě $x_1, \ldots, x_k$ je $\prod \{f(x_1, \ldots, x_{k-1}, y); y < x_k\}$.

Třídu elementárních rekurzivních funkcí budeme značit $\mathcal{ER}$ a budeme někdy stručněji říkat jen *elementární funkce*. Snadno se nahlédne, že uvedené uzávěrové vlastnosti se zachovávají při průniku, takže $\mathcal{ER}$ je právě průnik všech tříd s těmito vlastnostmi. Při sumaci a multiplikaci přijímáme obvyklou konvenci, podle které je prázdná suma nulová a prázdný součin jednotkový; tedy $(\sum f)(0) = 0$ a $(\prod f)(0) = 1$ pro každou funkci f.

5.2.2 Příklad. Každá z následujících funkcí je elementární:

(i) k-ární konstanta $C^k_m(x_1, \ldots, x_k) = m$, pro každé $k, m \in \mathbb{N}$;

(ii) unární *signum* s hodnotami $sign(0) = 0$ a $sign(x) = 1$ pro $x > 0$;

(iii) unární funkce $\overline{sign}(x)$ s hodnotami opačnými než u funkce $sign(x)$;

(iv) unární *následník* s hodnotami $s(x) = x + 1$ pro každé x;

(v) binární *mocnina* x^y (kde $x^0 = 1$ pro každé x a $0^y = 0$ pro každé $y > 0$);

(vi) unární *faktoriál* $x!$ (přičemž $0! = 1$);

(vii) každý polynom s přirozenými koeficienty.

Funkce $\overline{sign}(x)$ vrací 1 jen pro $x = 0$, což je jediné $x \in \mathbb{N}$ splňující $x = x+x$; získáme ji tedy složením $\chi_=(x, x+x)$, tj. formálněji $\chi_=(\pi^1_1(x), +(\pi^1_1(x), \pi^1_1(x)))$. Funkce $sign(x)$ je potom $\overline{\overline{sign}(x)}$. Relace $\neq$ má charakteristickou funkci $\overline{sign}(\chi_=(x, y))$. Konstantní unární nula $C^1_0(x)$ je funkce $\chi_{\neq}(x, x)$, konstantní unární jednička je $C^1_1(x) = \overline{sign}(C^1_0(x))$. Indukcí podle m se pak ukáže, že i každou další unární konstantu získáme složením jako $C^1_{m+1}(x) = C^1_m(x) + C^1_1(x)$; konstanty C^k_m pak získáme jako složení $C^k_m(x_1, \ldots, x_k) = C^1_m(\pi^k_1(x_1, \ldots, x_k))$. Následník je $s(x) = \pi^1_1(x) + C^1_1(x)$, mocnina je $x^y = \prod_{t<y} \pi^2_1(x, t)$ a faktoriál je $x! = \prod_{y<x} s(y)$. Polynom je z definice součtem násobků mocnin.

5.2.3 Cvičení. Je-li f elementární, je elementární i každá její konečná modifikace. Tedy například funkce, která je odněkud dále konstantní, je elementární. Podobně pro funkci, která je odněkud dále periodická.

5.2.4 Příklad. Buď f unární funkce s hodnotami $f(x) = 1$, pokud platí Goldbachova hypotéza, a $f(x) = 0$ jinak. Goldbachova hypotéza je dosud otevřený problém, nicméně funkce f je buďto konstanta C^1_0 nebo konstanta C^1_1; v každém případě je elementární. Smysl této poznámky je následující: efektivní vyčíslitelnost je vlastností funkce samotné, nikoli vlastnost jejího popisu. Po rozhodnutí Goldbachovy hypotézy se na funkci f nic nezmění.

5.2.5 Definice. Řekneme, že $R \subseteq \mathbb{N}^k$ je *elementární* nebo obšírněji *elementární rekurzivní relace*, právě když její charakteristická funkce χ_R je elementární.

Triviálními příklady elementárních relací jsou kromě rovnosti a nerovnosti též $\emptyset \subseteq \mathbb{N}$ s charakteristickou funkcí C_0^1 a $\mathbb{N} \subseteq \mathbb{N}$ s charakteristickou funkcí C_1^1. Také každá jednoprvková $\{n\} \subseteq \mathbb{N}$ je elementární, její charakteristická funkce je $\chi_=(\pi_1^1(x), C_n^1(x))$. Podle následujícího pozorování je třída elementárních relací uzavřená na základní množinové operace. To znamená, že každá konečná i každá kokonečná podmnožina $\mathbb{N}$ je elementární.

5.2.6 Lemma. *Buďte $R, S \subseteq \mathbb{N}^k$ elementární relace. Potom také relace $\mathbb{N}^k \setminus R$, $R \cap S$ a $R \cup S$ jsou elementární.*

Důkaz. Pro průnik máme $\chi_{R\cap S}(\overline{x}) = \chi_R(\overline{x}) \cdot \chi_S(\overline{x})$, pro doplněk $\chi_{\mathbb{N}^k\setminus R}(\overline{x}) = \overline{sign}(\chi_R(\overline{x}))$; uzavřenost na sjednocení plyne z deMorganových formulí. □

5.2.7 Příklad. Binární relace $<, \leq, =, \neq, \geq, >$ jsou elementární. O relacích $=$ a $\neq$ to již víme. Relace $x \geq y$ platí právě když se x nerovná žádnému $t < y$, takže její charakteristickou funkcí je $\prod_{t<y} \chi_{\neq}(x, t)$, včetně případu $y = 0$. Relace $<$ je jejím doplňkem; sjednocením relací $<$ a $=$ je $\leq$, a relace $>$ je jejím doplňkem.

5.2.8 Lemma. *Buď $R \subseteq \mathbb{N}^n$ elementární relace, buďte $f_1, \ldots, f_n$ elementární. Potom i relace $\{\overline{x}; (f_1(\overline{x}), \ldots, f_n(\overline{x})) \in R\} \subseteq \mathbb{N}^k$ je elementární.*

Důkaz. Charakteristická funkce takové relace je složená z funkcí f_i a χ_R. □

Podle právě dokázaného lemmatu lze do $\mathcal{ER}$ relací dosazovat hodnoty $\mathcal{ER}$ funkcí. Důležitým případem takových relací jsou rovnosti a nerovnosti funkcí: pro $f, g \in \mathcal{ER}$ jsou i relace $\{\overline{x}; f(\overline{x}) = g(\overline{x})\}$ a $\{\overline{x}; f(\overline{x}) < g(\overline{x})\}$ elementární.

V příkladech jsme explicitně popsali, jak daná elementární funkce vznikne „zdola“ pomocí uzávěrových operací aplikovaných na jednodušší funkce. Takový „rodokmen“ lze sestavit pro každou elementární funkci. Totiž funkce h je elementární, právě když existuje konečná posloupnost elementárních funkcí $h_1, \ldots, h_m$ taková, že h_m je funkce h, a pro každou h_j platí jedna z následujících podmínek: h_j je sčítání nebo násobení nebo některá z projekcí π_i^k; nebo existuje n-ární funkce $h_l, l < j$ a n-tice k-árních funkcí $h_{j_1}, \ldots, h_{j_n}, j_\iota < j$ tak, že h_j je k-ární složená funkce $h_j(\overline{x}) = h_l(h_{j_1}(\overline{x}), \ldots, h_{j_n}(\overline{x})$; nebo existuje nějaká $h_l, l < j$ tak, že $h_j = \sum h_l$; nebo existuje nějaká $h_l, l < j$ tak, že $h_j = \prod h_l$.

Jako důsledek získáváme, že třída elementárních funkcí a relací je spočetná. Totiž základních elementárních funkcí je spočetně mnoho, uzávěrové operace jsou tři, a každé elementární funkci odpovídá nějaká konečná posloupnost popsaná výše. To znamená, že $|\mathcal{ER}| \leq |\omega^{<\omega}| = \omega$.

Třída $\mathcal{ER}$ má i další uzávěrové vlastnosti kromě těch, které zaručuje definice: pokud například u dané elementární funkce permutujeme, opakujeme, či ignorujeme proměnné, ze třídy $\mathcal{ER}$ tím nevybočíme.

5.2.9 Lemma. *Je-li funkce $f(x_1, \ldots, x_k)$ elementární, jsou elementární i funkce $(x_1, \ldots, x_k, x_{k+1}) \mapsto f(x_1, \ldots, x_k)$, funkce $(x_1, \ldots, x_{k-1}) \mapsto f(x_1, \ldots, x_{k-1}, x_1)$ a funkce $(x_1, \ldots, x_k) \mapsto f(x_{\iota(1)}, \ldots, x_{\iota(k)})$, kde ι je libovolná permutace indexů.*

Každá z uvažovaných funkcí se složí z funkce f a příslušných projekcí. Lemma se pak snadno zobecní na více než jednu proměnnou: je-li například funkce $f(x, y, z)$ elementární, budou elementární i funkce $g(x) = f(x, x, x)$ a $h(p, q, r, s) = f(q, r, p)$. Těchto obratů budeme dále bez komentáře využívat.

Je-li $R \subseteq \mathbb{N}^k$ elementární relace, tedy pokud lze, podle naší představy, efektivně rozhodovat o vztahu $R(x_1, \ldots, x_k)$, mělo by být možné efektivně rozhodovat i o tom, zda je $R(x_1, \ldots, x_{k-1}, y)$ pro každé $y < x_k$, nebo alespoň nějaké: v obou případech stačí vyzkoušet konečně (omezeně) mnoho hodnot. Podle následujícího lemmatu je tomu skutečně tak; říkáme, že třída elementárních relací je uzavřená na *omezenou kvantifikaci*.

5.2.10 Lemma. *Pro relaci $R \subseteq \mathbb{N}^k$ buď $E(R) \subseteq \mathbb{N}^k$ relace obsahující taková $(x_1, \ldots, x_k)$, že pro nějaké $y < x_k$ je $R(x_1, \ldots, x_{k-1}, y)$; podobně buď $A(R) \subseteq \mathbb{N}^k$ relace obsahující taková $(x_1, \ldots, x_k)$, že pro všechna $y < x_k$ je $R(x_1, \ldots, x_{k-1}, y)$. Potom je-li R elementární, jsou i $A(R), E(R)$ elementární.*

Důkaz. Pro $A(R)$ je $\chi_{A(R)}(x_1, \ldots, x_k) = \prod_{y<x_k} \chi_R(x_1, \ldots, x_{k-1}, y)$, a relace $E(R)$ je doplňkem $A(\mathbb{N}^k \setminus R)$. Tím jsme díky uzavřenosti na doplňky hotovi. □

Ze stejného důvodu by pro elementární relaci $R \subseteq \mathbb{N}^k$ měla být elementární i funkce $\mu[y < x_k]R(x_1, \ldots, x_{k-1}, y)$, která vrací nejmenší $y < x_k$ splňující $R(x_1, \ldots, x_{k-1}, y)$, pokud existuje, a v opačném případě vrací x_k. Podle následujícího lemmatu je tomu skutečně tak. Říkáme, že třída elementárních funkcí je uzavřená na *omezenou minimalizaci*.

5.2.11 Lemma. *Funkce $\mu[y < x_k]R(x_1, \ldots, x_{k-1}, y)$ je elementární, kdykoli je $R(x_1, \ldots, x_k)$ elementární relace.*

Důkaz. Elementární funkce $\overline{sign}(\sum_{z<y} \chi_R(x_1, \ldots, x_{k-1}, z))$ vrací 1 na právě takových $(x_1, \ldots, x_{k-1}, y)$, pro která $R(x_1, \ldots, x_{k-1}, z)$ neplatí pro žádné $z < y$; označme tuto funkci jako $g(x_1, \ldots, x_{k-1}, y)$. Potom $\mu[y < x_k]R(x_1, \ldots, x_{k-1}, y)$ je právě funkce $\sum_{y<x_k} g(x_1, \ldots, x_{k-1}, s(y))$: přičítá jedničku za každý index, pod kterým všechna $R(x_1, \ldots, x_{k-1}, y)$ selhávají. □

Podobně se ukáže uzavřenost na *omezenou maximalizaci*. Je-li $R \subseteq \mathbb{N}^k$, buď $\overline{\mu}[y < x_k]R(x_1, \ldots, x_{k-1}, y)$ funkce, která vrací největší $y < x_k$ splňující $R(x_1, \ldots, x_{k-1}, y)$, pokud existuje, a v opačném případě vrací 0.

5.2.12 Lemma. *Funkce $\overline{\mu}[y < x_k]R(x_1, \ldots, x_{k-1}, y)$ je elementární, kdykoli $R(x_1, \ldots, x_k)$ je elementární relace.*

Důkaz. Největší $y < x_k$ splňující $R(x_1, \ldots, x_{k-1}, y)$ je právě nejmenší takový index $y < x_k$, pro který platí $(\forall z < x_k)(y < z \to \neg R(x_1, \ldots, x_{k-1}, z))$. □

Následující lemma říká, že rozložíme-li definiční obor na konečně mnoho efektivně rozpoznatelných částí, můžeme efektivní funkci definovat po částech.

5.2.13 Lemma. *Buďte $R_1, \ldots, R_n$ elementární relace, tvořící rozklad $\mathbb{N}^k$, a buďte $f_1, f_2, \ldots, f_n$ elementární k-ární funkce. Položme $f(\overline{x}) = f_i(\overline{x})$ právě když $\overline{x} \in R_i$. Potom f je elementární funkce.*

Důkaz. Předně f je dobře definovaná: $\bigcup R_i = \mathbb{N}^k$ je rozklad, takže nastane právě jedna z možností. Přitom $f(\overline{x}) = \chi_{R_1}(\overline{x}) \cdot f_1(\overline{x}) + \cdots + \chi_{R_n}(\overline{x}) \cdot f_n(\overline{x})$. □

5.2.14 Příklad. Následující funkce a relace jsou elementární:

(i) k-ární minimum: $\min(x_1, \ldots, x_k)$;

(ii) k-ární maximum: $\max(x_1, \ldots, x_k)$;

(iii) $x \dot{-} y = x - y$ pro $x \geq y$, jinak $x \dot{-} y = 0$;

(iv) předchůdce $p(x) = x - 1$ pro $x > 0$, přičemž $p(0) = 0$;

(v) binární odčítání $|x - y|$;

(vi) binární dělení $[x/y]$ = největší $z \leq x/y$ pokud $y \neq 0$, jinak 0;

(vii) $rm(x, y)$ = zbytek po dělení x/y pro $y > 0$, jinak 0;

(viii) relace dělitelnosti $\{(x, y); x \text{ dělí } y\} \subseteq \mathbb{N}^2$;

(ix) množina prvočísel $\mathbb{P} \subseteq \mathbb{N}$.

Binární $\min(x, y)$ je x pokud $x \leq y$, a y jinak. Pro více argumentů podobně, jenom případů (vzájemných nerovností) je více. Maxima se ukáží analogicky. Hodnota $x \dot{-} y$ je největší $z \leq x$ takové, že $x = y + z$, včetně případu $x \dot{-} y = 0$, kdy takové z neexistuje. Předchůdce je $p(x) = x \dot{-} 1$, a $|x - y|$ je $(x \dot{-} y) + (y \dot{-} x)$.

Podíl $[x/y]$ pro $y \neq 0$ je největší $z \leq x$ takové, že $x \geq y \cdot z$. Zbytek $rm(x, y)$ pro $y \neq 0$ je $x - y \cdot [x/y]$. Vztah $x|y$ platí právě když $(\exists q \leq y)(y = x \cdot q)$. Konečně $x \in \mathbb{P}$ právě když $1 < x$ a žádné jiné $y < x$ nedělí x.

5.2.15 Příklad. Označme $(n+1)$-té prvočíslo jako p_n (tj. $p_0 = 2$). Potom unární funkce $n \mapsto p_n$ je elementární. Připomeňme nejprve horní odhad $p_n \leq 2^{2^n}$. Pro $p_0 = 2$ odhad platí, dále indukcí: číslo $p_0 \cdot p_1 \cdot \cdots \cdot p_n$ je dělitelné všemi prvočísly do p_n, takže $p_0 \cdot p_1 \cdot \cdots \cdot p_n - 1$ není dělitelné žádným provočíslem do p_n, v jeho prvočíselném rozkladu tedy musí figurovat nějaké další prvočíslo, takže $p_{n+1} \leq p_0 \cdot p_1 \cdot \cdots \cdot p_n - 1$ a podle indukčního předpokladu máme[6] také $p_{n+1} \leq 2^{2^0} \cdot 2^{2^1} \cdot \cdots \cdot 2^{2^n} - 1 = 2^{\sum\{2^k; k \leq n\}} - 1 = 2^{2^{n+1}-1} - 1 \leq 2^{2^{n+1}}$. Relace $R = \{(n, p_n); n \in \mathbb{N}\}$ je elementární, neboť $(n, x) \in R$ právě když $x \in \mathbb{P}$ a $\sum_{y<x} \chi_{\mathbb{P}}(y) = n$. Tedy $p_n = \mu[x \leq 2^{2^n}]R(n, x)$.

5.2.16 Příklad. Pro čísla $x, i \in \mathbb{N}$ buď $(x)_i$ exponent provočísla p_i v prvočíselném rozkladu čísla x. Pro úplnost definujeme také $(0)_i = (1)_i = 0$. Potom $(x)_i$ je elementární funkce, neboť $(x)_i = \mu[y < x](p_i^y$ dělí x a p_i^{y+1} nedělí $x)$. Podobně buď l unární funkce, která číslu x přiřadí index $l(x)$ jeho největšího prvočíselného dělitele (pro úplnost položme též $l(0) = 0 = l(1)$). Z definice je pak $l(x) = \overline{\mu}[i < x](p_i$ dělí $x)$.

5.2.17 Příklad. S pomocí výše uvedených funkcí lze efektivně kódovat a dekódovat konečné posloupnosti čísel.[7] Přirozeným kandidátem na kód konečné posloupnosti $(x_1, \ldots, x_k)$ je číslo $\langle x_1, \ldots, x_k \rangle = \prod_{i<k} p_i^{x_i+1}$. Takový kód je jednoznačný[8] díky jednoznačnosti prvočíselných rozkladů. Kód $x \in \mathbb{N}$ se naopak jednoznačně dekóduje[9] jako posloupnost $((x)_1 \dot{-} 1, \ldots, (x)_{lx} \dot{-} 1) \in \mathbb{N}^k$.

Přirozeným operacím zkracování a řetězení konečných posloupností pak odpovídají tyto elementární operace s jejich kódy: funkce $x \restriction j = \prod_{i<j} p_i^{(x)_i}$

[6] Mnohem lepší odhad dává článek P. Erdős, *Beweis eines Satzes von Tschebyschef*, Acta Litt. Sci. Szeged. 5 (1932), 194–198: první prvočíslo nad x leží pod $2x$. Jde nám ale jen o to, že nějaký elementární horní odhad existuje, takže můžeme použít omezenou kvantifikaci.

[7] Při nějaké zvolené korespondenci mezi přirozenými čísly a symboly nějakého jazyka (podobně jako v ASCII tabulce) pak můžeme na přirozená čísla hledět jako na kódy slov, formulí, důkazů. Toho využijeme později při *aritmetizaci syntaxe*.

[8] Kódy $\langle 5, 9 \rangle$ a $\langle 5, 9, 0 \rangle$ rozliší právě $+1$ přidaná v exponentu.

[9] Ne každé $x \in \mathbb{N}$ je kódem, ale množina kódů je elementární.

zkrátí kód posloupnosti $(x_1, \ldots, x_k)$ na kód posloupnosti $(x_1, \ldots, x_j)$, a funkce $x^\frown y = x \cdot \prod_{i<l_y} p_{i+l_x}^{(y)_i}$ zřetězí kód x posloupnosti $(x_1, \ldots, x_{l_x})$ a kód y posloupnosti $(y_1, \ldots, y_{l_y})$ do kódu posloupnosti $(x_1, \ldots, x_{l_x}, y_1, \ldots, y_{l_y})$.

5.2.18 Cvičení. Ukažte, že funkce $f(m,n) = ((m+n)^2 + 3m + n)/2$ je bijekce. To znamená, že by se též dala použít pro jednoznačné kódování dvojic čísel, navíc její hodnoty rostou jen polynomiálně a každé číslo je kódem. Pro jednoduchost ale zůstaneme u kódování prvočíselnými rozklady.

Primitivní rekurzivní funkce Třídu elementárních rekurzivních funkcí nyní rozšíříme do třídy primitivních rekurzivních funkcí. Prostředkem k tomu je idea *primitivní rekurze*: hodnota funkce v nule je nějak pevně zvolena, a hodnota v každém dalším bodě $y+1$ se počítá z hodnoty v bodě y. Pro výpočet $h(3)$ potom stačí znát hodnotu $h(2)$, která se vypočítá z hodnoty $h(1)$, a ta se vypočítá z hodnoty $h(0)$, kterou známe. Ukážeme, že pomocí primitivní rekurze lze efektivně vyčíslit všechny elementární funkce, ale i některé další.

5.2.19 Definice. Třída *primitivních*, nebo obšírněji *primitivních rekurzivních funkcí*, kterou budeme v dalším značit $\mathcal{PR}$, je nejmenší třída funkcí, která obsahuje unární funkci *následníka* $s(x) = x + 1$, všechny *projekce* $\pi_i^k(x_1, \ldots, x_k) = x_i$, za každé $1 \leq i \leq k \in \mathbb{N}$, a je uzavřena na:

(a) *skládání*: jsou-li $f, g_1, \ldots, g_n \in \mathcal{PR}$, kde f je n-ární funkce a g_i jsou k-ární, pak do $\mathcal{PR}$ padne i složená k-ární funkce $h(\overline{x}) = f(g_1(\overline{x}), \ldots, g_n(\overline{x}))$;

(b) *primitivní rekurzi*: s každou k-ární funkcí $f \in \mathcal{PR}$ a každou $(k+2)$-ární funkcí $g \in \mathcal{PR}$ padne do $\mathcal{PR}$ i $(k+1)$-ární funkce h splňující $h(\overline{x}, 0) = f(\overline{x})$ a $h(\overline{x}, y+1) = g(\overline{x}, y, h(\overline{x}, y))$ pro každé $y \in \mathbb{N}$.

Hodnotu $h(\overline{x}, 0)$ určuje funkce $f(\overline{x})$, a každá další hodnota $h(\overline{x}, y+1)$ se z předchozí hodnoty $h(\overline{x}, y)$ a předchozích argumentů $\overline{x}, y$ (tedy „rekurzivně“) získá pomocí funkce g. Jako speciální případ primitivní rekurze pro $k = 0$ pak máme pro číslo $m \in \mathbb{N}$ (jakožto nulární funkci f) a binární funkci $g \in \mathcal{PR}$ též funkci $h \in \mathcal{PR}$ splňující $h(0) = m$ a $h(y+1) = g(y, h(y))$.

5.2.20 Příklad. Sčítání je primitivní, neboť $x + 0 = x$ a $x + s(y) = s(x + y)$; formálněji $x + 0 = \pi_1^1(x)$ a $x + (y+1) = s(\pi_3^3(x, y, x+y))$. Pro součin je $x \cdot 0 = 0$ a $x \cdot (y+1) = x \cdot y + x$; formálně $x \cdot 0 = C_0^1(x)$ a $x \cdot (y+1) = \pi_3^3(x, y, x \cdot y) + \pi_2^3(x, y, x \cdot y)$, přitom sčítání je primitivní a konstantní nulu lze též definovat rekurzí jako $C_0^1(0) = 0$ a $C_0^1(y+1) = \pi_2^2(y, C_0^1(y))$. Pomocí primitivní rekurze definujeme funkci předchůdce jako $p(0) = 0$ a $p(y+1) = y$; funkce $x \dot{-} y$ se potom definuje jako $x \dot{-} 0 = x$ a $x \dot{-} (y+1) = p(x \dot{-} y)$, funkce $\overline{sign}(x)$ jako $1 \dot{-} x$, nerovnost $\chi_{\leq}(x, y)$ jako $\overline{sign}(x \dot{-} y)$, nerovnost $\geq$ analogicky, a rovnost je jejich průnikem.

5.2.21 Příklad. Primitivní funkci lze zřejmě vystavět více než jedním způsobem; ve skutečnosti nekonečně mnoha způsoby, neboť každou takovou konstrukci můžeme uměle zkomplikovat. Daná konstrukce předpisuje, jak vyčíslit hodnotu funkce v každém bodě. Například hodnota součtu $3+2$ se podle právě uvedené definice sčítání jakožto primitivní funkce vypočte následovně:

$$\begin{aligned}
h(3,2) &= h(3,1+1)\\
&= s(\pi^3_3(3,1,h(3,1)))\\
&= s(\pi^3_3(3,1,h(3,0+1)))\\
&= s(\pi^3_3(3,1,s(\pi^3_3(3,0,h(3,0)))))\\
&= s(\pi^3_3(3,1,s(\pi^3_3(3,0,\pi^1_1(3)))))\\
&= s(\pi^3_3(3,1,s(\pi^3_3(3,0,3))))\\
&= s(\pi^3_3(3,1,s(3)))\\
&= s(\pi^3_3(3,1,4))\\
&= s(4)\\
&= 5
\end{aligned}$$

5.2.22 Věta. *Každá elementární funkce je primitivní.*

Důkaz. Sčítání, násobení a rovnost jsou pokryty předchozím příkladem. Projekce a uzavřenost na skládání máme z definice. Zbývá ukázat, že je $\mathcal{PR}$ uzavřená na sumaci a multiplikaci. Přitom suma $g = \sum f$ funkce $f \in \mathcal{PR}$ se získá jako $g(x_1,\ldots,x_{k-1},0) = 0$ a $g(x_1,\ldots,x_{k-1},y+1) = g(x_1,\ldots,x_{k-1},y) + f(x_1,\ldots,x_{k-1},y)$. Uzavřenost na multiplikaci se ukáže stejně. □

Následující uzávěrové vlastnosti třídy $\mathcal{PR}$ ukážeme stejně jako u třídy $\mathcal{ER}$.

5.2.23 Lemma. *Buďte $R, S \subseteq \mathbb{N}^k$ primitivní relace. Potom také relace $\mathbb{N}^k \setminus R$, $R \cap S$ a $R \cup S$ jsou primitivní. Je-li $R \subseteq \mathbb{N}^n$ primitivní relace a jsou-li $f_1,\ldots,f_n$ primitivní funkce, pak i relace $\{\overline{x}; (f_1(\overline{x}),\ldots,f_n(\overline{x})) \in R\} \subseteq \mathbb{N}^k$ je primitivní. Třída $\mathcal{PR}$ je uzavřena na omezenou kvantifikaci a omezenou minimalizaci.*

Pro danou funkci f můžeme pomocí kódování z 5.2.17 definovat funkci $\overline{f}$, která v hodnotě $\overline{f}(x_1,\ldots,x_k)$ kóduje celý dosavadní průběh $f(x_1,\ldots,x_{k-1},y)$ pro $y < x_k$, totiž funkci

$$\overline{f}(x_1,\ldots,x_k) = \prod_{y<x_k} p_y^{f(x_1,\ldots,x_{k-1},y)+1}.$$

Například pro funkci následníka je $\overline{s}(4) = \langle s(0), s(1), s(2), s(3)\rangle \in \mathbb{N}$ kód posloupnosti $(s(0), s(1), s(2), s(3))$. Podle následující věty zůstává funkce $\overline{f}$ ve třídě elementárních, respektive primitivních funkcí.

5.2.24 Věta. *Pro každou funkci platí:*

(i) *$f \in \mathcal{ER}$ právě když $\overline{f} \in \mathcal{ER}$.*

(ii) *$f \in \mathcal{PR}$ právě když $\overline{f} \in \mathcal{PR}$.*

Důkaz. (i) Je-li funkce $f(x_1,\ldots,x_k)$ elementární, je i funkce $p_y^{f(x_1,\ldots,x_{k-1},y)+1}$ elementární, neboť je složená z mocniny, přičítání jedničky, a funkcí f a p, které jsou všechny elementární. Přitom $\overline{f}$ je právě multiplikací této funkce. Naopak je-li $\overline{f}$ elementární, získáme z ní hodnoty funkce f elementárním dekódováním jako $f(x_1,\ldots,x_k) = (\overline{f}(x_1,\ldots,x_k+1))_{x_k}$. (ii) Pro funkci f lze funkci $\overline{f}$ definovat primitivní rekurzí jako $\overline{f}(\overline{x},0) = 1$ a dále $\overline{f}(\overline{x},y+1) = \overline{f}(\overline{x},y)\cdot p_y^{f(\overline{x},y)+1}$. Dekódování se provede stejně jako v prvním případě. □

Podle následující věty se můžeme při výpočtu hodnoty $f(x_1, \ldots, x_k, y+1)$ primitivní rekurzí odkazovat nejen na předcházející argument $x_1, \ldots, x_k, y$ a předchozí hodnotu $f(x_1, \ldots, x_k, y)$, ale dokonce na celý dosavadní průběh.

5.2.25 Věta. *Buď f libovolná k-ární funkce. Existuje-li $(k+1)$-ární primitivní funkce g tak, že $f(\overline{x}, y) = g(\overline{x}, y, \overline{f}(\overline{x}))$ pro každé $\overline{x}, y \in \mathbb{N}^k$, pak i f je primitivní.*

Důkaz. Z definice je $\overline{f}(x_1, \ldots, x_{k-1}, 0) = 1$ a dále podle předpokladu platí $\overline{f}(\overline{x}, y+1) = \overline{f}(\overline{x}, y) \cdot p_y^{f(\overline{x},y)+1} = \overline{f}(\overline{x}, y) \cdot p_y^{g(\overline{x},y,\overline{f}(\overline{x},y))+1}$, takže $\overline{f}$ je primitivní. Podle předchozí věty je tedy i funkce f primitivní. □

Třída $\mathcal{PR}$ je z definice uzavřena na primitivní rekurzi, narozdíl od $\mathcal{ER}$. Nabízí se otázka, zda je $\mathcal{ER}$ uzavřena alespoň na nějakou formu rekurze. Podle následující věty je uzavřena na *omezenou rekurzi*.

5.2.26 Věta. *Buďte $f, g \in \mathcal{ER}$, buď h sestavena z funkcí f, g primitivní rekurzí, tj. $h(\overline{x}, 0) = f(\overline{x})$ a $h(\overline{x}, y+1) = g(\overline{x}, y, h(\overline{x}, y))$. Pokud existuje elementární funkce u taková, že $h \leq u$, pak i funkce h je elementární.*

Důkaz. Pro $x_1, \ldots, x_k, x_{k+1} \in \mathbb{N}$ uvažme průběh $h(\overline{x}, 0), h(\overline{x}, 1), \ldots, h(\overline{x}, x_{k+1})$. To je konečná posloupnost, označme ji $t = (t_i)$ a položme $c = \prod_{i<x_{k+1}} p_i^{t_i+1}$. Kód c splňuje $(c)_0 = f(\overline{x})$ a $(c)_{i+1} = g(\overline{x}, i, (c)_i)$ pro každé $i < x_{k+1}$.

Pro $i < x_{k+1}$ je $t_i = h(\overline{x}, i) \leq u(\overline{x}, i) \leq \sum_{i<x_{k+1}} u(\overline{x}, i)$, takže $c \leq p_{x_{k+1}}^{q(\overline{x},x_{k+1})}$, kde $q(\overline{x}, x_{k+1}) = (1+x_{k+1}) \cdot \sum_{i<x_{k+1}} u(\overline{x}, i)$. Jde o to, že máme horní $\mathcal{ER}$ odhad hodnot funkce h i kódu c, takže budeme moci použít omezenou minimalizaci.

Buď R relace obsahující taková $(\overline{x}, x_{k+1}, y)$, pro která existuje $c \leq p_{x_{k+1}}^{q(\overline{x},x_{k+1})}$ splňující podmínky výše a navíc $y = (c)_{x_{k+1}}$. Potom R je elementární relace, takže stačí ukázat $h(\overline{x}, x_{k+1}) = \mu[y \leq u(\overline{x}, x_{k+1})]R(\overline{x}, x_{k+1}, y)$. Přitom pro $\overline{x}, x_{k+1} \in \mathbb{N}$ je $R(\overline{x}, x_{k+1}, h(\overline{x}, x_{k+1}))$ a pro $R(\overline{x}, x_{k+1}, y)$ je $y = h(\overline{x}, x_{k+1})$. □

Na druhou stranu ukážeme, že třída $\mathcal{ER}$ není uzavřena na primitivní rekurzi, takže $\mathcal{ER} \subset \mathcal{PR}$: iterovaná mocnina roste rychleji než všechny $\mathcal{ER}$ funkce.

5.2.27 Příklad. Položme $h(x, 0) = x$ a $h(x, y+1) = x^{h(x,y)}$. Přímo z definice je $h \in \mathcal{PR}$, a snadno dokážeme následující odhady. (i) Pro každé $x, y \in \mathbb{N}$ je $x \leq h(x, y)$. Pro $x = 0$ jistě, pro $x > 0$ dokazujeme indukcí podle y. Pro $y = 0$ je $x = h(x, 0)$; máme-li $x \leq h(x, y)$, je také $x \leq x^x \leq x^{h(x,y)} = h(x, y+1)$. (ii) Pro $x > 1$ a každé y je $h(x, y) < h(x, y+1)$, neboť $h(x, y) < x^{h(x,y)} = h(x, y+1)$. (iii) Pro každé x, y je $h(x, y) < h(x+1, y)$. Dokazujeme indukcí podle y: pro $y = 0$ je $h(x, 0) = x < x+1 < h(x+1, 0)$, a pokud pro $y > 0$ již máme $h(x, y) < h(x+1, y)$, je také $h(x, y+1) = x^{h(x,y)} \leq (x+1)^{h(x,y)} < (x+1)^{h(x+1,y)} = h(x+1, y+1)$. Funkce h tedy roste v obou proměnných.

Pro součty, součiny a mocniny hodnot na nižších argumentech pak dostaneme následující. (iv) Pro $x > 1$ je $h(x, y) + h(x, z) < h(x, \max(y, z) + 1)$. Totiž $h(x, y) + h(x, z) \leq 2 \cdot h(x, max(y, z)) \leq 2^{h(x,max(y,z))} \leq x^{h(x,max(y,z))} = h(x, max(y, z)+1)$. (v) Pro $x > 1$ je $h(x, y) \cdot h(x, z) < h(x, \max(y, z) + 1)$. Totiž $h(x, y) \cdot h(x, z) \leq h(x, max(y, z))^2 = (x^{h(x,max(y,z)-1)})^2 = x^{2 \cdot h(x,max(y,z)-1)} \leq x^{exp(2,h(x,max(y,z)-1))} \leq x^{h(x,max(y,z))} = h(x, max(y, z) + 1)$. (vi) Pro $x > 1$ je $h(x, y)^{h(x,z)} < h(x, \max(y, z) + 2)$. Pro $y = 0$ máme dokonce $h(x, 0)^{h(x,z)} = x^{h(x,z)} = h(x, z+1)$, a pro $y > 0$ dostáváme $h(x, y)^{h(x,z)} = x^{h(x,y-1) \cdot h(x,z)} \leq x^{h(x,max(y-1,z)+1)} = h(x, max(y, z+1)+1) \leq h(x, max(y, z) + 2)$. Vidíme, že

h roste v druhém argumentu velice rychle: součty, součiny i mocniny předchozích hodnot přeroste už jen zvednutím druhého argumentu o 1 nebo 2. Konečně i pro iteraci platí (vii) pro $x > 1$ je $h(h(x,y),z) \leq h(x,y+2z)$. Dokazujeme indukcí podle z. Pro $z = 0$ je $h(h(x,y),0) = h(x,y) = h(x,y+2\cdot 0)$. Pokud pro $z > 0$ již máme $h(h(x,y),z) \leq h(x,y+2z)$, je také $h(h(x,y),z+1) = h(x,y)^{h(h(x,y),z))} \leq h(x,y)^{h(x,y+2z))} \leq h(x,max(y,y+2z)+2) = h(x,y+2(z+1))$.

S pomocí těchto odhadů můžeme nyní ukázat následující: pro každou k-ární funkci $f \in \mathcal{ER}$ existuje $y \in \mathbb{N}$ takové, že pokud $x = \max\{x_1,\ldots,x_k\} > 1$, je $f(x_1,\ldots,x_k) < h(x,y)$. To se ukáže indukcí podle složitosti $f \in \mathcal{ER}$: třída funkcí, pro které takové $y \in \mathbb{N}$ existuje, obsahuje sčítání, násobení, všechny projekce, a je uzavřená na skládání, sumaci i multiplikaci. Obsahuje tedy všechny elementární funkce.[10]

Sama funkce h pak není elementární: pokud ano, je elementární i $e(x) = h(x,x)$; v tom případě má rank $y \in \mathbb{N}$ a je $e(x) < h(x,y)$ pro každé $x > 1$. Speciálně pro $x = y+2 > 1$ je pak $h(y+2,y+2) = e(y+2) < h(y+2,y)$, spor.

5.2.28 Cvičení. Určete rank mocniny x^y, faktoriálu $x!$, polynomu $\sum_{i<k} \alpha_i x^i$.

5.2.29 Cvičení. Všechny funkce z oddílu o elementárních funkcích se pomocí primitivní rekurze snadno definují i přímo – napište jejich formální definice.

Obecné rekurzivní funkce Další rozšíření třídy primitivních rekurzivních funkcí získáme tím, že zkoumanou třídu uzavřeme na *minimalizaci speciálních funkcí*. Funkce $g(\overline{x},y)$ je speciální, pokud pro každé $\overline{x} \in \mathbb{N}^k$ existuje $y \in \mathbb{N}$ takové, že $g(\overline{x},y) = 0$. Je-li g speciální, definujeme funkci f tak, že $f(\overline{x})$ je nejmenší takové $y \in \mathbb{N}$. To zachycuje představu předem neohraničeného *hledání*: narozdíl od třídy $\mathcal{PR}$, která je uzavřená na *omezenou* minimalizaci, nemáme nyní žádný horní odhad hledaného $f(\overline{x}) = y \in \mathbb{N}$; víme však předem, že existuje.

5.2.30 Definice. Řekneme, že $(k+1)$-ární funkce g je *speciální*, pokud pro každé $\overline{x} \in \mathbb{N}^k$ existuje $y \in \mathbb{N}$ takové, že $g(x,y) = 0$. Je-li g speciální funkce, buď $f(\overline{x})$ nejmenší $y \in \mathbb{N}$ splňující $g(x,y) = 0$. Funkci f pak značíme $\mu y(g(\overline{x},y) = 0)$ a říkáme, že vznikla *minimalizací* speciální funkce g.

5.2.31 Definice. Třída *obecných*, nebo obšírněji *obecných rekurzivních funkcí*, kterou budeme v dalším značit $\mathcal{OR}$, je nejmenší třída funkcí, která obsahuje unární funkci následníka $s(x) = x+1$, všechny projekce $\pi_i^k(x_1,\ldots,x_k) = x_i$, za každé $1 \leq i \leq k \in \mathbb{N}$, a je uzavřena na skládání, primitivní rekurzi, a minimalizaci speciálních funkcí.

Hledíme-li na funkci $g(\overline{x},y)$ jako na charakteristickou funkci $(k+1)$-ární relace $R = \{(\overline{x},y); g(\overline{x},y) = 0\}$, můžeme relaci R nazvat speciální, je-li g speciální; to je právě když pro každé $\overline{x} \in \mathbb{N}^k$ existuje $y \in \mathbb{N}$ tak, že $(\overline{x},y) \in R$.

[10] Na číslo $y \in \mathbb{N}$ zaručené tímto tvrzením můžeme hledět jako na *rank* funkce f: kolik úsilí musí funkce h vynaložit, aby přerostla danou f. Chceme vlastně ukázat, že každá elementární funkce je konečného ranku. Snadno ověříme, že sčítání, násobení i každá projekce je ranku 1. Necháváme na čtenáři, aby ověřil, že při skládání, sumaci i multiplikaci zůstane rank konečný (a jak přesně vzroste). Podmínku $\max\{x_1,\ldots,x_k\} > 1$ potřebujeme proto, že $h(1,y)$ naopak vůbec neroste; argumentů $x_1,\ldots,x_k$, které tuto podmínku nesplňují, je ale jen konečně mnoho.

Ekvivalentně tedy můžeme třídu $\mathcal{OR}$ definovat jako nejmenší třídu uzavřenou na primitivní operace a minimalizaci speciálních relací.

Každá primitivní rekurzivní funkce je z definice též obecná rekurzivní. Zároveň všechny obecné rekurzivní funkce jsou totální: následník a projekce jistě, skládáním a primitivní rekurzí dostáváme z totálních funkcí opět totální, a minimalizací totální speciální funkce vznikne totální funkce.

Nejprve ukážeme, že třída obecných rekurzivních funkcí je podstatně bohatší než třída primitivních funkcí: obsahuje například funkci, která svými hodnotami kóduje průběh každé primitivní funkce.

5.2.32 Definice. Buď $\mathcal{F}$ libovolná třída funkcí. Řekneme, že binární funkce u je *univerzální* pro unární funkce z $\mathcal{F}$, pokud pro každou unární funkci $f \in \mathcal{F}$ existuje $c \in \mathbb{N}$ takové, že pro všechna $x \in \mathbb{N}$ je $f(x) = u(c, x)$.

5.2.33 Věta. *Existuje $\mathcal{OR}$ funkce, která je univerzální pro unární $\mathcal{PR}$ funkce.*

Důkaz. Popíšeme způsob, jak zakódovat danou primitivní funkci f číslem $c \in \mathbb{N}$: číslo $(c)_0$ bude kódovat počet argumentů, další exponenty pak jednotlivé případy rekurzivní výstavby $\mathcal{PR}$ funkce (ve smyslu definice 5.2.19) z jednodušších funkcí. Hledanou univerzální funkci u nakonec definujeme tak, aby $u(c, x)$ byla právě hodnota $\mathcal{PR}$ funkce s kódem c v bodě x.

Zvolíme nejprve kódy jednotlivých případů výstavby primitivní funkce. Následníka bude kódovat číslo 2; projekci π_i^k číslo $2^k \cdot 3^i$; funkci složenou z vnější n-ární funkce s kódem q a k-árních funkcí s kódy $r_1, \ldots, r_n$ bude kódovat číslo $2^k \cdot 5^n \cdot 7^q \cdot p_4^{r_1} \cdots p_{n+3}^{r_n}$; funkci sestavenou primitivní rekurzí z k-ární funkce s kódem q pro hodnotu v nule a $(k+2)$-ární funkce s kódem r pro další hodnoty bude kódovat číslo $2^{k+1} \cdot 7^q \cdot 11^r$. Volba kódů je pochopitelně zcela arbitrární.[11] Podstatné je, že tyto kódy jsou rozlišitelné, dokonce elementárními prostředky.[12]

Indexující funkci $j(c, x)$ definujeme podle případů — tj. na právě zavedených kódech — jediným možným způsobem. Na číslo c hledíme jako na kód dané primitivní funkce, na číslo x jako na kód jejího vstupu $(x_1, \ldots, x_k)$. Pro kód následníka buď $j(2, x) = (x)_0 + 1$; na kódech projekcí buď $j(2^k \cdot 3^i, x) = (x)_i$; pro složené funkce položme $j(2^k \cdot 5^n \cdot 7^q \cdot p_4^{r_1} \cdots p_{n+3}^{r_n}, x) = j(q, p_1^{j(r_1,x)} \cdots p_n^{j(r_n,x)})$; konečně na kódech $c = 2^{k+1} \cdot 7^q \cdot 11^r$ funkcí získaných primitivní rekurzí klademe $j(c, x) = j(q, x)$ pro $(x)_k = 0$ a $j(c, x \cdot p_k^{y+1}) = j(r, x \cdot p_k^y \cdot p_{k+1}^{j(c,x \cdot p_k^y)})$; ve všech ostatních případech buď $j(c, x) = 0$.

Funkce $j(c, x)$ pak indexuje všechny $\mathcal{PR}$ funkce, v tomto smyslu: pro každou k-ární funkci $f \in \mathcal{PR}$ existuje $c \in \mathbb{N}$ takové, že pro $x_1, \ldots, x_k \in \mathbb{N}$ je $f(x_1, \ldots, x_k) = j(c, p_1^{x_1} \cdots p_k^{x_k})$. Dokazujeme indukcí podle složitosti. Pro následníka je $s(x) = x + 1 = j(2, 2^x)$; pro projekci π_i^k je $j(2^k \cdot 3^i, p_1^{x_1} \cdots p_k^{x_k}) = x_i$. Pro funkci složenou z vnější n-ární funkce f, ke které již známe index q, a vnitřních k-árních funkcí $g_1, \ldots, g_n$, ke kterým již známe indexy $r_1, \ldots, r_n$, označme

[11] Jako příklad uvažme binární sčítání. Funkce $h(x, y) = x + y$ vzniká primitivní rekurzí jako $h(x, 0) = \pi_1^1(x)$ a $h(x, y+1) = s(\pi_3^3(x, y, x+y))$ z unární funkce π_1^1 s kódem $2^1 \cdot 3^1$ a ternární funkce $s \circ \pi_3^3$, která sama je složená. Kódem následníka s je 2, kódem projekce π_3^3 je $2^3 \cdot 3^3$, takže ternární $s \circ \pi_3^3$ má kód $2^3 \cdot 5^1 \cdot 7^2 \cdot 11^{2^3 \cdot 3^3}$; binární sčítání má tedy kód $c = 2^2 \cdot 7^{2^1 \cdot 3^1} \cdot 11^{2^3 \cdot 5^1 \cdot 7^2 \cdot 11^{2^3 \cdot 3^3}} \in \mathbb{N}$.

[12] Například množina čísel $c \in \mathbb{N}$, která splňují $(c)_0 > 0, (c)_1 > 0, (c)_1 \leq (c)_0$ a $l(c) = 1$, je elementární; taková čísla jsou právě kódy všech projekcí.

$x = p_1^{x_1} \cdots p_k^{x_k}$ pro dané $x_1, \ldots, x_k \in \mathbb{N}$; z indukčního předpokladu je pak

$$\begin{aligned} j(2^k \cdot 5^n \cdot 7^q \cdot p_4^{r_1} \cdots p_{n+3}^{r_n}, x) &= j(q, p_1^{j(r_1,x)} \cdots p_n^{j(r_n,x)}) \\ &= j(q, p_1^{g_1(\overline{x})} \cdots p_n^{g_n(\overline{x})}) \\ &= f(g_1(\overline{x}), \ldots, g_n(\overline{x})) \end{aligned}$$

Konečně uvažme $(k+1)$-ární funkci h zkonstruovanou primitivní rekurzí z k-ární funkce f s kódem q a $(k+2)$-ární funkce g s kódem r. Pro dané $x_1, \ldots, x_k \in \mathbb{N}$ označme opět $x = p_1^{x_1} \cdots p_k^{x_k}$. Pro $(x)_k = 0$ potom z indukčního předpokladu máme $j(2^{k+1} \cdot 7^q \cdot 11^r, x) = j(q, x) = f(\overline{x}) = h(\overline{x}, 0)$, a v dalších případech

$$\begin{aligned} j(2^{k+1} \cdot 7^q \cdot 11^r, x \cdot p_k^{y+1}) &= j(r, x \cdot p_k^y \cdot p_{k+1}^{j(2^{k+1} \cdot 7^q \cdot 11^r, x \cdot p_k^y)}) \\ &= g(\overline{x}, y, h(\overline{x}, y)) \\ &= h(\overline{x}, y+1). \end{aligned}$$

Z právě dokázaného tvrzení plyne, že funkce $u(c, x) = j(c, 2^x)$ je univerzální pro třídu unárních[13] primitivních funkcí. Zbývá ukázat, že j (a tedy také u) je obecná rekurzivní funkce. Předně, funkce j formálně není definována žádným dosud popsaným způsobem rekurze; odvolává se na své vlastní hodnoty na nižších argumentech (snadno se nahlédne, že jednodušší funkce má menší kód), ale pro různé tvary argumentů různým způsobem. Ukážeme nicméně, že funkci j lze ekvivalentně zavést minimalizací speciální relace. Pro každé $c, x \in \mathbb{N}$ je totiž $j(c, x) = ((z)_{lz})_2$, kde z je nejmenší číslo takové, že $((z)_{lz})_0 = c, ((z)_{lz})_1 = x$, a pro každé $i \leq l(z)$ platí jedna z následujících podmínek.[14] Podmínky popisují jednotlivé případy výpočtu hodnot funkce j, potažmo jednotlivé způsoby konstrukce primitivní rekurzivní funkce, podle kterých jsme tyto případy definovali.

následník: $(z)_{i,0} = 2$ a $(z)_{i,2} = (z)_{i,1} + 1$

projekce: $l((z)_{i,0}) = 1$, $(z)_{i,0,1} \leq (z)_{i,0,0}$ a $(z)_{i,2} = (z)(i, 1, (z)_{i,0,1})$

skládání: $(z)_{i,0,0} \neq 0, (z)_{i,0,1} = 0, (z)_{i,0,2} \neq 0$, $l((z)_{i,0}) = (z)_{i,0,2} + 3$, existuje $j < i$ tak, že $(z)_{j,0} = (z)_{i,0,3}, l((z)_{j,1}) = (z)_{i,0,2}$ a $(z)_{j,2} = (z)_{i,2}$, a pro každé $k < (z)_{i,0,2}$ existuje nějaké $m < i$ takové, že $(z)_{m,0} = (z)_{i,0,k+4}$, $(z)_{j,1} = (z)_{i,1}$ a $(z)_{m,2} = (z)_{j,1,k}$.

rekurze: $(z)_{i,0,0} \neq 0 = (z)_{i,0,1} = (z)_{i,0,2}$, $(z)_{i,0,3} \neq 0 \neq (z)_{i,0,4}$, $(lz)_{i,0} = 4$, a nastává jeden z následujících případů (píšeme $n = (z)_{i,0,0}$)): $(z)_{i,1,n} = 0$ a pro nějaké $j < i$ je $(z)_{j,0} = (z)_{i,0,3}$, $(z)_{j,1} = (z)_{i,1}$ a $(z)_{j,2} = (z)_{i,2}$; nebo $(z)_{i,1,n} > 0$, tedy $(z)_{i,1} = t \cdot p_n$, a pro nějaká $j, k < i$ je $(z)_{j,0} = (z)_{i,0,4}$, $(z)_{j,1} = t \cdot p_{n+1}^{(z)_{k,2}}$, $(z)_{k,0} = (u)_{i,0}$, $(u)_{k,1} = t$, a $(z)_{j,2} = (z)_{i,2}$.

zbytek: neplatí nic z předchozího, a je $(z)_{i,2} = 0$.

[13] Omezení na unární funkce není podstatné: stejně tak bychom mohli ukázat, že ternární $v(c, x_1, x_2) = j(c, 2^{x_1} \cdot 3^{x_2})$ je univerzální pro binární primitivní funkce.

[14] Podmínky jsou technicky složité, ale idea je jednoduchá: číslo z kóduje konečný protokol postupného výpočtu hodnoty $j(a, b)$ z jednodušších případů $j(a_1, b_1), \ldots, j(a_n, b_n)$. Tyto předchozí případy kódujeme jako $(z)_i = 2^{a_i} \cdot 3^{b_i} \cdot 5^{j(a_i,b_i)}$ a číslo z je kódem celé posloupnosti $(z)_1, \ldots, (z)_{lz}$; specielně poslední člen $(z)_{lz}$ kóduje výsledné $a, b, j(a, b)$. Vzhledem k množství indexů píšeme $(((z)_i)_j)_k$ též jako $(z)_{i,j,k}$ nebo $(z)(i, j, k)$.

Z výše uvedeného plyne, že pro každé $c, x \in \mathbb{N}$ takové z existuje; to znamená, že minimalizujeme speciální relaci, takže j je obecná rekurzivní funkce. □

Popsali jsme univerzální funkci pro třídu primitivních funkcí. Podle následujícího lemmatu tato funkce sama nemůže být primitivní.

5.2.34 Lemma. *Buď u totální binární funkce, buď $\mathcal{F}$ třída funkcí uzavřená na elementární operace. Potom je-li u univerzální pro unární funkce z $\mathcal{F}$, je $u \notin \mathcal{F}$.*

Důkaz. Je-li $u \in \mathcal{F}$, je také $f(x) = u(x, x) + 1$ funkce z $\mathcal{F}$, a má tedy svůj index $c \in \mathbb{N}$ vůči u; to jest, pro každé $x \in \mathbb{N}$ je $f(x) = u(c, x)$. Speciálně pro $x = c$ je tedy $u(c, c) = f(c) = u(c, c) + 1$, spor. □

5.2.35 Cvičení. (a) Buď f primitivní ostře rostoucí unární funkce. Potom její obor hodnot je primitivní množina. (b) Buď f primitivní neklesající neomezená unární funkce. Potom její obor hodnot je obecná rekurzivní množina.

Částečné rekurzivní funkce Třídu obecných rekurzivních funkcí ještě nemůžeme dost dobře prohlásit za adekvátní formální protějšek intuitivní představy efektivně vyčíslitelných funkcí. Jsou k tomu přinejmenším dva důvody.

Považujeme-li za efektivní proceduru to, co jsme provedli při minimalizaci speciálních funkcí, měli bychom za efektivně vyčíslitelné přijmout i funkce, které vzniknou minimalizací jakýchkoli efektivních funkcí; vyčíslující procedura bude totiž stejná: zkoušej postupně hodnoty $y \in \mathbb{N}$ počínaje nulou a dívej se, zdali už je $g(\overline{x}, y) = 0$. Může se pak samozřejmě stát, že taková funkce nebude všude definovaná: pokud takové $y \in \mathbb{N}$ neexistuje, výpočet nikdy neskončí.

Ze třídy obecných rekurzivních funkcí lze zároveň vykročit diagonalizací: obecných rekurzivních funkcí je spočetně mnoho (jako u předchozích tříd), buď tedy f_n nějaké jejich očíslování. Při troše snahy můžeme takové očíslování provést dokonce efektivně, totiž procházet obecné rekurzivní funkce indukcí podle složitosti a jejich konstrukce kódovat, podobně jako v důkazu věty o univerzální funkci. Potom ale musíme za efektivní považovat i funkci definovanou předpisem $d(n) = f_n(n) + 1$; její efektivní vyčíslující procedura je nasnadě, totiž pro dané $n \in \mathbb{N}$ spočti hodnotu funkce f_n v bodě $n \in \mathbb{N}$ a přičti jedna. Funkce d se ale nemůže všude shodovat s žádnou f_n: kdyby d byla funkce f_n, měli bychom $f_n(n) = d(n) = f_n(n)+1$ — což je spor, *pokud* je funkce d v bodě n definovaná. Funkci d lze efektivně vyčíslit, ale v uvažované třídě přitom neleží.

Příčina je v obou případech stejná: hlavním omezením třídy obecných rekurzivních funkcí je to, že jsou nutně všude definované. Bez tohoto požadavku nám nic nebrání přijmout za efektivní i minimalizace jiných než speciálních funkcí, a vzniklá třída bude navíc stabilní v tom smyslu, že odolá diagonalizaci.

Neomezenou minimalizaci můžeme chápat jako předem neohraničené hledání: pokud nepožadujeme, aby $g(\overline{x}, y)$ byla speciální, potom $y \in \mathbb{N}$ splňující $g(\overline{x}, y) = 0$ nemusí existovat, takže funkce, která vznikne z g minimalizací, nebude všude definovaná. Proto se takové funkce nazývají obšírněji *částečné* rekurzivní funkce, narozdíl od obecných rekurzivních funkcí, které jsou totální.

Podrobněji: funkce f vznikne *minimalizací* funkce g, pokud pro každé $\overline{x} \in \mathbb{N}^k$ je $f(\overline{x})$ nejmenší $y \in \mathbb{N}$ takové (pokud existuje), že $g(\overline{x}, y) = 0$, zatímco pro všechna $z < y$ je hodnota $g(\overline{x}, z)$ definovaná a nenulová. To odpovídá naší představě o efektivní proceduře, která takové hodnoty postupným ověřováním

zjišťuje: pokud pro nějaké $z < y$ není $g(\overline{x}, z)$ definováno, pak test $g(\overline{x}, z) = 0$ nikdy neproběhne a na test $g(\overline{x}, y) = 0$ ani nedojde.[15]

5.2.36 Definice. Třída *rekurzivních* či obšírněji *částečných rekurzivních funkcí*, kterou budeme v dalším značit $\mathcal{RF}$, je nejmenší třída funkcí, která obsahuje funkci následníka $s(x) = x + 1$, všechny projekce $\pi_i^k(x_1, \ldots, x_k) = x_i$, za každé $1 \leq i \leq k \in \mathbb{N}$, a je uzavřena na skládání, primitivní rekurzi, a minimalizaci.

Každá obecná rekurzivní funkce je z definice částečná rekurzivní. Rozdíl oproti předchozím třídám funkcí je podstatný: částečné rekurzivní funkce nejsou nutně všude definované; definičním oborem k-ární funkce $f \in \mathcal{RF}$ je jen podmnožina $\mathbb{N}^k$. Třída $\mathcal{RF}$ je stále spočetná, ale pro diagonálu $d(n) = f_n(n) + 1$ seznamu $\{f_n; n \in \mathbb{N}\}$ není nutně $d \notin \mathcal{RF}$: je-li d funkce $f_n \in \mathcal{RF}$, pak o jejím indexu $n \in \mathbb{N}$ můžeme říci je tolik, že $n \notin dom(d)$. Později uvidíme, že diagonála třídy částečných rekurzivních funkcí je sama částečná rekurzivní, narozdíl od diagonalizace předchozích tříd.

S tím souvisí obvyklá úmluva: pokud pro $f, g \in \mathcal{RF}$ napíšeme $f(\overline{x}) = g(\overline{x})$, máme tím na mysli, že obě funkce f, g jsou v $\overline{x} \in \mathbb{N}^k$ definované a jejich hodnoty se rovnají. Podobně $f = g$ znamená, že definiční obory obou funkcí jsou totožné, a pro $\overline{x}$ z definičního oboru platí $f(\overline{x}) = g(\overline{x})$.

Níže takové konkrétní očíslování částečných rekurzivních funkcí ukážeme, ba dokonce popíšeme univerzální rekurzivní funkci, která při znalosti $n \in \mathbb{N}$ efektivně vyčísluje hodnoty funkce f_n.

Ke třídě částečných rekurzivních funkcí se ještě vrátíme. Nejdříve však ukážeme, že je velice přirozená: jedná se přesně o Turingovsky vyčíslitelné funkce.

5.3 Churchova teze

V tomto oddíle ukážeme, že Turingovy stroje a částečné rekurzivní funkce představují tutéž výpočetní sílu: rekurzivní funkce jsou právě vyčíslitelné funkce. To je jeden ze silných argumentů pro přijetí Churchovy teze: obě formalizace intuitivní představy efektivní vyčíslitelnosti, přestože jsou ve svých technických aspektech dosti odlišné, představují ve skutečnosti tutéž třídu funkcí.

5.3.1 Věta. *Každou rekurzivní funkci vyčísluje nějaký Turingův stroj.*

Důkaz. Stroje pro konstanty, projekce a následníka jsme sestrojili již v oddíle o Turingových strojích. Zbývá ukázat, jakým způsobem se postaví složitější stroje, které vyčíslují funkce vytvořené skládáním, primitivní rekurzí a minimalizací.

Buď $h(\overline{x}) = g(f_1(\overline{x}), \ldots, f_n(\overline{x}))$ funkce vytvořená složením, přičemž pro funkce $f_1, \ldots, f_n$ již máme stroje $M_1, \ldots, M_n$ a pro funkci g stroj M_g. Sestavíme třípáskový stroj M, který vyčísluje funkci h; víme již, že takový stroj lze redukovat na základní jednopáskový stroj. První páska je vstupní a obsahuje argumenty $\overline{x}$; druhá páska je pracovní, budeme na ní počítat hodnoty $f_i(\overline{x})$; třetí páska je výstupní, na ní budeme počítat vnější funkci g, jejíž hodnota je

[15] Buď například g funkce s hodnotami $g(0,0) = 1$, $g(0,1) = 0$, $g(1,1) = 0$, která není nikde jinde definovaná. Pro funkci f, která vznikne minimalizací g, je $f(0) = 1$, ale $f(1)$ není definováno: je sice $g(1,1) = 0$ a číslo 1 je první takové, ale není definováno $g(1,0)$. Jako krajní příklad pak poslouží prázdná funkce získaná minimalizací konstantní funkce $g(x,y) = 42$.

hodnotou složené funkce h. Výsledný stroj M funguje následovně: postupně pro každé $i \leq n$ zkopíruje argumenty $\overline{x}$ z první pásky na druhou a spustí nad druhou páskou stroj M_i. Pokud stroj M_i nikdy nezastaví, je $\overline{x} \notin dom(f_i)$, tedy také $\overline{x} \notin dom(h)$. Pokud M_i zastaví, je na druhé pásce napsána hodnota $f_i(\overline{x})$; stroj M tuto hodnotu zkopíruje na třetí pásku, vpravo od předchozích hodnot $f_j(\overline{x})$. Nakonec stroj M spustí stroj M_g nad třetí páskou, která nyní obsahuje argumenty $f_1(\overline{x}), \ldots, f_n(\overline{x})$. Pokud stroj M_g nezastaví, pak funkce g není definována v bodě $(f_1(\overline{x}), \ldots, f_n(\overline{x}))$, takže h není definována v bodě $\overline{x}$; pokud zastaví, pak na třetí pásce zanechá hodnotu $h(\overline{x}) = g(f_1(\overline{x}), \ldots, f_n(\overline{x}))$.

Buď h funkce vytvořená primitivní rekurzí z funkcí f a g, tj. $h(\overline{x}, 0) = f(\overline{x})$ a $h(\overline{x}, y+1) = g(\overline{x}, y, h(\overline{x}, y))$, přičemž funkci f resp. g vyčísluje Turingův stroj M_f resp. M_g. Stroj M vyčíslující funkci h je opět třípáskový: na první pásce čte argumenty $x_1, \ldots, x_k, x_{k+1}$, na druhé pásce iteruje postupně hodnoty všech $y < x_{k+1}$, a na třetí pásce spouští rekurzivní volání stroje M_g resp. M_f. Výsledný stroj M pracuje následovně. Nejprve na třetí pásku zkopíruje argumenty $x_1, \ldots, x_k$ a spustí nad nimi stroj M_f; pokud skončí, ocitne se tím na třetí pásce hodnota $f(\overline{x}) = h(\overline{x}, 0)$. Dále pak stroj M prochází následujícím cyklem: na druhou pásku nanese nejdříve hodnotu $y = 0$; dokud je hodnota y na druhé pásce menší než hodnota x_{k+1} na první pásce, zkopíruje stroj M před dosavadní hodnotu na třetí pásce hodnotu z druhé pásky, před ni ještě argumenty $x_1, \ldots, x_k$ z první pásky, a spustí M_g. Při první iteraci se tak počítá hodnota $g(\overline{x}, 0, h(\overline{x}, 0)) = h(\overline{x}, 1)$. Poté, co stroj M_g skončí (pokud skončí), zvětší stroj M hodnotu na druhé pásce o jedna a pokračuje stejně; na třetí pásce se tak dále počítají hodnoty $g(\overline{x}, 1, h(\overline{x}, 1)) = h(\overline{x}, 2)$ atd. Cyklus končí, když se na druhé pásce ocitne x_{k+1}; v tu chvíli je na třetí pásce napsáno $h(\overline{x}, x_{k+1})$. Opět platí, že pokud běh stroje M_f nebo některý z běhů stroje M_g neskončí, znamená to, že v příslušném bodě není definována funkce f resp. g, a tedy ani funkce h.

Buď konečně $h(\overline{x}) = \mu y(g(\overline{x}, y) = 0)$ funkce vzniklá minimalizací funkce g, přičemž funkci g vyčísluje stroj M_g. Stroj M vyčíslující funkci h bude mít dvě pásky: na první udržuje argumenty $\overline{x}, y$, na druhé spouští postupně výpočty $g(\overline{x}, 0), g(\overline{x}, 1)$, atd. Stroj M pracuje v následujícím cyklu: nejprve k argumentům $\overline{x}$ na první pásce připíše vpravo hodnotu 0; argumenty z první pásky překopíruje na druhou pásku a spustí nad nimi stroj M_g; pokud M_g doběhne, zkontroluje stroj M, zda je výsledná hodnota $g(\overline{x}, y)$ na druhé pásce nulová; pokud ano, znamená to, že současná hodnota posledního argumentu na první pásce je hledané $h(\overline{x}) = \mu y(g(\overline{x}, y) = 0)$; pokud ne, zvedne M poslední argument na první pásce o jedna a pokračuje stejně. Tak jako v předchozích případech, pokud některý z běhů stroje M_g neskončí, není ani funkce h v daném bodě definovaná; stejně tak pokud hledané y neexistuje, stroj M nikdy neskončí. □

V důkaze vynecháváme některé detaily, které považujeme za technikální: zkopírovat argument před nebo za posloupnost předchozích argumentů, hlídat nerovnost dvou argumentů či ověřovat nulovou funkční hodnotu je samo o sobě úloha, kterou má provést nějaký Turingův stroj, jehož instrukce jsou ovšem přímočaré. Vybízíme nicméně laskavého čtenáře, aby tyto detaily doplnil.[16]

[16] Programátor si zároveň všimne, že hodnotu $h(\overline{x}, y)$ funkce vytvořené primitivní rekurzí lze implementovat `for` cyklem, který postupně počítá nultou, první, druhou, až konečně hledanou y-tou hodnotu, a že v indukčním kroku pro primitivní rekurzi emulujeme právě takový cyklus. Podobně hodnotu funkce vytvořené minimalizací lze implementovat (potenciálně nekonečným)

V opačném směru máme nyní ukázat, že každá vyčíslitelná funkce je rekurzivní. K tomu potřebujeme zakódovat pojmy teorie Turingových strojů pomocí rekurzivních funkcí. Provedeme to pomocí kódování konečných posloupností, které jsme zavedli již v oddíle o elementárních funkcích. Jakmile budeme mít k dispozici rekurzivní funkce a relace zachycující pojmy jako *obsah pásky* či *krok výpočtu*, budeme schopni rekurzivní funkcí kódovat celý výpočet, včetně hodnoty, kterou při tom stroj případně vyčísluje.

Potřebujeme předně zavést formalismus, pomocí kterého budeme moci o Turingových strojích mluvit jako o matematických objektech, narozdíl od intuitivní představy fyzického počítacího stroje, o kterou jsme se dosud opírali.

5.3.2 Definice. *Turingův stroj* je funkce M, ke které existuje $N \in \mathbb{N}$ splňující $dom(M) \subseteq \{1,\dots,N\} \times \{0,1\}$ a $rng(M) \subseteq \{0,1\} \times \{0,1,2\} \times \{1,\dots,N\}$. *Obsah pásky* je zobrazení $T : \mathbb{Z} \to \{0,1\}$ s konečným nosičem. *Konfigurace* stroje M je čtveřice (M,T,s,j), kde M je Turingův stroj, T je obsah pásky, $s \leq N$ a $j \in \mathbb{Z}$. *Krok výpočtu* je dvojice konfigurací $(M,T,s,j),(M,T',s',j')$ taková, že při $M(s,Tj) = (w,m,s')$ je $T'j' = w$ a pro $m = 0,1,2$ je v odpovídajícím pořadí $j' = j-1, j, j+1$. *Výpočet* Turingova stroje M je konečná posloupnost konfigurací $(M,T_0,s_0,j_0),\dots,(M,T_n,s_n,j_n)$, ve které každá sousední dvojice tvoří výpočetní krok, a zároveň $(s_n,T_nj_n) \notin dom(M)$.

Zavedli jsme formálně pojmy související s během Turingových strojů. Nyní všechny tyto pojmy postupně zakódujeme pomocí rekurzivních funkcí. Potřebujeme předně kódovat samotné stroje: instrukci $M(s_i,r_i) = (w_i,m_i,t_i)$ budeme kódovat číslem $c_i = 2^{s_i+1}3^{r_i+1}5^{w_i+1}7^{m_i+1}11^{t_i+1}$, a kódem Turingova stroje M bude pro nás číslo $gM = \prod p_i^{c_i+1}$. Snadno se ukáže, že množina kódů Turingových strojů je elementární: číslo $x \in \mathbb{N}$ je takovým kódem právě když pro každé $i \leq lx$ je $l((x)_i) = 4$, platí $((x)_i)_1 \in \{0,1\}$, $((x)_i)_2 \in \{0,1\}$, $((x)_i)_3 \in \{0,1,2\}$, a pro žádné jiné $j \leq lx$ není $((x)_i)_0 = ((x)_j)_0$ a $((x)_i)_1 = ((x)_j)_1$.

Dále potřebujeme kódovat pozice na pásce a obsah pásky. Pozice na pásce jsou celá čísla, nabízí se kódovat kladná čísla sudými a záporná čísla lichými přirozenými čísly; položme tedy $g(j) = 2j$ pro $j \geq 0$ a $g(j) = -2j-1$ pro $j < 0$. Na těchto kódech[17] pak potřebujeme provádět operace, které zachycují pohyb na pásce. Definujeme tedy ještě funkce R a L očividným způsobem, totiž $R(x) = x+2$ pro x sudé, $R(x) = x-2$ pro x liché, a $R(1) = 0$. Potom pro každé $j \in \mathbb{Z}$ máme $R(g(j)) = g(j+1)$. Analogicky definujeme $L(x) = x+2$ pro x liché, $L(x) = x-2$ pro x sudé, a $L(0) = 1$, potom je $L(g(j)) = g(j-1)$. Přitom L,R jsou elementární funkce. Kódem pásky s obsahem T bude potom číslo $gT = \prod_{j=0}^{\infty} p_j^{b_j}$, kde $b_j = T(j/2)$ pro j sudé a $b_j = T(-(j+1)/2)$ pro j liché. Součin je ve skutečnosti konečný, neboť $\{j \in \mathbb{Z}; Tj \neq 0\} \subseteq \mathbb{Z}$ je konečná. Množina takových kódů je opět elementární: číslo x je kódem nějaké pásky, právě když $(\forall i < l(x))((x)_i < 2)$.

Dále potřebujeme kódovat základní operace na pásce, totiž přepis daného pole danou hodnotou. Buď W množina všech čtveřic (x,gj,b,y), kde $x = gT$ je nějaký kód obsahu pásky, $j \in \mathbb{Z}$, $b \in \{0,1\}$, a $y = g(T_j^b)$. To znamená, že y

`while` cyklem, a právě takový cyklus emulujeme v indukčním kroku pro minimalizaci.

[17] I kódy dalších objektů (pásek, výpočetních kroků, předávaných argumentů, …) budeme značit jako hodnoty téže funkce g. Tyto kódy se většinou obšírněji nazývají *Gödelova čísla*, neboť se používají při kódování syntaxe v Gödelových větách o neúplnosti aritmetiky (jak uvidíme později). Mohli bychom zavést zvláštní jména pro kódy strojů, pásek, výpočtů, atd.; na *jménech* kódů ale pochopitelně nezáleží.

kóduje pásku T^b_j, která se od pásky T liší jen hodnotou $Tj = b$. Relace W je opět elementární: $(x, gj, b, y) \in W$ platí právě když x je kód pásky a $y = [x/p_j^{(x)_j}] \cdot p_j^b$.

Kódem konfigurace (M, T, s, j) bude číslo $g(M, T, s, j) = 2^{gM} 3^{gT} 5^s 7^{gj}$. Množina takových kódů je opět elementární: číslo x je takovým kódem právě když $lx \geq 3$, $(x)_0$ je kódem nějakého stroje, $(x)_1$ je kódem nějaké pásky, a pro nějaké $i \leq l((x)_0)$ je $(((x)_0)_i)_0 = (x)_2$.

Výpočetní kroky zakódujeme přirozeným způsobem: buď S množina dvojic $(g(M, T, s, j), g(M, T', s', j'))$ takových, že $((M, T, s, j), (M, T', s', j'))$ tvoří výpočetní krok. Množina S kódů výpočetních kroků je pak opět elementární.

Kódem výpočtu $(M, T_0, s_0, j_0), \ldots, (M, T_n, s_n, j_n)$ Turingova stroje M bude konečně číslo $\prod_{i=0}^n p_i^{g(M,T_i,s_i,j_i)}$. Toto číslo je protokolem vůbec všeho, co se během výpočtu stalo: o který stroj kde, ve kterém kroku četl kde na pásce kterou hodnotu, kterou instrukci při tom použil, co se tím kde na pásce změnilo, atd. Množina C kódů všech výpočtů je opět elementární: $x \in C$ právě když $((x)_0)_2 = ((((x)_0)_0)_0)_0$, pro každé $i < l(x)$ je $((x)_i, (x)_{i+1}) \in S$, a pro žádné $i \leq l(((x)_0)_0)$ není $((((x)_0)_0)_i)_0 = ((x)_{lx})_2$ a zároveň $((((x)_0)_0)_i)_1 = (((x)_{lx})_1)_{((x)_{lx})_3}$; to jest, pro závěrečný stav a znak na pásce už stroj další instrukci nemá.

Pomocí elementárních funkcí jsme zachytili pojmy související obecně s během Turingových strojů. Nyní zakódujeme ještě pojmy související s vyčíslováním funkcí, tj. zadání argumentů na pásce a naopak čtení výsledné funkční hodnoty.

5.3.3 Definice. Pro konečnou posloupnost $b = (b_0, \ldots, b_n) \in 2^{n+1}$ buď $g(b) = \Pi_{i \leq n} p_i^{b_i+1}$. Speciálně pro $x \in \mathbb{N}$ buď $num(x) = g(1^{(x+1)}) = \Pi_{i \leq x} p_i^2$. Řekneme, že *na pásce T je mezi $q, r \in \mathbb{Z}$ napsáno číslo* $x \in \mathbb{N}$, pokud $q < r$, $r - q = x + 2$, $T(q) = 0 = T(r)$, a pro každé $q < j < r$ je $T(j) = 1$. Řekneme, že mezi q, r je napsána posloupnost čísel $x_1, \ldots, x_k \in \mathbb{N}$, pokud $r = q + \sum_{i \leq k}(x_i + 2)$ a pro každé $i \leq k$ je mezi $q + \sum_{m<i}(x_m + 2)$ a $q + \sum_{m \leq i}(x_m + 2)$ napsáno číslo x_i.

Přirozenou operaci řetězení binárních posloupností zachytíme i na jejich kódech: pokud pro čísla $x, y \in \mathbb{N}$ položíme $cat(x, y) = x \cdot \prod_{i \leq ly} p_{lx+i+1}^{(y)_i}$, bude $g(b^\frown c) = cat(g(b), g(c))$. Specielně posloupnosti jedniček jsme dříve zavedli jako způsob, jak předávat Turingovu stroji přirozená čísla jakožto argumenty; kódy takových posloupností dává elementární funkce num. Obecně pro k-ární funkce potřebujeme předávat k-tice čísel, totiž sekvence jedniček oddělených nulami. Zavedeme tedy ještě funkce $t_1(x) = cat(2, num(x))$ a $t_k(x_1, \ldots, x_k) = cat(t_{k-1}(x_1, \ldots, x_{k-1}), cat(2, x_k))$. Indukcí se ověří, že všechny t_k jsou elementární a $t_k(x_1, \ldots, x_k) = g(01^{(x_1+1)}0 \ldots 01^{(x_k+1)})$.

Buď konečně A množina čtveřic (gT, gb, gq, gr) takových, že T je obsah pásky a b je nějaká konečná binární posloupnost, která leží na pásce T od q do r. Ukážeme, že množina A takových kódů je elementární. Kódy pásek, posloupností a pozic jsou samy o sobě elementární. Čtveřice (x, y, m, n) padne do A právě tehdy, když $x \neq 0$ je kódem pásky, $y \neq 0$ je kódem binární posloupnosti, a platí jedno z následujících. Buďto je $y = 1$ a $m = n$, totiž y kóduje prázdnou posloupnost; nebo je $y > 1$, pro každé $i \leq l(y)$ je $(y)_i = 0 + 1$ nebo $(y)_i = 1 + 1$, a existuje $z \leq (m + 2y)^y$ s následující vlastností: $l(z) = l(y)$, $(z)_0 = m$, $(z)_{lz} = n$, pro každé $i < l(z)$ je $R((z)_i) = (z)_{i+1}$, a pro každé $i \leq l(z)$ je $(x)_{(z)_i} = (y)_i \dot{-} 1$. Takové z kóduje souvislý úsek pásky, který délkou odpovídá posloupnosti s kódem y, a na příslušných pozicích stojí příslušné hodnoty.

Pomocí elementárních funkcí a relací jsme zachytili pojmy teorie Turingových strojů potřebné k důkazu věty. Zavedeme nyní důležitou relaci, která kóduje fakt, že na daném vstupu provedl stroj nějaký výpočet.

5.3.4 Definice. Pro $k > 0$ buď T^k relace obsahující taková $(e, x_1, \ldots, x_k, c)$, pro která platí: číslo $e = g(M)$ je kód Turingova stroje M; c je protokol výpočtu $(M, T_0, s_0, j_0), \ldots, (M, T_n, s_n, j_n)$; T_0 je prázdná až na posloupnost $01^{(x_1+1)}0 \ldots 01^{(x_k+1)}$, která začíná na j_0, a $T_n(j_n) = 1$. ∎

Relace T^k zachycuje situaci, kterou jsme popsali při zavádění pojmu Turingovsky vyčíslitelné funkce: vpravo od čtecí hlavy stojí na začátku výpočtu k-tice argumentů; po skončení výpočtu stojí hlava na první jedničce funkční hodnoty. Vzhledem k tomu, že naše stroje jsou deterministické, může pro dané $e, x_1, \ldots, x_k$ existovat nanejvýš jedno c takové, že $(e, x_1, \ldots, x_k, c) \in T^k$. Pokud takové c existuje, znamená to především, že výpočet vůbec doběhl.

O relaci T^k opět snadno ukážeme, že je elementární: kódy strojů a výpočtů jsou elementární; přitom $(e, x_1, \ldots, x_k, c) \in T^k$ právě když platí následující: $((c)_0)_0 = e$, $((c)_0)_3 = 0$, $(((c)_{lc})_1)_2 = 1$, a zároveň pro nějaké $i \leq ((c)_0)_1$ je $(((c)_0)_1, t_k(x_1, \ldots, x_k), i, 1) \in A$, totiž na původní pásce je napsána k-tice zadaných argumentů, a pro lichá $i < j \leq ((c)_0)_1$ i všechna sudá $j \leq ((c)_0)_1$ je $(((c)_0)_1)_j = 0$, totiž jinak je počáteční páska prázdná.

Z protokolu proběhlého výpočtu pak dokážeme vyčíst, jakou hodnotu po sobě stroj na pásce zanechal, a to opět prostřednictvím elementární funkce: pro $x \in \mathbb{N}$ buď $val(x) = \mu[y \leq x]((((((x)_{lx})_1, cat(2, num(y)), ((x)_{lx})_3, ((x)_{lx})_3 + y)) \in A)$. Jedná se o omezenou minimalizaci, takže val je skutečně elementární. Pro $x \in C$ je $val(x)$ číslo, které je v poslední konfiguraci napsané na pásce vpravo od hlavy.

5.3.5 Věta. *Každá Turingovsky vyčíslitelná funkce je rekurzivní.*

Důkaz. Buď M stroj, který počítá funkci $f : \mathbb{N}^k \to \mathbb{N}$, buď $e = g(M)$ jeho kód. Pro $x_1, \ldots, x_k \in \mathbb{N}$ je potom $f(x_1, \ldots, x_k) = val(\mu c(T^k(e, x_1, \ldots, x_k, c))$. □

5.4 Univerzální funkce

Relací T^k popsanou výše je určena částečná rekurzivní funkce

$$\Phi^k(e, \overline{x}) = val(\mu c((e, \overline{x}, c) \in T^k));$$

tato funkce je v následujícím smyslu *univerzální*: je-li f libovolná k-ární částečná rekurzivní funkce, je $f(\overline{x}) = \Phi(e, \overline{x})$, kde $e \in \mathbb{N}$ je kód nějakého Turingova stroje, který funkci f vyčísluje. Říkáme, že $e \in \mathbb{N}$ je *index* funkce f.

Ekvivalentně můžeme Turingův stroj, který funkci Φ vyčísluje, považovat za *univerzální stroj*: jako svůj vstup přijímá popis (kód) nějakého jiného stroje, jehož běh bude emulovat, a jeho vstupní data.[18] V Turingově článku je takový univerzální stroj výslovně popsán, my budeme pracovat s univerzální funkcí.

Položíme-li nyní $[[e]]^k(\overline{x}) = \Phi^k(e, \overline{x})$, bude posloupnost

$$[[0]]^k, [[1]]^k, [[2]]^k, [[3]]^k, \ldots$$

[18] Je zde přímá analogie s moderními operačními systémy: instrukce programu lze kódovat čísly stejně jako jakákoli jiná data; úkolem operačního systému je pak tyto programy spouštět.

obsahovat vůbec všechny k-ární rekurzivní funkce, neboť každá taková funkce má nějaký index. Ve skutečnosti má nekonečně mnoho různých indexů a v uvedené posloupnosti se tedy nekonečněkrát opakuje — každý stroj, který danou funkci počítá, můžeme uměle učinit složitějším a tím jeho kód zvýšit. Posloupnost je dosti redundantní, neboť jen některá čísla jsou kódy Turingových strojů, které vyčíslují nějakou funkci. I v takovém případě je však $[[e]]$ nějaká částečná rekurzivní funkce (v krajním případě prázdná).

5.4.1 Věta (Kleene). *$\Phi^k : \mathbb{N}^{k+1} \to \mathbb{N}$ je částečná rekurzivní funkce. Pro každé číslo $e \in \mathbb{N}$ je $[[e]](\overline{x}) = \Phi^k(e, \overline{x})$ částečná rekurzivní funkce k proměnných, a naopak každá taková funkce $f : \mathbb{N}^k \to \mathbb{N}$ je právě funkcí $[[e]]^k$ pro nějaké $e \in \mathbb{N}$.*

Speciálně pro $k = 1$ máme enumeraci $[[0]], [[1]], [[2]], \ldots$ všech unárních částečných rekurzivních funkcí, a nabízí se možnost tuto posloupnost diagonalizovat stejně jako v 5.2.34. Položme tedy $d(x) = [[x]](x)+1$. Potom d je unární částečná rekurzivní funkce, má tedy nějaký index $e \in \mathbb{N}$. Jinými slovy, d je právě funkce $[[e]]$; ptejme se na hodnotu $d(e)$. Z definice je $d(e) = [[e]](e)+1 = d(e)+1$. Spor je jen zdánlivý, jedná se o částečnou funkci: vidíme jen tolik, že diagonální funkce d není definována na svém vlastním indexu, jinak máme okamžitě spor.[19]

Pro částečnou rekurzivní funkci je přirozené se ptát, zda má nějaké totální rekurzivní rozšíření. Naivně můžeme částečnou funkci rozšířit prostě tak, že ji všude mimo dosavadní definiční obor dodefinujeme třeba nulou. Nemáme ale nijak nezaručeno, že takové rozšíření bude rekurzivní. Diagonální funkce ve skutečnosti žádné rekurzivní zúplnění nemá.

5.4.2 Věta. *Neexistuje žádné totální rekurzivní rozšíření diagonální funkce.*

Důkaz. Buď $f : \mathbb{N} \to \mathbb{N}$ totální rozšíření diagonální funkce $d(x) = [[x]](x) + 1$. Potom f není rekurzivní: pokud ano, buď $e \in \mathbb{N}$ její index. Pokud $[[e]](e)$ není definováno, není f totální. Je-li $[[e]](e)$ definováno, je definováno i $[[e]](e)+1 = d(e) = f(e)$, takže f není funkce $[[e]]$. □

Srovnáme-li tuto situaci s větou o definici po částech, dojdeme k závěru, že definiční obor diagonální funkce nemůže být rekurzivní množina: potom by i její doplněk byl rekurzivní, a totální rekurzivní rozšíření bychom triviálně dodefinovali. Rozklad $\mathbb{N}$ na $dom(d)$ a doplněk tedy není rozklad na rekurzivní části.

5.5 Halting Problem

Viděli jsme, že diagonální funkce nemůže být definována na svém vlastním kódu. Některé jiné rekurzivní funkce na svém vlastním kódu definovány jsou (například všechny totální), označme tedy $K = \{x; x \in dom[[x]]\} \subseteq \mathbb{N}$. Obecněji se můžeme ptát, zda lze efektivně rozhodovat o tom, která částečná rekurzivní funkce je kde definována, nebo ekvivalentně jestli daný stroj vůbec zastaví na daném vstupu. To je rozhodovací problém známý jako *halting problem*. Formálně mluvíme o kódech strojů resp. indexech funkcí a uvažujeme relaci $H = \{(x, y); y \in dom[[x]]\}$.

[19] Stejně by posloužila funkce $d(x) = 1 \dot{-} [[x]](x)$, která navíc nabývá jen hodnot 0 nebo 1.

5.5.1 Věta. *K není rekurzivní množina. H není rekurzivní relace.*

Důkaz. Je-li K rekurzivní, buď $f(x) = d(x)$ pro $x \in K$ a $f(x) = 0$ pro $x \notin K$. To je totální rekurzivní rozšíření diagonální funkce, spor. Je-li H rekurzivní, je i K s charakteristickou funkcí $\chi_K(x) = \chi_H(x, x)$ rekurzivní, spor. □

Halting problem je příkladem algoritmicky nerozhodnutelného problému. Od objevu fenoménu nerozhodnutelnosti se podařilo o rozličných rozhodovacích problémech v matematice a informatice ukázat, že jsou algoritmicky nerozhodnutelné. Jedním z nejznámějších je problém diofantických rovnic, známý též jako Hilbertův desátý problém. Otázka zní, zda má daný polynom v konečně mnoha proměnných s celočíselnými koeficienty (např. $3x^2y + 5xy + 2xy^2 + 11$) nějaké celočíselné kořeny. Ukázalo se překvapivě (Matijasevič, 1970), že neexistuje algoritmus, který by tuto otázku rozhodoval. Níže popíšeme jiný zásadní případ, totiž nerozhodnutelnost dokazatelnosti v aritmetice.

5.5.2 Cvičení. Definujte elementární relaci $H(x, y, t)$, která platí právě když stroj s kódem x skončí svůj výpočet na pásce s kódem y po $\leq t$ krocích. Takovou relaci lze sestavit podobně jako při kódování výpočtu, a vzhledem k omezenému počtu kroků vystačíme s omezenou kvantifikací. Tím bude ukázáno, že zatímco obecná otázka po zastavení stroje je algoritmicky nerozhodnutelná, otázka po zastavení v určitém předem daném čase je dokonce elementární.[20] Obecně můžeme definovat funkci $time(e, \overline{x}) = l(\mu c((e, \overline{x}, c) \in T^k))$, která udává počet kroků, tedy čas strávený výpočtem. Tato funkce však není totální, a z předchozího tvrzení plyne, že není shora omezená žádnou rekurzivní funkcí, tj. neexistuje žádná totální rekurzivní funkce $f(e, x)$ taková, aby platilo $time(e, \overline{x}) \leq f(e, \overline{x})$ kdykoli je levá strana definovaná: kdyby ano, byl by halting problem rozhodnutelný.

5.6 Rekurzivní spočetnost

V tomto oddíle rozvolníme pojem rekurzivní množiny, potažmo relace. Motivací je nám následující vlastnost relací K a H z předchozího. Víme, že nejsou rekurzivní, tj. jejich charekteristické funkce nejsou rekurzivní. Problém je v tom, že po charakteristické funkci χ_A množiny $A \subseteq \mathbb{N}$ požadujeme totalitu: o *každém* $x \in \mathbb{N}$ chceme rozhodnout, zda je či není prvkem A. Pokud se spokojíme s kladnou odpovědí v případech $x \in A$, dostaneme následující pojem.

5.6.1 Definice. Buď $A \subseteq \mathbb{N}^k$. Potom její *semicharakteristická funkce* je částečná funkce $c_A : \mathbb{N}^k \to \mathbb{N}$, která v bodech $\overline{x} \in A$ má hodnotu $c_A(\overline{x}) = 1$ a nikde jinde není definována. Řekneme, že množina $A \subseteq \mathbb{N}^k$ je *rekurzivně spočetná*, pokud c_A je částečná rekurzivní funkce.

Relace K, H z předchozího nejsou rekurzivní, jsou však rekurzivně spočetné, totiž jejich semicharakteristické funkce c_K, c_H jsou (částečné) rekurzivní. Výpočetní procedura pro c_K je nasnadě: pro dané $x \in \mathbb{N}$ spočti $[[x]](x)$; až výpočet doběhne (pokud doběhne), vrať 1. Podobně pro $c_H(x, y)$: spusť stroj s kódem x na vstupu y; až výpočet skončí (pokud skončí), vrať 1. Formálně máme částečné rekurzivní funkce $c_K(x) = 1 + 0 \cdot [[x]](x)$ a $c_H(x, y) = 1 + 0 \cdot [[x]](y)$.

[20] Intuitivně je rozhodovací procedura zřejmá: zkontroluj po t krocích, zda stroj ještě běží.

Můžeme obrazně říci, že pro rekurzivně spočetné relace sice nemáme efektivní *rozhodovací* proceduru, ale máme alespoň efektivní *ověřovací* proceduru: o prvku $x \in A$ dokážeme v konečném čase ověřit, že je skutečně prvkem. Vzhledem k ekvivalenci rekurzivních funkcí a Turingových strojů můžeme za rekurzivně spočetné množiny považovat takové, na kterých se Turingův stroj zastaví; říkáme pak, že takový stroj *přijímá* prvky rekurzivně spočetné množiny.

Všimněme si, že množina $K = \{x; x \in dom[[x]]\}$ je právě definiční obor diagonální funkce $d(x)$ a relace $H = \{(x,y); y \in dom[[x]]\}$ je právě definiční obor univerzální funkce $\Phi(x,y)$. Tuto vlastnost jsme vlastně přijali za definici; rekurzivně spočetná relace je právě definiční obor své semicharakteristické funkce.

5.6.2 Věta. *Pro relaci $A \subseteq \mathbb{N}^k$ jsou následující podmínky ekvivalentní.*

(i) *A je rekurzivně spočetná.*

(ii) *A je definičním oborem nějaké částečné rekurzivní funkce.*

(iii) *$A = \{\overline{x}; (\exists y) R(\overline{x}, y)\}$ pro nějakou elementární relaci $R \subseteq \mathbb{N}^{k+1}$.*

(iv) *$A = \{\overline{x}; (\exists y_1) \dots (\exists y_n) R(\overline{x}, \overline{y})\}$ pro nějakou elementární relaci $R \subseteq \mathbb{N}^{k+n}$.*

Důkaz. (i $\to$ ii) $A = dom(c_A)$. (ii $\to$ i) $c_{dom(f)}(\overline{x}) = 0 \cdot f(\overline{x}) + 1$. (ii $\to$ iii) Je-li $A = dom([[e]]^k)$, je $\overline{x} \in A$ právě když $(\exists c) T^k(e, \overline{x}, c)$, přitom T^k je elementární relace. (iii $\to$ ii) Je-li R elementární relace taková, že $A = \{\overline{x}; (\exists y) R(\overline{x}, y)\}$, je $f(x) = \mu y R(\overline{x}, y)$ částečná rekurzivní funkce a $dom(f) = A$. (iv $\to$ iii) S pomocí kódování máme $(\exists y_1) \dots (\exists y_n) R(\overline{x}, \overline{y})$ právě když $(\exists y) R(\overline{x}, (y)_1, \dots, (y)_n)$. □

Jako důsledek máme enumeraci rekurzivně spočetných relací: položíme-li $R_e^k = dom([[e]]^k)$, obsahuje posloupnost $R_0^k, R_1^k, R_2^k, \dots$ všechny rekurzivně spočetné relace $R \subseteq \mathbb{N}^k$, každou v nekonečně mnoha opakováních. V případě $k = 1$ budeme horní index často vynechávat a psát stručně $R_e \subseteq \mathbb{N}$ místo $R_e^1 \subseteq \mathbb{N}^1$.

5.6.3 Věta. *Pro neprázdnou $A \subseteq \mathbb{N}$ jsou následující podmínky ekvivalentní.*

(i) *A je rekurzivně spočetná.*

(ii) *A je oborem hodnot nějaké unární elementární funkce.*

(iii) *A je oborem hodnot nějaké částečné rekurzivní funkce.*

Důkaz. (i $\to$ ii) Podle předchozího je $A \subseteq \mathbb{N}$ tvaru $\{x; (\exists y) R(x,y)\}$, kde $R \subseteq \mathbb{N}^2$ je nějaká elementární relace. V tom případě je A oborem hodnot elementární funkce g s hodnotami $g(x,y) = x$ pro $R(x,y)$ a $g(x,y) = a$ jinak, kde $a \in A \neq \emptyset$ je nějaký předem zvolený prvek. Zbývá najít unární funkci se stejnou vlastností, k tomu ale stačí kódovat dvojice: položme $f(x) = (x)_0$ pro $R((x)_0, (x)_1)$ a $f(x) = a$ jinak. (iii $\to$ i) Je-li $A = rng(f)$, položme $g(x) = \mu y(f(y) = x) = \mu y(|f(y) - x| = 0)$. Potom g je částečná rekurzivní funkce a $dom(g) = A$. □

Pro rekurzivně spočetnou množinu $A \subseteq \mathbb{N}$ máme totální unární rekurzivní funkci f, pro kterou je $A = \{f(0), f(1), f(2), \dots\}$. Ekvivalentně můžeme říci, že máme Turingův stroj, který na každém vstupu zastaví, a postupně pro $0, 1, 2, \dots$

vypisuje právě a jen všechny prvky množiny A. Řekli jsme dříve, že pro rekurzivně spočetnou množinu $A \subseteq \mathbb{N}$ sice nemáme efektivní rozhodovací proceduru, máme však alespoň efektivní ověřovací proceduru. Ekvivalentně můžeme nyní říci, že máme k dispozici efektivní vypisující proceduru.

Nabízí se přirozená otázka, zda funkce zaručená předchozí větou může být navíc rostoucí, tedy zda lze prvky rekurzivně spočetné množiny $A \subseteq \mathbb{N}$ efektivně vypisovat podle velikosti. Podle následující věty to obecně možné není.

5.6.4 Věta. *Pro totální rekurzivní rostoucí funkci f je $rng(f) \subseteq \mathbb{N}$ rekurzivní.*

Myšlenka důkazu je jednoduchá: pokud prvky množiny $A = rng(f)$ vypisuje rostoucí funkce, potom pro dané $y \in \mathbb{N}$ nastane po konečně mnoha krocích buďto $y = f(x)$, a máme $y \in A$, nebo $y < f(x)$, a máme $y \notin A$.

Důkaz. Je-li f rostoucí, indukcí se snadno ukáže, že je $f(x) \geq x$ pro každé $x \in \mathbb{N}$. Charakteristickou funkcí množiny $rng(f)$ je potom $\sum_{x \leq y} \chi_{=}(f(x), y)$. □

5.6.5 Věta. *$R \subseteq \mathbb{N}^k$ je rekurzivní právě když R i $\mathbb{N}^k \setminus R$ jsou rekurzivně spočetné.*

Jsou-li relace R a $\overline{R} = \mathbb{N}^k \setminus R$ rekurzivně spočetné, mají rekurzivní semicharakteristické funkce. Rozhodovací procedura se potom nabízí: počítej zároveň $c_R(\overline{x})$ a $c_{\overline{R}}(\overline{x})$, a odpověz podle toho, která funkce dříve vrátí jedničku — po konečně mnoha krocích nastane právě jeden z obou případů.

Důkaz. Jeden směr je triviální: je-li R rekurzivní, je také $\overline{R} = N^k \setminus R$ rekurzivní; speciálně jsou obě rekurzivně spočetné. Jsou-li naopak R i $\overline{R}$ rekurzivně spočetné, buďte $P, Q \subseteq \mathbb{N}^{k+1}$ elementární relace takové, že $R = \{\overline{x}; (\exists y) P(\overline{x}, y)\}$ a $\overline{R} = \{\overline{x}; (\exists y) Q(\overline{x}, y)\}$. Potom $f(\overline{x}) = \mu y (P(\overline{x}, y) \vee Q(\overline{x}, y))$ je totální rekurzivní funkce. Můžeme obrazně říci, že $f(\overline{x})$ je svědkem buďto pro $P(\overline{x}, f(\overline{x}))$ nebo pro $Q(\overline{x}, f(\overline{x}))$, podle toho, která z možností $R(\overline{x})$ nebo $\overline{R}(\overline{x})$ nastane. Přitom $R(\overline{x})$ právě když $P(\overline{x}, f(\overline{x}))$, takže R je rekurzivní. □

Jako důsledek získáváme, že pro relaci $R \subseteq \mathbb{N}^k$, která není rekurzivní, není alespoň jedna z množin $R, \mathbb{N}^k \setminus R \subseteq \mathbb{N}^k$ ani rekurzivně spočetná. Speciálně relace $\overline{K}$ a $\overline{H}$ nejsou ani rekurzivně spočetné, neboť K i H jsou rekurzivně spočetné, ale nikoli rekurzivní. Pro $\overline{K}$ dokonce platí, že je-li $R_e \subseteq \overline{K}$ rekurzivně spočetná, je $e \in \overline{K} \setminus R_e$, tj. sám index e je svědkem nerovnosti $R_e \neq \overline{K}$. Je-li totiž $R_e \subseteq \overline{K}$, nemůže být $e \in K$: to by znamenalo $e \in dom([[e]]) = R_e \subseteq \overline{K}$. Je tedy $e \in \overline{K}$, takže $e \notin dom([[e]]) = R_e$.

Systém rekurzivně spočetných $R \subseteq \mathbb{N}^k$ sice podle věty výše není uzavřen na doplňky, je ale uzavřen na průniky a sjednocení. Pro částečné rekurzivní funkce f, g a rekurzivně spočetné $A = dom(f), B = dom(g)$ máme $A \cap B = dom(f+g)$; pro rekurzivně spočetné $C = rng(f)$ a $D = rng(g)$ je $C \cup D = rng(h)$, kde $h(x) = f([x/2]) \cdot \chi_S(x) + g([x/2]) \cdot \chi_{\mathbb{N} \setminus S}(x)$ a $S \subseteq \mathbb{N}$ je množina sudých čísel.

5.6.6 Věta. *Buď $R \subseteq \mathbb{N}^m$ rekurzivně spočetná, buďte $f_1, \ldots, f_m$ částečné rekurzivní k-ární funkce. Potom $\{\overline{x}; (f_1(\overline{x}), \ldots, f_m(\overline{x})) \in R\}$ je rekurzivně spočetná.*

Připomeňme opět konvenci zápisu pro částečné funkce: zápisem $f(\overline{x}) \in A$ máme na mysli, že funkce f je v $\overline{x}$ definovaná a funkční hodnota leží v A. Ve větě bychom mohli pečlivěji psát $\{\overline{x} \in \bigcap dom(f_i); (f_1(\overline{x}), \ldots, f_m(\overline{x})) \in R\}$, nicméně to přesně máme na mysli zápisem $\{\overline{x}; (f_1(\overline{x}), \ldots, f_m(\overline{x})) \in R\}$,

Důkaz. Semicharakteristická funkce c_R rekurzivně spočetné relace R je částečná rekurzivní. Semicharakteristická funkce $\overline{x} \mapsto c_R(f_1(\overline{x}), \ldots, f_m(\overline{x}))$ relace $\{\overline{x}; (f_1(\overline{x}), \ldots, f_m(\overline{x})) \in R\}$ je tedy složená z částečných rekurzivních funkcí, takže je sama částečná rekurzivní. □

Pojem rekurzivnosti resp. rekurzivní spočetnosti jsme zavedli zvlášť pro funkce a relace. Přitom funkce $f : \mathbb{N}^k \to \mathbb{N}$ je speciálním typem relace $R \subseteq \mathbb{N}^{k+1}$, je tedy přirozené se ptát, jak spolu oba pojmy souvisí. Proto v následujícím tvrzení mluvíme zvlášť o funkci a jejím grafu, přestože jde o tutéž množinu.

5.6.7 Věta. *Funkce $f : \mathbb{N}^k \to \mathbb{N}$ je částečná rekurzivní právě tehdy, když její graf $\{(\overline{x}, f(\overline{x})); \overline{x} \in dom(f)\} \subseteq \mathbb{N}^{k+1}$ je rekurzivně spočetná relace. Funkce f je totální rekurzivní právě tehdy, když její graf je rekurzivní relace.*

Důkaz. Je-li f částečná rekurzivní, pak i funkce $g(\overline{x}, y) = (f(\overline{x}) \dot{-} y) + (y \dot{-} f(\overline{x}))$ je částečná rekurzivní, a jejím definičním oborem je právě graf funkce f. Je-li naopak graf $G \subseteq \mathbb{N}^{k+1}$ funkce $f : \mathbb{N}^k \to \mathbb{N}$ rekurzivně spočetnou množinou, je i množina $A \subseteq \mathbb{N}$ kódů $\langle f(\overline{x}), \overline{x} \rangle$ rekurzivně spočetná, a tedy je oborem hodnot nějaké elementární funkce g. Hodnotu $f(\overline{x})$ pak získáme minimalizací jako $\mu y((\exists z)(g(z) = \langle y, \overline{x} \rangle)$ a f je částečná rekurzivní funkce. Je-li f totální rekurzivní, je i její graf rekurzivní relace s charakteristickou funkcí $\chi_G(\overline{x}, y) = \chi_=(f(\overline{x}), y)$. Opačný směr se ukáže jako výše, funkce f je ale navíc totální. □

5.6.8 Příklad. Každá nekonečná rekurzivně spočetná množina obsahuje nekonečnou rekurzivní množinu. Buď totiž $A = rng(f)$ pro nějakou unární elementární funkci f. Potom funkce $k(z) = \mu y(f(y) > z)$ je z definice částečná rekurzivní, ve skutečnosti je ale totální, neboť $A = rng(f)$ je nekonečná. Primitivní rekurzí pak definujme $h(0) = f(0)$ a $h(x+1) = f(k(h(x)))$. Potom h je totální rekurzivní funkce a snadno se nahlédne, že je ostře rostoucí; tedy $B = rng(h) \subseteq rng(f) = A$ je nekonečná rekurzivní množina.

5.6.9 Příklad. Pro částečnou rekurzivní funkci je přirozené se ptát, zda je totální. Položme tedy $\mathcal{F} = \{e \in \mathbb{N}; [[e]] \text{ je totální}\}$; otázka pak zní, zda je $e \in \mathcal{F}$. Ukážeme, že $\mathcal{F}$ není ani rekurzivně spočetná: je-li $E \subseteq \mathcal{F}$ rekurzivně spočetná, pak existuje totální unární rekurzivní funkce, jejíž index není v E. Můžeme předpokládat, že E je neprázdná, takže podle předchozího je $E = rng(f)$ pro nějakou totální unární rekurzivní funkci f. Položme $g(x) = [[f(x)]](x) + 1$. To je totální unární rekurzivní funkce, neboť každé $f(x) \in E \subseteq \mathcal{F}$, takže $[[f(x)]]$ je totální. Tvrdíme, že index $e \in \mathcal{F}$ funkce g není v E; pro totální funkce můžeme v podstatě zopakovat argument z 5.2.34. Kdyby totiž $e \in E$, je $e = f(x) \in E$ pro nějaké $x \in \mathbb{N}$, neboť f enumeruje E. Pak ale $[[e]](x) = [[f(x)]](x)$, zatímco $g(x) = [[f(x)]](x) + 1$, takže $[[e]]$ není g.

5.6.10 Cvičení. Ukažte, že množina $A = \{x; [[x]](x) = 0\}$ je rekurzivně spočetná, avšak nikoli rekurzivní, a stejně tak množina $B = \{x; [[x]](x) = 1\}$. Máme tedy dvě disjunktní rekurzivně spočetné množiny; ukažte, že nejsou *rekurzivně oddělitelné*, tj. neexistuje rekurzivní množina C taková, aby $A \subseteq C$ a $B \subseteq \overline{C}$.

5.7 Relativní složitost

O náležení do rekurzivní množiny dokážeme efektivně rozhodnout, zatímco náležení do rekurzivně spočetné množiny dokážeme jen efektivně ověřit. Rekurzivně spočetné množiny jsou v tomto smyslu složitější než rekurzivní. Obecně lze pak složitost množin porovnávat pomocí rekurzivních funkcí:

5.7.1 Definice. Řekneme, že totální rekurzivní funkce $f : \mathbb{N} \to \mathbb{N}$ *redukuje* množinu $A \subseteq \mathbb{N}$ na množinu $B \subseteq \mathbb{N}$, pokud pro každé $x \in \mathbb{N}$ je $x \in A$ právě když $f(x) \in B$; v tom případě píšeme $A \leq_f B$. Pokud nějaká taková funkce existuje, řekneme, že A je *redukovatelná* na B a píšeme $A \leq B$.

Jde o to, že otázku po náležení do A dokážeme efektivně převést na otázku po náležení do B. Hned z definice máme, že při $A \leq_f B$ je zároveň $\overline{A} \leq_f \overline{B}$. Relace $\leq$ porovnává množiny podle složitosti, totiž podle složitosti otázky po náležení. V případě dvou tříd složitosti, které dosud známe, totiž rekurzivních a rekurzivně spočetných množin, se jedná o následující tvrzení.

5.7.2 Věta. *Buď $A \leq B$. Potom je-li B rekurzivní, je také A rekurzivní, a je-li B rekurzivně spočetná, je také A rekurzivně spočetná.*

Důkaz. V prvním případě stačí použít větu o dosazování rekurzivních funkcí do rekurzivních relací: je-li B rekurzivní, je také $A = \{x; f(x) \in B\}$ rekurzivní. Analogickou větu jsme výše dokázali i pro rekurzivně spočetné množiny. □

Jako důsledek získáváme možnost ukázat o dané množině, že je resp. není rekurzivní (či rekurzivně spočetná) porovnáním s jinou takovou množinou. Pokud například ukážeme, že $\overline{K} \leq B$, pak B není rekurzivně spočetná.

5.7.3 Cvičení. Relace $A \leq B$ není uspořádáním: při $A \leq B$ a $B \leq A$ není nutně $A = B$ (najděte dvě takové množiny). Stačí však systém $P(\mathbb{N})$ vhodně faktorizovat: položíme-li $A \equiv B$ právě když $A \leq B \wedge B \leq A$, je $\equiv$ ekvivalence na $P(\mathbb{N})$, a definujeme-li na ekvivalenčních třídách $[A] = \{A'; A \equiv A'\}$ relaci $[A] \leq [B]$ podmínkou $A \leq B$, je $\leq$ uspořádáním na kvocientu $P(\mathbb{N})/\equiv$.

5.8 Počítání indexů

Z předchozího víme, že každá rekurzivní funkce má svůj index. To znamená, že při počítání s čísly počítáme s indexy funkcí, potažmo s kódy strojů; funkce, která číslu přiřazuje číslo, tak může stroji přiřazovat stroj. Ukazuje se, že mnohé vztahy mezi funkcemi (stroji, programy) jsou samy rekurzivní. V tomto oddíle se budeme zabývat právě takovým *počítáním programů* — půjde o algoritmy, které prostřednictvím kódů počítají s algoritmy jako s objekty.

Víme například, že každá konstantní funkce je (totální) rekurzivní, má tedy svůj index. Ukážeme, že ve skutečnosti existuje totální rekurzivní funkce f, která danému $k \in \mathbb{N}$ přiřazuje právě kód $f(k)$ funkce s konstantní hodnotou $[[f(k)]](x) = k$. Podobně složení rekurzivních funkcí je rekurzivní funkce, dá se však ukázat více: existuje binární rekurzivní funkce, která argumentům x, y, jakožto kódům rekurzivních funkcí $[[x]], [[y]]$ přiřadí kód jejich složení.

Rekurzivně spočetná množina $dom[[x]]$ je zároveň oborem hodnot $rng[[y]]$ nějaké jiné rekurzivní funkce. Dá se opět ukázat, že existuje dokonce rekurzivní

funkce f taková, že pro každé $x \in \mathbb{N}$ je $dom[[x]] = rng[[f(x)]]$. Konečně pro binární rekurzivní funkci $g(x,y)$ a každou pevně zvolenou hodnotu $y \in \mathbb{N}$ je funkce $f(x) = g(x,y)$ opět rekurzivní. Tato korespondence je opět efektivní: existuje binární rekurzivní funkce $\varrho(e,y)$, tedy uniformní postup, který kódu e binární funkce $[[e]](x,y) = g(x,y)$ a zvolenému $y \in \mathbb{N}$ přiřadí právě kód $\varrho(e,y)$ unární funkce $[[\varrho(e,y)]](x) = f(x) = g(x,y) = [[e]](x,y)$.

5.8.1 Věta. *Existuje elementární binární funkce ϱ taková, že pro každé e, x, y*

$$[[e]](x,y) = [[\varrho(e,y)]](x).$$

Důkaz. Definovat takovou funkci obnáší dekódovat dlouhou indukcí kód $e \in \mathbb{N}$ a postavit s ohledem na dané $y \in \mathbb{N}$ jiný stroj, respektive jeho kód. Popíšeme takový stroj jen slovy — vypouštíme podrobný argument opírající se o (de)kódování strojů, které jsme prováděli při důkazu věty o ekvivalenci Turingových strojů a rekurzivních funkcí. Hodnotou $\varrho(e,y)$ je kód stroje, který se na daném vstupu x chová následovně. Vlevo od vstupu x napíše na pásku zadaný kód e stroje, který počítá funkci $[[e]]$; vpravo od x napíše zadaný argument y; nad páskou s těmito třemi argumenty pak spustí stroj, který vyčísluje univerzální ternární funkci Φ^3. Výsledkem takového výpočtu, pokud doběhne, je pak hodnota $\Phi^3(e,x,y) = [[e]](x,y)$, jak jsme požadovali.

Detaily konstrukce takového kódu $\varrho(e,y)$ přenecháváme zapálenému čtenáři: jde o zdlouhavé cvičení v kódování a dekódování. □

Například $g(x,y) = x^y$ je rekurzivní funkce, má svůj index $e \in \mathbb{N}$. Pokud zafixujeme $y = 2$, bude $f(x) = g(x,2) = x^2$ opět rekurzivní. Smysl věty je v tom, že existuje *jedna* funkce $\varrho(e,y)$, která efektivně vyčísluje *všechny* takové korespondence. Pro $y = 2$ a index $e \in \mathbb{N}$ funkce x^y je $\varrho(e,2)$ index funkce x^2; stejně tak ale pro $y = 3$ a index $e \in \mathbb{N}$ funkce $x \cdot y$ je $\varrho(e,3)$ index funkce $3x$.

Věta má přirozené zobecnění na více proměnných a libovolnou jejich podmnožinu: nejde o to, že fixujeme jako parametr právě druhý argument ze dvou.

5.8.2 Lemma. *Buď $Y \subseteq \mathbb{N}$ rekurzivně spočetná, buď f libovolná částečná rekurzivní funkce. Potom existuje prostá elementární unární funkce g taková, že pro každé $y \in Y$ je $[[g(y)]]$ funkce f a pro každé $y \notin Y$ je $[[g(y)]]$ prázdná funkce.*

Důkaz. Položme $h(x,y) = f(x)$ pro $y \in Y$. To je částečná rekurzivní funkce, má tedy svůj kód $e \in \mathbb{N}$, pro který je $h(x,y) = [[e]](x,y)$. Položme $g(y) = \varrho(e,y)$, kde ϱ je funkce z předchozí věty. Potom $[[g(y)]](x) = [[\varrho(e,y)]](x) = h(x,y)$, takže pro $y \in Y$ je $[[g(y)]](x) = f(x)$ a pro $y \notin Y$ není $[[g(y)]]$ nikde definována. □

5.8.3 Příklad. Je-li v předchozím lemmatu $Y = K = \{x; x \in dom[[x]]\}$ a $f = \pi^1_1$ (nebo jakákoli jiná totální rekurzivní funkce), máme totální rekurzivní funkci g s následující vlastností: pro $x \in K$ je $[[g(x)]] = \pi^1_1$ a pro $x \notin K$ je $[[g(x)]]$ prázdná funkce. Tato funkce g je svědkem, že $K \leq \mathcal{F} = \{x; [[x]] \text{ je totální}\}$. Zároveň tedy máme $\overline{K} \leq \overline{F}$. Stejně se ukáže $\overline{K} \leq \{x; [[x]] \text{ je prázdná funkce}\}$; tedy množina indexů prázdné funkce není ani rekurzivně spočetná.

5.8.4 Příklad. Množina $K = \{x; x \in dom[[x]]\}$ je maximální mezi všemi rekurzivně spočetnými množinami, ve smyslu výše zavedeného uspořádání. Je-li totiž $A \subseteq \mathbb{N}$ rekurzivně spočetná, máme stejně jako v předchozím příkladě totální rekurzivní funkci g takovou, že pro $a \in A$ je $[[g(a)]] = \pi^1_1$ totální, a tedy $g(a) \in K$, a pro $a \notin A$ je $[[g(a)]]$ prázdná, a tedy $g(a) \notin K$. Tedy $A \leq_g K$.

5.8.5 Věta (Rice). *Buď F libovolná množina unárních rekurzivních funkcí. Potom množina indexů $I = \{x; [[x]] \in F\}$ je rekurzivní jen pro $I = \emptyset$ a $I = \mathbb{N}$.*

Riceova věta je silným negativním výsledkem: říká, že efektivně rozhodovat lze jen o triviálních vlastnostech rekurzivních funkcí. Je-li F množina unárních funkcí, která je netriviální alespoň v tom smyslu, že některé funkce obsahuje a některé ne, pak množina indexů I funkcí z F není rekurzivní. Jedinými rekurzivní případy jsou prázdná množina funkcí a množina všech funkcí.

To znamená, že ani zdánlivě banální množiny indexů jako $\{x; [[x]](3) = 7\}$, $\{x; [[x]]$ je prázdná$\}$ nebo $\{x; [[x]]$ je funkce $n \mapsto 2n\}$ nejsou rozhodnutelné.

Důkaz. Případy $I = \emptyset$ a $I = \mathbb{N}$ jsou jasné. V ostatních případech pevně zvolíme nějakou funkci,[21] která do F padne, a nějakou jinou, která do F nepadne. Je-li $\emptyset \notin F$, zvolme nějakou $f \in F$. Podle předchozího lemmatu existuje totální unární rekurzivní funkce g taková, že pro $y \in K$ je $[[g(y)]]$ funkce $f \in F$ a pro $y \notin K$ je $[[g(y)]]$ prázdná funkce $\emptyset \notin F$. Tato funkce g je svědkem, že $K \leq I$, takže I není rekurzivní. Opačný případ je podobný: pokud $\emptyset \in F$, zvolme nějakou $f \notin F$. Opět máme totální unární rekurzivní funkci g, tentokrát takovou, že pro $y \in K$ je $[[g(y)]]$ funkce $f \notin F$ a pro $y \notin K$ je $[[g(y)]]$ prázdná funkce $\emptyset \in F$. To znamená, že $\overline{K} \leq I$, takže I není ani rekurzivně spočetná. □

5.9 Reprezentace

5.9.1 Definice. Buď T teorie v jazyce aritmetiky. Řekneme, že formule φ s volnými proměnnými $x_1, \ldots, x_k$ ***reprezentuje*** v T relaci $R \subseteq \mathbb{N}^k$, pokud pro každé $n_1, \ldots, n_k \in \mathbb{N}$ platí: je-li $(n_1, \ldots, n_k) \in R$, pak $T \vdash \varphi(\overline{n}_1, \ldots, \overline{n}_k)$; a je-li $(n_1, \ldots, n_k) \notin R$, pak $T \vdash \neg\varphi(\overline{n}_1, \ldots, \overline{n}_k)$. Pokud pro relaci $R \subseteq \mathbb{N}^k$ taková formule φ existuje, řekneme, že R je ***reprezentovatelná*** v T.

5.9.2 Definice. Buď T teorie v jazyce aritmetiky. Řekneme, že formule φ s proměnnými $x_1, \ldots, x_k, y$ ***reprezentuje*** v T částečnou funkci $f : \mathbb{N}^k \to \mathbb{N}$, pokud pro každé $n_1, \ldots, n_k \in dom f$ je $T \vdash \varphi(\overline{n}_1, \ldots, \overline{n}_k, y) \leftrightarrow y = \overline{f(n_1, \ldots, n_k)}$ Pokud pro f taková formule existuje, řekneme, že f je ***reprezentovatelná*** v T.

Platnost ve standardním modelu lze u reprezentovatelných relací a funkcí zachytit formální dokazatelností patřičné formule o numerálech. Nabízí se obecná otázka, ve kterých teoriích lze reprezentovat které relace a funkce. Přímo z definice máme, že ve sporné teorii lze reprezentovat každou relaci a každou funkci, a pro teorie $S \subseteq T \subseteq Th(\mathbb{N})$ platí, že je-li daná relace či funkce reprezentovatelná v S, je reprezentovatelná i v T. Následující tvrzení uvádíme bez důkazu:

5.9.3 Věta. *Buď T bezesporná, rekurzivně axiomatizovaná teorie v jazyce aritmetiky. Potom každá relace a funkce reprezentovatelná v T je rekurzivní.*

5.9.4 Věta. *Každá rekurzivní relace a každá rekurzivní funkce je reprezentovatelná v Robinsonově aritmetice, a tedy také v Peanově aritmetice a v $Th(\mathbb{N})$.*

[21] Důkaz nestojí na nějaké speciální vlastnosti prázdné funkce, jde jen o to, že zrovna prázdnou funkci jsme v lemmatu zvolili pro negativní případ, kdy $g(y)$ není kódem funkce f.

5.10 Aritmetizace

Výrazy formálního jazyka, jakož i další syntaktické pojmy, lze pomocí rekurzivních funkcí kódovat do přirozených čísel. Aritmetika potom prostřednictvím aritmetických sentencí, které mluví o číslech (speciálně tedy o kódech), mluví i o syntaxi kódované teorie – včetně dokazatelnosti v aritmetice samotné.

Ukážeme, jak v aritmetice kódovat syntax formálního jazyka. Nejprve je potřeba přiřadit kódy jednotlivým symbolům. Formule, jakožto konečné posloupnosti symbolů, lze potom kódovat jako konečné posloupnosti čísel, jak jsme předvedli v oddíle o elementárních funkcích. Podobně lze pak kódovat důkazy, jakožto konečné posloupnosti formulí. Předvedeme toto kódování na jazyce aritmetiky, ale podobně lze kódovat důkazy v každé teorii se spočetným jazykem.

Připomeňme, že jazyk aritmetiky má speciální symboly $\mathbf{0}, s, +, *, \leq$, dále symbol $=$ pro rovnost, symboly $\neg$ a $\rightarrow$ pro výrokové spojky, kvantifikátor $\forall$, a neomezeně mnoho symbolů pro proměnné, řekněme x_1, x_2, x_3, atd.

Kódy jednotlivých symbolů zvolme třeba takto: proměnnou x_i kódujeme číslem $c(x_i) = 2i$, pro ostatní symboly položme $c(\mathbf{0}) = 1$, $c(s) = 3$, $c(+) = 5$, $c(*) = 7$, $c(<) = 9$, $c(=) = 11$, $c(\neg) = 13$, $c(\rightarrow) = 15$, $c(\forall) = 17$.

Kód $\#t$ termu t pak definujeme indukcí následovně: pro proměnnou x_i buď $\#x_i = \langle c(x_i)\rangle = \langle 2i\rangle$; pro konstantu $\mathbf{0}$ buď $\#\mathbf{0} = \langle c(\mathbf{0})\rangle = \langle 1\rangle$; pro složené termy buď $\#s(u) = \langle c(s), \#u\rangle, \#(u + v) = \langle c(+), \#u, \#v\rangle, \#(u * v) = \langle c(*), \#u, \#v\rangle$.

Kódy formulí pak definujeme induktivně takto: pro atomické formule buď $\#(u = v)$ číslo $\langle c(=), \#u, \#v\rangle$ a $\#(u < v)$ číslo $\langle c(<), \#u, \#v\rangle$; pro formule vytvořené spojkami buď $\#(\neg\varphi) = \langle c(\neg), \#\varphi\rangle$ a $\#(\varphi \rightarrow \psi) = \langle c(\rightarrow), \#\varphi, \#\psi\rangle$; pro formule vzniklé kvantifikací buď $\#(\forall x_i)\varphi = \langle c(\forall), \#x_i, \#\varphi\rangle$.

Z předchozího víme, že takové kódy jsou jednoznačné, a kódování i dekódování se děje prostřednictvím rekurzivních funkcí.

5.10.1 Cvičení. Spočtěte kód aritmetické formule $(\forall x_1)\neg(x_2 = x_1 + x_1)$; zjistěte, kterou aritmetickou formuli (pokud vůbec nějakou) kóduje 32400000000.

5.10.2 Cvičení. Množina všech kódů $\#x_i$ proměnných je rekurzivní, a stejně tak množina všech kódů $\#t$ aritmetických termů a množina všech kódů $\#\varphi$ aritmetických formulí. Napište explicitně jejich charakteristické funkce.

5.10.3 Cvičení. Definujte rekurzivní predikát $fla(x)$ takový, že $\mathbb{N} \models fla(\overline{n})$ právě když $n \in \mathbb{N}$ je kódem aritmetické formule.

5.10.4 Cvičení. Definujte binární rekurzivní predikát $fvar(x, y)$ tak, aby platilo $\mathbb{N} \models fvar(\overline{m}, \overline{n})$ právě když $m \in \mathbb{N}$ je kód aritmetické formule, která obsahuje volnou proměnnou s kódem $n \in \mathbb{N}$.

Dále chceme efektivně kódovat a rozpoznávat kódy substitucí, totiž termů a formulí, do kterých je dosazeno za nějakou (volnou) proměnnou. Používáme při tom rekurzivní predikát $var(x) \leftrightarrow x = \langle (x)_0\rangle \wedge (\exists y \leq x)((x)_0 = 2 * y)$.

Zavedeme funkci $sub(e, v, w)$ takovou, že pro term t, formuli φ, proměnnou x a term u platí $sub(\#t, \#x, \#u) = \#t_x[u]$, resp. $sub(\#\varphi, \#x, \#u) = \#\varphi_x[u]$. To jest, hodnotou $sub(e, v, w)$ je právě kód termu resp. formule s kódem e, do které je za proměnnou s kódem v dosazen term s kódem w.

Položme $sub(e, v, w) = w$ pokud $var(v) \wedge (e = v)$: jde o triviální případ dosazení do termu, který sám je proměnnou. Je-li $l(e) = 2$, položme $sub(e, v, w) =$

$\langle (e)_0, sub((e)_1, v, w)\rangle$; kódy délky $l(e) = 2$ jsou právě kódy následníků (v případě termů) resp. negací (v případě formulí), a dosadit do $s(u)$ resp. $\neg\varphi$ znamená dosadit do u resp. φ. Je-li $l(e) = 3 \wedge (e)_0 \neq c(\forall)$, položme $sub(e, v, w) = \langle (e)_0, sub((e)_1, v, w), sub((e)_2, v, w)\rangle$; kódy délky $l(e) = 3$ jsou právě kódy binárních funkcí resp. spojek (případ kvantifikace jsme vyloučili), a dosadit do termů $p + q, p * q$ resp. do formule $\varphi \to \psi$ znamená dosadit do podtermů p, q resp. podformulí φ, ψ. Je-li $e = \langle c(\forall), (e)_1, (e)_2\rangle \wedge (e)_1 \neq v$, jde naopak o dosazení do formule začínající kvantifikátorem (za jinou než kvantifikovanou proměnnou); přitom dosadit za volnou proměnnou do $(\forall x)\varphi$ znamená dosadit do φ, položme tedy $sub(e, v, w) = \langle (e)_0, (e)_1, sub((e)_2, v, w)\rangle$. Ve všech ostatních případech položme $sub(e, v, w) = e$; to je triviální případ, kdy nic nedosazujeme. Takto definovaná funkce je rekurzivní, a indukcí podle složitosti termu resp. formule se snadno předvede, že se chová výše požadovaným způsobem, totiž kóduje instance. Speciálně pro kódy $\#\overline{n}$ dosazených numerálů $\overline{n}$ máme $sub(\#t, \#x, \#\overline{n}) = \#t_x[\overline{n}]$ a $sub(\#\varphi, \#x, \#\overline{n}) = \#\varphi_x[\overline{n}]$.

Popsali jsme, jak pomocí rekurzivních funkcí kódovat základní syntax. Ukážeme nyní, jak kódovat další syntaktické pojmy až po pojem dokazatelnosti. Chceme-li efektivně rozpoznávat důkazy, musíme nejprve umět rozpoznat axiomy. Definujeme tedy nejprve predikát $propax(x)$, který popisuje kódy instancí axiomů výrokové logiky. Položme $propax(x)$ právě když

$$(\exists u < x)(\exists v < x)(\exists w < x)[fla(u) \wedge fla(v) \wedge fla(w) \wedge (x = \langle \sigma(\to), u, \langle \sigma(\to), v, u\rangle\rangle \vee \dots)]$$

Kódy tvaru $\langle \sigma(\to), u, \langle \sigma(\to), v, u\rangle\rangle$ zachycují axiomy tvaru $A \to (B \to A)$, kde $\#A = u$ a $\#B = v$; laskavý čtenář sám snadno doplní ostatní případy. Podobně se s pomocí predikátů $fla(x)$ a $fvar(x, y)$ definuje predikát $predax(x)$, který popisuje kódy axiomů predikátové logiky, a konečně predikát $eqax(x)$, který zachycuje axiomy rovnosti. Můžeme pak definovat rekurzivní predikát $lax(x)$, který popisuje kódy všech axiomů logiky.

Je-li navíc ve hře nějaká teorie T, uvažme množinu $\{\#\varphi; \varphi \in T\} \subseteq \mathbb{N}$ všech kódů jejích axiomů. Obecně jakákoli množina formulí je teorie, tato množina tedy nemusí být rekurzivní. V případě, že rekurzivní je, řekneme, že teorie T je *rekurzivně axiomatizovaná* či krátce *rekurzivní*. Je přirozené požadovat, aby uvažovaná teorie měla rekurzivní axiomatiku: v opačném případě bychom nedokázali efektivně rozpoznat ani axiomy, natož důkazy a dokazatelné formule. Zřejmě každá konečná teorie je rekurzivně axiomatizovaná. Snadno se nahlédne, že tradiční teorie z matematické praxe (teorie grup, aritmetika, teorie množin, …) mají rekurzivní axiomatiku.

Chceme-li rozpoznat kódy důkazů (případně důkazů v nějaké dané teorii), musíme dále umět rozpoznat případy užití odvozovacích pravidel, tedy modus ponens a pravidla generalizace. Definujme-li predikát $mp(x, y, z)$ jako

$$fla(x) \wedge fla(y) \wedge fla(z) \wedge y = \langle \sigma(\to), x, z\rangle,$$

potom $mp(x, y, z)$ platí právě když formule s kódem z je odvozena pomocí modus ponens z formulí s kódy x a y. Podobně položíme-li $gen(x, y)$ právě když

$$fla(x) \wedge fla(y) \wedge (\exists v < y)(y = \langle \sigma(\forall), \langle 2 * v\rangle, x\rangle),$$

pak $gen(x, y)$ platí právě když y je kód formule odvozené pravidlem generalizace z formule s kódem x. Podobně jako u předchozích syntaktických pojmů se snadno ukáže, že se jedná o rekurzivní predikáty.

Můžeme konečně přistoupit ke kódování důkazů. Buď T rekurzivně axiomatizovaná teorie; položíme-li $ax_T(x)$ právě když x je kódem axiomu teorie T, bude $ax_T(x)$ rekurzivní predikát. Potom také $ax(x) \leftrightarrow lax(x) \vee ax_T(x)$ je rekurzivní. Můžeme pak definovat predikát $proof_T(x,y)$ následovně

$$l(x) > 0 \wedge y = (x)_{lx} \wedge$$
$$(\forall i < l(x))[ax((x)_i) \vee (\exists j,k < i) mp((x)_j,(x)_k,(x)_i) \vee (\exists j < i) gen((x)_j,(x)_i)]$$

a formální dokazatelnost v teorii T zachytit predikátem

$$thm_T(x) \leftrightarrow (\exists p) proof_T(p,x)$$

Potom $thm_T(x)$ splňují právě kódy formulí dokazatelných v T. Je-li T rekurzivní, je i $\{(x,y); proof_T(x,y)\}$ rekurzivní, kdežto $\{x; thm_T(x)\} \subseteq \mathbb{N}$ je obecně jen rekurzivně spočetná množina. Laskavý čtenář si povšimne souvislosti s předem neomezeným hledáním u minimalizace rekurzivních relací.

5.11 Nerozhodnutelnost a neúplnost

5.11.1 Definice. Teorie T je *rozhodnutelná*, je-li $\{x; thm_T(x)\} \subseteq \mathbb{N}$ rekurzivní.

Teorie T je rozhodnutelná, pokud lze efektivně rozhodovat o tom, které formule jejího jazyka jsou v T dokazatelné. Efektivním rozhodováním se myslí rekurzivnost ve smyslu předchozích odstavců. Obecně je i pro rekurzivní teorii T množina $\{x; thm_T(x)\} \subseteq \mathbb{N}$ kódů dokazatelných formulí jen rekurzivně spočetná: z definice je definičním oborem rekurzivní relace.

5.11.2 Věta (Post)**.** *Úplná rekurzivní teorie je rozhodnutelná.*

Důkaz. Jelikož je T rekurzivní, je $tmh_T(x)$ rekurzivně spočetný predikát. Existuje tedy rekurzivní funkce, jejímž oborem hodnot jsou právě všechny kódy dokazatelných formulí; jinými slovy, existuje efektivní procedura, která postupně vypisuje kódy všech dokazatelných formulí.

Přitom T je úplná, takže pro každou danou sentenci φ se na tomto seznamu dříve nebo později objeví buďto kód $\#\varphi$ nebo kód $\#\neg\varphi$. Rozhodovací procedura pro T je tedy nasnadě: vypisuj postupně všechny kódy dokazatelných formulí; pokud se na tomto seznamu objeví $\#\varphi$, je φ dokazatelná; pokud se na seznamu objeví $\#\neg\varphi$, pak φ není dokazatelná. □

5.11.3 Lemma (diagonální lemma)**.** *Buď $T \subseteq Th(\mathbb{N})$ teorie, která reprezentuje všechny rekurzivní relace a funkce. Pak pro každou aritmetickou formuli $\varphi(x)$ s jednou volnou proměnnou existuje sentence δ taková, že $T \vdash \delta \leftrightarrow \varphi_x[\overline{\#\delta}]$.*

Taková sentence δ tvrdí totéž, co $\varphi_x[\overline{\#\delta}]$; prostřednictvím tvrzení φ o svém vlastním kódu tedy „říká“ toto: „já mám vlastnost φ“; odtud název lemmatu.[22]

Důkaz. Buď $f : \mathbb{N}^2 \to \mathbb{N}$ funkce, která pro kód $n = \#\psi(v)$ formule ψ s volnou proměnnou v a dané číslo $m \in \mathbb{N}$ vrací $f(n,m) = sub(\#\psi, \#v, num(m))$ a ve všech ostatních případech vrací nulu. Podle odstavce o aritmetizaci je tato funkce rekurzivní, jedná se o dosazení numerálu do formule. Podle odstavce o

[22] Čtenář obeznámený s paradoxem lháře už tuší, kam taková situace vede.

reprezentovatelnosti je tato funkce reprezentovatelná v aritmetice, tj. existuje formule $\varrho(x,y,z)$ taková, že $T \vdash \varrho(\overline{n},\overline{m},z) \leftrightarrow z = \overline{f(n,m)}$. Buď nyní $\vartheta(x)$ formule $(\forall z)(\varrho(x,x,z) \to \varphi(z))$. Formule ϑ říká o své proměnné x prostřednictvím reprezentující formule ϱ toto: hodnota $f(x,x)$ má vlastnost φ; to jest, pokud je x kódem nějaké formule s jednou proměnnou, a za tuto proměnnou dosadíme samotný kód x, vznikne instance, jejíž kód má vlastnost φ. Buď konečně δ sentence $\vartheta(\overline{\#\vartheta})$. Ukážeme, že δ je hledaná diagonální sentence.

Z definice je $f(\#\vartheta,\#\vartheta) = \#\delta$; přitom formule ϱ reprezentuje funkci f, takže máme $T \vdash \varrho(\overline{\#\vartheta},\overline{\#\vartheta},z) \leftrightarrow z = \overline{\#\delta}$. Z axiomů rovnosti máme hned $\vdash \varphi_x[\overline{\#\delta}] \leftrightarrow (\forall z)(z = \overline{\#\delta} \to \varphi(z))$, a nahradíme-li podfromuli $z = \overline{\#\delta}$ dokazatelně ekvivalentní formulí $\varrho(\overline{\#\vartheta},\overline{\#\vartheta},z)$, máme $T \vdash \varphi_x[\overline{\#\delta}] \leftrightarrow (\forall z)(\varrho(\overline{\#\vartheta},\overline{\#\vartheta},z) \to \varphi(z))$. Přitom na pravé straně ekvivalence stojí právě sentence δ. □

5.11.4 Věta (Gödelova první věta o neúplnosti)**.** *Buď T bezesporná, rekurzivně axiomatizovaná teorie rozšiřující PA. Potom T je nerozhodnutelná a neúplná.*

Důkaz. Buď T bezesporné rozšíření Peanovy aritmetiky. Pokud je T rozhodnutelná, tj. pokud množina $\{\#\varphi; T \vdash \varphi\} \subseteq \mathbb{N}$ kódů dokazatelných formulí je rekurzivní, buď $\varrho(x)$ artimetická formule, která tuto množinu reprezentuje v T. To znamená, že pro každou sentenci φ je $T \vdash \varrho(\overline{\#\varphi})$ pro $T \vdash \varphi$ a $T \vdash \neg\varrho(\overline{\#\varphi})$ pro $T \nvdash \varphi$. V tom případě buď δ diagonální sentence pro formuli $\neg\varrho(x)$; to znamená, že $T \vdash \delta \leftrightarrow \neg\varrho(\overline{\#\delta})$. Ptejme se, zda je $T \vdash \delta$.

Pokud ano, je též $T \vdash \neg\varrho(\overline{\#\delta})$, což znamená, že $T \nvdash \delta$, jelikož ϱ reprezentuje dokazatelnost v T. Pokud naopak $T \nvdash \delta$, máme $T \vdash \neg\varrho(\overline{\#\delta})$, tedy i $T \vdash \delta$. Ani jeden z případů není možný, neboť T je bezesporná.

Neúplnost potom plyne z Postovy věty: kdyby T byla úplná, pak by díky rekurzivní axiomatice byla rozhodnutelná, což podle předchozího není. □

Uvedený důkaz emuluje v aritmetizované podobě paradox lháře: diagonální sentence δ říká „já jsem nedokazatelná sentence“; taková sentence je potom dokazatelná, právě když dokazatelná není.

Gödelova věta je silným negativním výsledkem, a ve své době znamenala definitivní konec Hilbertova programu, totiž formulovat v predikátovém kalkulu bezespornou, úplnou axiomatiku pro zásadní matematické teorie, aritmetikou počínaje. Ani v případě aritmetiky to není možné. Navíc i kdybychom se pro nějakou sentenci φ, kterou aritmetika ani nedokazuje, ani nevyvrací, arbitrárně rozhodli přijmout do nové, silnější axiomatiky formuli φ (nebo $\neg\varphi$, obě možnosti jsou bezesporné), získali bychom sice silnější, ale stále nerozhodnutelnou (a tedy neúplnou) aritmetiku — totiž pokud má nová teorie zůstat rekurzivně axiomatizovaná; jinak stačí vzít bezesporné zúplnění $Th(\mathbb{N}) = \{\#\varphi; \mathbb{N} \models \varphi\}$, o kterém ale nelze efektivně zjistit ani to, které axiomy obsahuje.

Jako důsledek získáváme následující větu, podle které neexistuje pro délku aritmetických důkazů žádný efektivní horní odhad. Pro aritmetickou sentenci φ označme její délku (tj. počet symbolů) jako $|\varphi|$. Je-li φ dokazatelná v aritmetice, označme jako $\pi(\varphi)$ délku jejího nejkratšího možného formálního důkazu.

5.11.5 Věta. *Buď T bezesporné rozšíření Robinsonovy aritmetiky s rekursivní axiomatikou. Pak pro každou rekursivní funkci $f : \mathbb{N} \to \mathbb{N}$ existuje nekonečně mnoho aritmetických sentencí $\varphi \in Thm(T)$, pro které je $\pi(\varphi) > f(|\varphi|)$.*

Důkaz. Sporem: předpokládejme, že pro nějakou dostatečně rychle rostoucí rekursivní funkci $f : \mathbb{N} \to \mathbb{N}$ je $\pi(\varphi) \leq f(|\varphi|)$ pro všechny $\varphi \in Thm(T)$ až na konečně mnoho vyjímek. Takovou funkci f stačí pozměnit na konečně mnoha místech tak, aby uvedený odhad platil pro vůbec všechny dokazatelné sentence (přitom takto vylepšená funkce je stále rekursivní). Pak je ovšem T rozhodnutelná: pro zadanou sentenci φ vyzkoušej všechny důkazy délky $f(|\varphi|)$; takových je jen konečně[23] mnoho. Je-li některý z nich důkazem φ, je dokazatelná v T; v opačném případě dokazatelná není. To je efektivní rozhodovací procedura pro dokazatelnost v T. Přitom T je podle Gödelovy věty nerozhodnutelná. □

Laskavý čtenář si povšimne souvislosti s příkladem 5.5.2: tak jako neexistuje efektivní odhad pro délku výpočtu nad daným vstupem, neexistuje eni efektivní odhad pro délku důkazu formule dané délky.

Následující věta jde v opačném směru nežli Gödelova věta o neúplnosti aritmetiky. Úplnost teorie T znamená z definice tolik, že T každou sentenci buďto dokazuje nebo vyvrací. Přirozeným zobecněním je rozvolnit tento požadavek jen na některé formule, a přirozeným kandidátem na takovou třídu formulí je omezení jejich syntaktické složitosti.

5.11.6 Definice. Formule φ jazyka aritmetiky je *omezená*, pokud je buďto atomická, nebo tvaru $\neg\psi$, $\psi \wedge \vartheta$, $\psi \vee \vartheta$, $\psi \to \vartheta$ či $\psi \leftrightarrow \vartheta$, kde ψ a ϑ jsou omezené, nebo je tvaru $(\forall x \leq y)\psi$ či $(\exists x \leq y)\psi$, kde ψ je omezená. Formule φ je Σ_1, pokud je tvaru $(\exists x)\psi$, kde ψ je omezená.

V témže duchu lze definovat Π_1 formule tvaru $(\forall x)\psi$, kde ψ je omezená, a dále indukcí třídy Σ_{n+1} a Π_{n+1} formulí tvaru $(\exists)\psi$ respektive $(\forall x)\psi$, kde ψ je Ψ_n formule, respektive Σ_n formule. Tyto třídy tvoří tzv. *aritmetickou hierarchii*, kterou se dále zabývat nebudeme.

5.11.7 Věta (o Σ_1-úplnosti aritmetiky)**.** *Buď T bezesporné, rekurzivně axiomatizované rozšíření aritmetiky, buď φ aritmetická Σ_1-sentence. Potom $T \vdash \varphi \to \Box\varphi$, a v případě $\mathbb{N} \models \varphi$ je též $T \vdash \varphi$.*

Pro důkaz Gödelovy věty jsme zavedli několik technických prostředků: aritmetizovali jsme syntaktické pojmy až po formální dokazatelnost. Pro teorii T a aritmetickou formuli φ pišme[24] v dalším stručně $\Box\varphi$ místo $thm_T(\overline{\#\varphi})$. Ptáme se nyní, nakolik tento aritmetizovaný pojem formální dokazatelnosti reflektuje to, co o dokazatelnosti známe z kapitoly o predikátové logice.

5.11.8 Věta (Löbovy podmínky)**.** *Buď T bezesporné, rekurzivně axiomatizované rozšíření aritmetiky. Potom predikát $\Box\varphi$ splňuje následující.*

(L1) $T \vdash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$

(L2) *Pokud $T \vdash \varphi$, pak i $T \vdash \Box\varphi$*

(L3) $T \vdash \Box\varphi \to \Box\Box\varphi$

[23] Přísně vzato je důkazů dané délky nekonečně mnoho: pokud například v důkaze přejmenujeme nějakou vázanou proměnnou, získáme formálně jiný důkaz téže délky. Až na takové umělé rozdíly je však důkazů dané délky jen konečně mnoho.

[24] Značení je vypůjčeno z modální logiky, kterou se v tomto textu jinak nezabýváme. Modální logika klasickou predikátovou logiku podstatně rozšiřuje o pojem *modality*: některá tvrzení platí *nutně*, některá jsou *možná*. Tyto dva případy se typicky značí $\Box\varphi$ a $\Diamond\varphi$.

Důkaz. Podmínka L1 říká, že predikát $\Box\varphi$ reflektuje dokazatelnost pomocí modus ponens. Důkaz lze provést syntakticky: existuje term $t(x,y)$ takový, že $T \vdash proof_T(x, \overline{\#\varphi}) \wedge proof_T(y, \overline{\#(\varphi \to \psi)}) \to proof_T(t(x,y), \overline{\#\psi})$ pro jakékoli formule φ, ψ. Stačí vzít $t(x,y) = x^\frown y^\frown z$, kde $a^\frown b$ je kód zřetězené posloupnosti, pokud $(\exists u < x)(\exists v < y)(proof_T(x,u) \wedge proof_T(y,v) \wedge mp(x,y,z))$

Při důkazu L2 a L3 využijeme Σ_1-úplnost. Totiž $\Box\varphi$ je Σ_1-sentence pro jakoukoli formuli φ, takže ze Σ_1-úplnosti máme $T \vdash \Box\varphi \to \Box\Box\varphi$, což je podmínka L3. Pokud je $T \vdash \varphi$, potom $\mathbb{N} \models \Box\varphi$, máme tedy Σ_1-sentenci platnou v $\mathbb{N}$, takže ze Σ_1-úplnosti máme též $T \vdash \Box\varphi$, jak požaduje podmínka L2. □

Z věty o korektnosti víme, že pokud aritmetika dokazuje formuli φ, potom φ musí platit ve standardním (jakož i každém jiném) modelu. Aritmetizovanou podobou tohoto tvrzení je princip *reflexe*, totiž tvrzení $T \vdash \Box\varphi \to \varphi$. Nabízí se potom otázka, pro které aritmetické formule tento princip platí. Nejlepší možnou odpověď dává následující věta: právě pro ty formule, které jsou v aritmetice skutečně dokazatelné.

5.11.9 Věta (Löb). *Buď T bezesporné, rekurzivně axiomatizované rozšíření aritmetiky, buď φ aritmetická formule. Potom $T \vdash \Box\varphi \to \varphi$ právě když $T \vdash \varphi$.*

Důkaz. Jeden směr je snadný: je-li $T \vdash \varphi$, pak z instance $\varphi \to (\Box\varphi \to \varphi)$ axiomu H1 máme hned i $T \vdash \Box\varphi \to \varphi$. Buď tedy naopak $T \vdash \Box\varphi \to \varphi$. Podle diagonálního lemmatu existuje diagonální sentence pro formuli $thm_T(x) \to \varphi$, totiž aritmetická sentence δ taková, že $T \vdash \delta \leftrightarrow (\Box\delta \to \varphi)$. S využitím Löbových podmínek ukážeme, že T dokazuje φ. Při $T \vdash \delta \to (\Box\delta \to \varphi)$ máme podle L2 také $T \vdash \Box(\delta \to (\Box\delta \to \varphi))$, načež podle L1 také $T \vdash \Box\delta \to \Box(\Box\delta \to \varphi)$. Potom je ale podle L1 a věty o dedukci též $T, \Box\delta \vdash \Box\Box\delta \to \Box\varphi$. Přitom podle L2 a věty o dedukci je je $T, \Box\delta \vdash \Box\Box\delta$, tedy i $T, \Box\delta \vdash \Box\varphi$. Předpokládáme přitom $T \vdash \Box\varphi \to \varphi$, takže $T, \Box\delta \vdash \varphi$. Ukážeme, že i předpoklad $\Box\delta$ je dokazatelný v T. Z věty o dedukci máme $T \vdash \Box\delta \to \varphi$. Tato sentence je ovšem dokazatelně ekvivalentní právě se sentencí δ, takže $T \vdash \delta$, načež $T \vdash \Box\delta$ podle L2. □

Nad Gödelovou větou o neúplnosti aritmetiky se nabízí následující úvaha. Pokud aritmetika není úplná, potom existují aritmetická tvrzení, která sice platí ve standardním modelu $\mathbb{N}$, ale v aritmetice nejsou dokazatelná. O která tvrzení se jedná? Jistě neradi bychom se smiřovali s nedokazatelností nějakých užitečných tvrzení o přirozených číslech; diagonální sentence, použitá v důkaze Gödelovy věty, je však jiné povahy — jde o sebereferenční, paradoxní tvrzení o sobě, totiž formalizovaný, aritmetizovaný paradox lháře. Skeptik tak může namítnout, že tím jsme o mnoho nepřišli. Odpovědí na tuto námitku je druhá Gödelova věta o neúplnosti: takovým nedokazatelným tvrzením je například (aritmetizované) tvrzení o vlastní bezespornosti.

5.11.10 Věta (Gödel). *Buď T bezesporné, rekurzivně axiomatizované rozšíření Peanovy aritmetiky. Potom T nedokazuje $\neg\Box(0=1)$.*

Důkaz. Předpokládejme, že $T \vdash \neg\Box(0=1)$. Z výrokového axiomu H1 pak ihned máme i $T \vdash \Box(0=1) \to (0=1)$. To podle Löbovy věty znamená právě tolik, že $T \vdash 0=1$, což není možné, pokud je T bezesporná. □

5.12 Přepisovací systémy

Popíšeme na závěr ještě jeden formalismus, který zachycuje intuitivní pojem algoritmu. Markovovy přepisovací systémy (A. A. Markov, 1956) představují opět tutéž výpočetní sílu jako rekurzivní funkce a Turingovy stroje. Ukážeme na nich další algoritmicky neřešitelný problém, totiž problém slov v pologrupách.

5.12.1 Definice. Bud A nějaká konečná množina (*abeceda*). Její prvky budeme nazývat *symboly*, a konečné posloupnosti symbolů *slova*. Každá dvojice slov je potom *přepisovacím pravidlem*, které buďto je nebo není *koncové*. Konečná posloupnost takových pravidel pak tvoří *Markovův algritmus*.

Popíšeme nejprve intuitivně chod takového algoritmu. Vstupem je nějaké slovo v dané abecedě. Pokud nějaký podřetezec tohoto slova tvoří levou část nějakého pravidla, přepíše se odpovídající pravou částí; přitom použijeme vždy první možné pravidlo (pravidla jsou předem uspořádána) a nahrazujeme vždy nejlevější možný výskyt. S takto vzniklým slovem provádíme dále totéž; skončíme po použití koncového pravidla, nebo když žádné pravidlo nelze použít.

5.12.2 Příklad. Následující pravidla uspořádají písmena `a,b,c` podle abecedy.

```
ba ab
ca ac
cb bc
```

5.12.3 Příklad. Následující sada přepisovacích pravidel[25] otáčí zadané slovo v abecedě `a,b,c`. Používá při tom pomocný (*neterminální*) symbol `+`, přičemž symbolem `_` značíme prázdné slovo a tečkou označujeme koncová pravidla.

```
+++ ++
++a a++
++b b++
++c c++
++  _.
+aa a+a
+ab b+a
+ac c+a
+ba a+b
+bb b+b
+bc c+b
+ca a+c
+cb b+c
+cc c+c
_   +
```

5.12.4 Definice. Řekneme, že sada R pravidel $(u_1, v_1), \ldots, (u_n, v_n)$ ***přepisuje*** slovo u na slovo v, pokud existuje $i \leq n$ a slova s, t tak, že $u = su_it, v = sv_it$, pro každé $j \leq n$ je $u_j \sqsubseteq u \to i \leq j$, a kdykoli je $xu_i = yu_iz$, je z prázdné. Píšeme pak $u \vdash v$. ***Výpočtem*** algoritmu R je každá posloupnost slov w_i takových, že každé $w_{i-1} \vdash w_i$. Je-li takový výpočet konečný, tj. nějaké $w_{i-1} \vdash w_i$ se získá použitím koncového pravidla nebo už žádné pravidlo použít nelze, píšeme $R(w_0) = w_i$.

[25] Běh Markovových algoritmů emuluje například `https://github.com/janstary/ma`

Například pravidlo $a \to o$ přepisuje slovo $padat$ na slovo $podat$, ale nikoli na slovo $padot$: pro $x\underline{u} = p\underline{a} = y\underline{u}z$ je z prázdné (jedná se o nejlevější výskyt a), ale pro $x\underline{u} = pad\underline{a} = p\underline{a}da = y\underline{u}z$ je $z = da$ neprázdné.

Markovovy algoritmy obecně přepisují slova na slova. Přijmeme-li nějakou konvenci pro zápis čísel jako slov, budeme mít k dispozici aparát, který číslům přiřazuje čísla — neboli vyčísluje funkce. Nabízí se unární notace, kdy počtem znaků 1 vyjadřujeme přirozené číslo, a symbol $|$ používáme jako oddělovač; slovo $111|11$ pak můžeme považovat za zápis dvojice $(3, 2)$. Při této konvenci pak algoritmus s jediným koncovým pravidlem $\epsilon \to 1$ přepisuje každé číslo na jeho následníka, a v tomto smyslu počítá funkci $s(x) = x + 1$. Podobně algoritmus s jediným koncovým pravidlem $| \to \epsilon$ počítá funkci $x + y$.

5.12.5 Příklad. Následující pravidla počítají největšího společného dělitele dle Euklidova algoritmu, opět s použitím pomocných symbolů `a`,`b`,`c`.

```
1a   a1
1|1  a|
1|   |b
b    1
a    c
c    1
|    _
```

5.12.6 Věta. *Markovovy algoritmy vyčíslují právě rekurzivní funkce.*

Důkaz. Naznačíme jen myšlenku důkazu, je podobný jako u Turingových strojů. V jednom směru je potřeba zakódovat běh Markovových algoritmů do rekurzivních funkcí, to se však provede skoro stejně jako při kódování Turingových strojů. V opačném směru je potřeba nejprve napsat pravidla, která vyčíslují následníka a projekce (což je dosti přímočaré) a poté ukázat, jak kombinacemi existujících pravidel získat nová pravidla pro funkce vytvořené skládáním, primitivní rekurzí a minimalizací. To přenecháváme čtenáři jako programovací úlohu.

Vzhledem k ekvivalenci rekurzivních a Turingovsky vyčíslitelných funkcí může být lehčí ukázat, že každá Turingovsky vyčíslitelná funkce je zároveň Markovovsky vyčíslitelná. Její Turingův stroj lze totiž poměrně snadno emulovat následujícím Markovovým algoritmem; abeceda obsahuje znaky `0` a `1`, znaky pro jména stavů, a dva pomocné znaky `*` a `#`.

`bsr`	`tbw`	za každou instrukci `srwLt` a každé $b \in \{0, 1\}$
`sr`	`t0w`	za každou instrukci `srwLt`
`sr`	`tw`	za každou instrukci `srwNt`
`srb`	`wtb`	za každou instrukci `srwRt` a každé $b \in \{0, 1\}$
`sr`	`wt0`	za každou instrukci `srwRt`
`s`	`_`	za každý stav `s`
`#0`	`#`	
`#1`	`1.`	
`*b`	`#sb`	pro výchozí stav `s` a každé $b \in \{0, 1\}$
`*`	`#s0`	pro výchozí stav `s`
`_`	`*`	

Prvních pět pravidel emuluje samotnou přechodovou funkci; výskytem pomocného symbolu pro vnitřní stav zachycujeme momentální konfiguraci stroje,

totiž stav a polohu na pásce. Zvlášť ošetřujeme případ, kdy překračujeme levý resp. pravý konec slova, a připisujeme na emulovanou pásku další nulu. Další tři pravidla pásku čistí od pomocných symbolů a mažou úvodní nuly, aby výsledek začínal opět první jedničkou. Poslední tři pravidla provádějí inicializaci: nastavují výchozí stav a polohu na pásce (včetně prázdné pásky). □

5.12.7 Příklad. Popíšeme Markovův algoritmus emulující Turingův stroj pro funcki $x + y$, jak je popsáno výše. Turingův stroj jen přepíše oddělující nulu jedničkou a umaže dvě jedničky ze začátku. U každé jeho instrukce uvádíme odpovídající přepisovací pravidla; pomocná pravidla vynecháváme.

```
A 1 1 R A     A11 → 1A1, A10 → 1A0, A1 → 1A0
A 0 1 L B     1A0 → B11, 0A0 → B01, A0 → B01
B 1 1 L B     1B1 → B11, 0B1 → B01, B1 → B01
B 0 0 R C     B01 → 0C1, B00 → 0C0, B0 → 0C0
C 1 0 R D     C11 → 0D1, C10 → 0D0, C1 → 0D0
D 1 0 R Z     D11 → 0Z1, D10 → 0Z0, D1 → 0Z0
```

Výpočet tohoto Turingova stroje a Markovova algoritmu pak například na vstupu $3+2$ proběhne následovně. U Turingova stroje vyznačujeme pozici hlavy a stav, u Markovova algoritmu naznačujeme použité pravidlo.

```
                 _11110111      *
                 *11110111      #A1
A   0111101110   #A11110111     1A1
A   0111101110   #1A1110111     1A1
A   0111101110   #11A110111     1A1
A   0111101110   #111A10111     1A0
A   0111101110   #1111A0111     B11
B   0111111110   #111B11111     B11
B   0111111110   #11B111111     B11
B   0111111110   #1B1111111     B11
B   0111111110   #B11111111     B01
B   0111111110   #B011111111    0C1
C   0111111110   #0C111111111   0D1
D   0011111110   #00D11111111   0Z1
Z   0001111110   #000Z1111111   _
                 #000111111     #
                 #00111111      #
                 #0111111       #
                 #111111        1
                 111111
```

5.13 Herbrandovská vyčíslitelnost

Jako další ekvivalentní formulaci efektivní vyčíslitelnosti popíšeme odvozování rovnic o aritmetických termech. Základní myšlenka je tato: funkce f je vyčíslitelná, pokud pro každou funkční hodnotu $f(n_1, n_2, \ldots, n_k) = m$ můžeme formálně odvodit rovnost $f(\overline{n}_1, \overline{n}_2, \ldots, \overline{n}_k) = \overline{m}$ o příslušných numerálech.

Z oddílu o formálních jazycích a z aritmetiky již známe pojmy *term* a *numerál*. V dalším uvažujeme jen termy sestavené z proměnných, numerálů, funkčního symbolu s pro následníka a dalších předem daných funkčních symbolů.

Uvažujeme potom *systémy rovností* termů $p_1 = q_1, p_2 = q_2, \ldots, p_n = q_n$ a následující pravidla, pomocí kterých lze z daného systému odvozovat další rovnosti.

Pravidlo *dosazení*: z rovnosti termů $p = q$ odvoď rovnost, která vznikne nahrazením všech výskytů nějaké proměnné týmž numerálem.

Pravidlo *nahrazení*: z rovnosti $p = q$, která neobsahuje žádné proměnné, a rovnosti $f(\overline{n}_1, \ldots, \overline{n}_k) = \overline{m}$, odvoď rovnost, která vznikne z rovnosti $p = q$ nahrazením některých výskytů termu $f(\overline{n}_1, \ldots, \overline{n}_k)$ numerálem $\overline{m}$.

5.13.1 Definice. Posloupnost rovností $e_1, \ldots, e_n$ je *odvozením* rovnosti e ze systému E, pokud e_n je rovnost e a pro $i \leq n$ je rovnost e_i buďto (a) ze systému E nebo (b) je odvozena z nějaké přechozí rovnosti pravidlem dosazení nebo (c) je odvozena z nějakých předchozích rovností pravidlem nahrazení. Pokud takové odvození existuje, řekneme, že e je *odvoditelná* z E, a píšeme $E \vdash e$.

5.13.2 Příklad. Následující posloupnost je odvozením ze systému rovností $f_1(x) = s(x), f_2(x, \overline{0}) = x, f_2(x, f_1(y)) = f_1(f_2(x, y))$.

$$
\begin{aligned}
f_1(x) &= s(x) \\
f_1(\overline{0}) &= s(\overline{0}), \text{ tj. } f_1(\overline{0}) = \overline{1} \\
f_1(\overline{1}) &= s(\overline{1}), \text{ tj. } f_1(\overline{1}) = \overline{2} \\
f_1(\overline{3}) &= s(\overline{3}), \text{ tj. } f_1(\overline{3}) = \overline{4} \\
f_1(\overline{4}) &= s(\overline{4}), \text{ tj. } f_1(\overline{4}) = \overline{5} \\
f_2(x, \overline{0}) &= x \\
f_2(\overline{3}, \overline{0}) &= \overline{3} \\
f_2(x, f_1(y)) &= f_1(f_2(x, y)) \\
f_2(\overline{3}, f_1(y)) &= f_1(f_2(\overline{3}, y)) \\
f_2(\overline{3}, f_1(\overline{0})) &= f_1(f_2(\overline{3}, \overline{0})) \\
f_2(\overline{3}, \overline{1}) &= f_1(f_2(\overline{3}, \overline{0})) \\
f_2(\overline{3}, \overline{1}) &= f_1(\overline{3}) \\
f_2(\overline{3}, \overline{1}) &= \overline{4} \\
f_2(\overline{3}, f_1(\overline{1})) &= f_1(f_2(\overline{3}, \overline{1})) \\
f_2(\overline{3}, \overline{2}) &= f_1(f_2(\overline{3}, \overline{1})) \\
f_2(\overline{3}, \overline{2}) &= f_1(\overline{4}) \\
f_2(\overline{3}, \overline{2}) &= \overline{5}
\end{aligned}
$$

5.13.3 Definice. Systém E *Herbrandovsky vyčísluje* funkci $f : \mathbb{N}^k \to \mathbb{N}$, pokud pro každé $n_1, \ldots, n_k \in dom(f)$ a každé $m \in \mathbb{N}$ platí: $E \vdash f(\overline{n}_1, \ldots, \overline{n}_k) = \overline{m}$ právě když $f(n_1, \ldots, n_k) = m$. Pokud pro danou funkci takový systém existuje, řekneme, že je *Herbrandovsky vyčíslitelná*.

Pojem odvození ze systému rovnic je dosti podobný pojmu formálního důkazu. Předem však nejsou dány žádné axiomy — Herbrandovsky vyčíslitelná funkce může být vyčíslena z libovolného systému rovnic. Jediným omezením je syntaktická podoba termů v těchto rovnicích.

5.13.4 Příklad. Systém $f(x) = 0$ vyčísluje konstantní nulu. Systém $f(x) = s(x)$ vyčísluje funkci následníka. Systém $f(x_1, \ldots, x_k) = x_k$ vyčísluje projekci π_i^k. Systém $f(x, 0) = x, f(x, s(y)) = s(f(x, y))$ vyčísluje binární sčítání.

5.13.5 Příklad. Herbrandovsky vyčíslitelná funkce nemusí být totální. Například systém E rovností $f(x) = 0, f(x) = x$ vyčísluje jednobodovou funkci $f(0) = 0$. Totiž $E \vdash f(\overline{0}) = \overline{0}$ máme okamžitě, kdežto v bodě $n \neq 0$ není funkce f definovaná: kdyby ano, muselo by z definice být $f(n) = 0$ díky $E \vdash f(\overline{n}) = 0$ a zároveň $f(n) = n$ díky $E \vdash f(\overline{n}) = \overline{n}$.

5.13.6 Věta. *Každá částečná rekurzivní funkce je Herbrandovsky vyčíslitelná.*

Důkaz. Následník, konstantní nula a projekce jsou pokryty příkladem výše. Indukcí podle složitosti ukážeme, že třída Herbrandovsky vyčíslitelných funkcí je uzavřena na skládání, primitivní rekurzi a minimalizaci.

Buď funkce $h(n_1, \ldots, n_k) = f(g_1(n_1, \ldots, n_k), \ldots, g_m(n_1, \ldots, n_k))$ složená z funkcí g_i a funkce f, přičemž funkci g_i vyčísluje systém E_i a funkci f vyčísluje systém E_f. Můžeme předpokládat, že systémy $E_f, E_1, \ldots, E_m$ používají navzájem různá jména pro všechny použité funkce i proměnné. Systém E, který vyčísluje funkci h, se potom získá rozšířením posloupnosti všech rovnic z $E_1, \ldots E_m, E_f$ o rovnici $h(x_1, \ldots, x_k) = f(g_1(x_1, \ldots, x_k), \ldots, g_m(x_1, \ldots, x_k))$. Je-li totiž $(n_1, \ldots, n_k) \in dom(h)$, je nutně $(n_1, \ldots, n_k) \in dom(g_i)$ pro každé $i \leq m$, a je-li $g_i(n_1, \ldots, n_k) = p_i$, máme již $E_i \vdash g_i(\overline{n}_1, \ldots, \overline{n}_k) = \overline{p}_i$; zároveň pro $f(p_1, \ldots, p_m) = p$ máme $E_f \vdash f(\overline{p}_1, \ldots, \overline{p}_m) = \overline{p}$. Potom ale pro $h(n_1, \ldots, n_k) = f(g_1(n_1, \ldots, n_k), \ldots, g_m(n_1, \ldots, n_k)) = p$ máme $E \vdash h(\overline{n}_1, \ldots, \overline{n}_m) = \overline{p}$. Je-li naopak $E \vdash h(\overline{n}_1, \ldots, \overline{n}_m) = \overline{p}$, musí být $E \vdash f(\overline{p}_1, \ldots, \overline{p}_m) = \overline{p}$ a $E \vdash g_i(\overline{n}_1, \ldots, \overline{n}_k) = \overline{p}_i$ pro každé $i \leq m$, neboť jinde se jméno funkce h nevyskytuje. To ale znamená $E_f \vdash f(\overline{p}_1, \ldots, \overline{p}_m) = \overline{p}$ a $E_i \vdash g_i(\overline{n}_1, \ldots, \overline{n}_k) = \overline{p}_i$, neboť ani jména $f, g_1, \ldots, g_m$ se jinde nevyskytují. Podle indukčního předpokladu tedy $g_i(n_1, \ldots, n_k) = p_i$ a $f(p_1, \ldots, p_m) = p$, neboli $h(n_1, \ldots, n_k) = p$.

Buď h funkce vytvořená z f, g primitivní rekurzí jako $h(n_1, \ldots, n_k, 0) = f(n_1, \ldots, n_k)$ a $h(n_1, \ldots, n_k, n+1) = g(n_1, \ldots, n_k, n, h(n_1, \ldots, n_k, n))$, přičemž funkce f, g vyčíslují systémy E_f, E_g. Stačí potom za rovnice z E_f, E_g připsat

$$h(x_1, \ldots, x_k, 0) = f(x_1, \ldots, x_k)$$
$$h(x_1, \ldots, x_k, s(y)) = g(x_1, \ldots, x_k, y, h(x_1, \ldots, x_k, y))$$

a získáme systém E, který vyčísluje funkci h. Je-li totiž $h(n_1, \ldots, n_k, n) = p$, máme očividně $E \vdash h(\overline{n}_1, \ldots, \overline{n}_k, \overline{n}) = \overline{p}$. Je-li naopak $E \vdash h(\overline{n}_1, \ldots, \overline{n}_k, \overline{n}) = \overline{p}$, máme předvést $h(n_1, \ldots, n_k, n) = p$; to se snadno ukáže indukcí podle $n \in \mathbb{N}$.

Buď konečně $h(n_1, \ldots, n_k) = \mu n[g(n_1, \ldots, n_k, n) = 0]$ vytvořená minimalizací funkce g, kterou vyčísluje systém E_g. Buď E systém rovnic rozšiřující E_g o rovnice E_m vyčíslující násobení $m(n_1, n_2) = n_1 \cdot n_2$ (takový systém existuje, neboť podle předchozího jsou elementární funkce Herbrandovsky vyčíslitelné) a následující rovnice E_f vyčíslující multiplikaci $f = \Pi g$ funkce g:

$$f(x_1, \ldots, x_k, 0) = 1$$
$$f(x_1, \ldots, x_k, s(y)) = m(f(x_1, \ldots, x_k, y), g(x_1, \ldots, x_k, y))$$

Minimální n splňující $g(n_1, \ldots, n_k, n) = 0$ budeme hledat pomocí funkce f: dokud jsou hodnoty $g(n_1, \ldots, n_k, n)$ definované a nenulové, je i multiplikace $f(n_1, \ldots, n_k, n+1)$ definovaná a nenulová; jakmile pro nějaké $n \in \mathbb{N}$ nastane $g(n_1, \ldots, n_k, n) = 0$, bude součin $f(n_1, \ldots, n_k, n+1)$ nulový. Přidáním rovnic

$$\pi(s(x_1), 0, x_3) = x_3$$
$$h(x_1, \ldots, x_k) = \pi(f(x_1, \ldots, x_k, y), f(x_1, \ldots, x_k, s(y)), y)$$

vznikne systém E vyčíslující h: je-li $h(n_1,\ldots,n_k) = \mu n[g(n_1,\ldots,n_k,n) = 0] = p$, je $f(n_1,\ldots,n_k,p) = q+1 \neq 0$ a $f(n_1,\ldots,n_k,p+1) = 0$, takže máme také $E_f \vdash f(\overline{n}_1,\ldots,\overline{n}_k,\overline{p}) = s(\overline{q})$ a $E_f \vdash f(\overline{n}_1,\ldots,\overline{n}_k,s(\overline{p})) = 0$. Potom je ale $E \vdash h(\overline{n}_1,\ldots,\overline{n}_k) = \pi(s(\overline{q}),0,\overline{p})$; přitom máme $E \vdash \pi(s(\overline{q}),0,\overline{p}) = \overline{p}$, tedy $E \vdash h(\overline{n}_1,\ldots,\overline{n}_k) = \overline{p}$. Je-li naopak $E \vdash h(\overline{n}_1,\ldots,\overline{n}_k) = \overline{p}$, musí být zároveň $E \vdash \pi(s(\overline{q}),0,\overline{p}) = \overline{p}$, neboť v jiné rovnici se jméno h nevyskytuje, a zároveň je $E_f \vdash f(\overline{n}_1,\ldots,\overline{n}_k,\overline{p}) = s(\overline{q})$ a $E_f \vdash f(\overline{n}_1,\ldots,\overline{n}_k,s(\overline{p})) = 0$. To ale znamená, že $f(n_1,\ldots,n_k,p) = q+1 \neq 0$ a $f(n_1,\ldots,n_k,p+1) = 0$; v tom případě je p nejmenší $n \in \mathbb{N}$ takové, že $g(n_1,\ldots,n_k,n) = 0$, neboli $h(n_1,\ldots,n_k) = p$. □

V opačném směru chceme nyní dokázat, že každá Herbrandovsky vyčíslitelná funkce je rekursivní. Podobně jako v případě Turingových strojů využijeme aritmetizaci pojmů rekursivními funkcemi: zavedeme kódy proměnných a jmen funkcí, kódy rovnic, systémů rovnic atd, až po pojem Herbrandovské vyčíslitelnosti samotné.

Kódem proměnné x_n bude číslo $4n$, kódem funkčního symbolu f_n bude $4n+2$. Lichými čísly budeme kódovat ostatní symboly: kódem konstanty **0** bude 1, kódem rovnosti $=$ číslo 3, kódem symbolu s pro následníka číslo 5, závorky (a) budou kódovat čísla 7 a 9. Všechny kódy budeme v dalším značit znakem $\#$, tedy např. $\#x_n = 4n$ a $\#f_n = 4n+2$, a stejně tak pro složitější syntaktické útvary.

Očividným způsobem též zavedeme elementární predikát $var(x)$, který platí právě o kódech proměnných, a podobně $fun(x)$ pro kódy funkcí.

Po přiřazení kódů jednotlivým symbolům můžeme kódovat i další syntaktické konstrukce, stejným způsobem jako v oddílu o aritmetice. Kódem $\#s(x)$ termu $s(x_n)$ bude kód čtveřice $\#s, \#(, \#x_n, \#)$, tedy číslo $2^5 \cdot 3^7 \cdot 5^{4n} \cdot 7^9$; analogicky pak pro kód termu $f(x_1,\ldots,x_n)$. Snadno pak zavedeme elementární predikát $term(x)$, který splňují právě kódy termů, a predikát $num(x)$, který jako speciální případ rozezná kódy numerálů.

Jsou-li r, s termy, pak kódem rovnosti $r = s$ je číslo $2^3 \cdot 3^{\#r} \cdot 5^{\#s}$, a kódem systému rovností $r_1 = s_1, \ldots, r_n = s_n$ je číslo $\Pi_{i \leq n} p_i^{\#r_i = s_i}$. Zavedeme pak predikáty $eq(x)$ a $sys(x)$, které platí právě o kódech rovností resp. systémů.

5.13.7 Věta. *Každá Herbrandovsky vyčíslitelná funkce je rekursivní.*

Důkaz. Je-li f vyčíslitelná systémem rovnic E, buď e kód tohoto systému. Potom je $f(x_1,\ldots,x_k) = val(\mu y[H_k(e,x_1,\ldots,x_k,y)])$; přitom predikát H_k je rekursivní, takže f je částečná rekursivní funkce. □

Literatura

[Ba] J. Barwise, *An Introduction to First-Order Logic*, in *Handbook of Mathematical logic*, Elsevier, 1977

[BŠ] B. Balcar, P. Štěpánek, *Teorie množin*, Academia, 2000

[Ca] G. Cantor, *Contributions to the Theory of Transfinite Numbers*, Dover Publications, 1915

[HBA] J. D. Monk (ed.), *Handbook of Boolean Algebras*, North-Holland, 1989

[J] T. Jech, *Set Theory*, Springer Verlag, 2003

[Kl] S. C. Kleene, *Mathematical logic*, Dover Publications, 1967

[Ku] K. Kunen, *The Foundations of Mathematics*, College Publications, 2012

[Me] E. Mendelsohn, *Introduction to mathematical logic*, Van Nostrand, 1979

[Mo] J. D. Monk, *Mathematical logic*, Springer Verlag, 1976

[P] E. Post, *Introduction to a General Theory of Elementary Propositions*, American Journal of Mathematics 43:3 (1921), 163–185

[So] A. Sochor, *Klasická matematická logika*, Karolinum, 2001

[Št] P. Štěpánek, *Matematická logika*, skripta MFF UK, 1982

[Šv] V. Švejdar, *Logika: neúplnost, složitost, nutnost*, Academia, Praha 2002

[T] A. Tarski, *Logic, Semantics, Metamathematics*, Clarendon Press, 1956