Diskrétní struktury 2

slajdy k přednáškám

Miroslav Kolařík

Obsah

Obsah

Připomeňme si nejprve, jak se počítají kombinační čísla:

$$\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!}$$

Tvrzení

$\binom{n}{k} = \binom{n}{n-k}$, $\binom{n}{n} = 1$, $\binom{n}{0} = 1$.

Důkaz: Přímo z definice (s tím, že $0! = 1$).

Tvrzení

Pro $k, n \in \mathbb{N}$, kde $k < n$ platí: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$.

Důkaz: na cvičení.

Binomická věta

Pro reálná čísla a, b a nezáporné celé číslo n je

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} \cdot b^k.$$

Důkaz: Indukcí dle n.

Příklad

$(a-b)^3 = a^3 - 3a^2b + 3ab^2 - b^3$

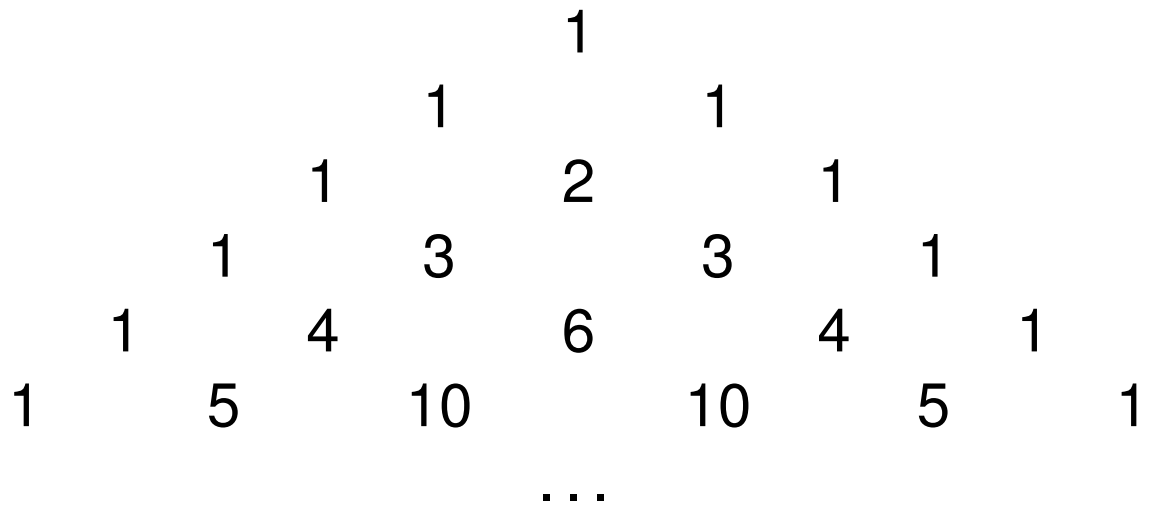
Pascalův trojúhelník

Příslušné binomické koeficienty můžeme velmi rychle najít v tzv. Pascalově trojúhelníku:

$$
\begin{array}{ccccccccccc}
 & & & & \binom{0}{0} & & & & & & \\
 & & & \binom{1}{0} & & \binom{1}{1} & & & & & \\
 & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & & & \\
 & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & & & \\
\binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4} & & \\
\binom{5}{0} & & \binom{5}{1} & & \binom{5}{2} & & \binom{5}{3} & & \binom{5}{4} & & \binom{5}{5} \\
 & & & & \dots & & & & & &
\end{array}
$$

Vzhledem k tomu, že $\binom{n}{0} = \binom{n}{n} = 1$, má trojúhelník po stranách samé jedničky. Z platnosti vztahu $\binom{n}{k} = \binom{n}{n-k}$ musí být trojúhelník souměrný podle jeho svislé osy. A na základě platnosti identity $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ vznikají neokrajová čísla v řadcích jako součet dvou čísel na předchozím řádku.

Pascalův trojúhelník



Příklad

Pomocí binomické věty a s využitím Pascalova trojúhelníka odvoďte vzorec pro $(a+b)^5$.

Řešení: Snadné.

Obsah

Princip inkluze a exkluze

Princip inkluze a exkluze je často používaný kombinatorický princip, který udává počet prvků sjednocení několika množin pomocí počtu prvků průniku jednotlivých množin.

Věta: princip inkluze a exkluze

Pro množiny $A_1, \ldots, A_n$ platí

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \sum_{\emptyset \neq I \subseteq \{1,2,\ldots,n\}} (-1)^{|I|+1} \left|\bigcap_{i \in I} A_i\right|.$$

Princip důkazu na přednášce.

Příklad

V jedné malé základní škole fungují tři kroužky. Florbalový kroužek navštěvuje 18 dětí, výtvarný 14 dětí a šachový je osmičlenný. Z florbalistů jsou dva šachisté a čtyři výtvarníci. Do výtvarného a zároveň do šachového kroužku chodí tři děti. Jedno dítě chodí na všechny tři kroužky. Kolik dětí navštěvuje alespoň jeden ze tří uvedených kroužků?

Řešení: Snadné.

Obsah

Dirichletův princip

Dirichletův (šuplíkový; přihrádkový) princip

Je-li alespoň $r+1$ objektů rozděleno do r šuplíků, pak musí existovat šuplík s nejméně dvěma objekty.

Příklad

Zřejmě žádné zobrazení z množiny A do množiny B nemůže být prosté, jestliže $|A| > |B|$.

Příklad

Mějme čtverec o straně 3 cm a v něm libovolně umístěných 10 bodů. Dokažte, že existují dva body (z těch deseti daných), které jsou od sebe vzdáleny nejvýše $\sqrt{2}$ cm.

Řešení: Snadné.

Dirichletův princip

Dirichletův (zobecněný) princip

Pro přirozená čísla r a m platí: je-li alespoň $mr + 1$ objektů rozděleno do r šuplíků, pak musí existovat šuplík, který má více než m objektů.

Příklad

Dokažte, že v libovolné skupině 97 lidí je určitě alespoň 9 z nich narozeno ve stejný měsíc.

Řešení: Snadné.

Obsah

Co je logika?

Logika je vědou o správném usuzování. V logice jde o formu usuzování, ne o obsah usuzování. Logika má proto symbolický charakter.

Pro uvedené rysy bývá moderní logika označována jako logika formální, popř. symbolická. Je pochopitelné, že symbolický charakter umožňuje logice snadněji odhlédnout od obsahu a soustředit se na formy usuzování.

Logika zkoumá pojmy jako je pravdivost, dokazatelnost, vyvratitelnost a zabývá se jejich vzájemnými vztahy. Logika si klade otázky typu: „Je každé dokazatelné tvrzení pravdivé?“, „Co plyne z toho, že jsme došli nějakou úvahou ke sporu?“.

Formalizace pojmů

Specifikujme nyní jak logika formalizuje pojmy jako je tvrzení a úsudek. Formalizace pojmů a práce s nimi je ve skutečnosti závislá na konkrétním logickém kalkulu (soubor pravidel, která mimo jiné specifikují jak „jemná“ formalizace se používá), se kterým pracujeme. Uvažujme tvrzení:

„Pokud má Petr 20 Kč, pak si může koupit čokoládu.“

V případě, že si nebudeme všímat struktury v jednotlivých větách tohoto souvětí, jde o tvrzení tvaru „Jestliže *A*, pak *B*“. Ve druhé větě se vyskytuje vazba „moci“, z tohoto úhlu pohledu se jedná o tvrzení tvaru „Jestliže *A*, pak může *B*“. Při ještě jemnější formalizaci bychom mohli zachytit i jednotlivé objekty („20 Kč“, „čokoláda“, „Petr“) a vztahy mezi nimi („mít“, „koupit“).

Poznamenejme už nyní, že výroková logika zkoumá usuzování (z pohledu formalizace) na úrovni vět v souvětích; predikátová logika zkoumá usuzování (z pohledu formalizace) až na úrovni jednotlivých větných členů.

Logika klasická a neklasická

Klasickou logikou se rozumí logika, ve které předpokládáme, že tvrzení mohou nabývat dvou pravdivostních hodnot (pravda a nepravda), ve které tvrzení mohou být spojována ve tvrzení složitější spojkami „není pravda, že . . . “, „. . . a . . . “, „. . . nebo . . . “, „jestliže . . . , pak . . . “, „. . . právě když . . . “ a kvantifikátory „pro každé x . . . “ a „existuje x, pro které . . . “ a ve které pravdivostní hodnoty složených tvrzení závisí na pravdivostních hodnotách skládaných tvrzení. Jiná logika se považuje za **neklasickou** (tvrzení mohou nabývat více pravdivostních hodnot nebo je možné používat i jiné spojky nebo mají spojky jiný význam apod.).

Příklady neklasických logik

(a) **modální logika** (logika modalit: možnosti, nutnosti): používá neklasické spojky „je možné, že . . . “, „je nutné, že . . . “

(b) **epistemická logika** (logika znalostí): používá neklasické spojky „ví se, že . . . “, „věří se, že . . . “

(c) **temporální logika** (logika času): zabývá se tvrzeními, ve kterých hraje roli čas.

(d) **fuzzy logika** (logika více pravdivostních hodnot): zabývá se tvrzeními, které mohou mít kromě pravdivostních hodnot pravda a nepravda i jiné hodnoty.

Vztah logiky a informatiky I

Vztah logiky a informatiky je bohatý a různorodý. Se základy logiky by měl být obeznámen každý informatik. Znalost základů logiky nám umožňuje srozumitelně a jednoznačně se vyjadřovat a argumentovat. To je pochopitelně užitečné pro každého, nejen pro informatika. Pro informatika je to však navýsost důležité, protože svoje konstrukce a návrhy musí „sdělit počítači“, například ve formě zdrojového kódu napsaného ve vhodném programovacím jazyce. Zdrojový kód obvykle obsahuje výrazy, které se vyhodnocují podle pravidel logiky (například podmínky v příkazech větvení „if . . . then . . . else . . . “). Logika nás těmto pravidlům učí.

Vztah logiky a informatiky II

Zdrojový kód musí být přesný, jinak je program chybný. Chyby mohou mít dalekosáhlé následky (pomysleme na program pro výpočet mezd, program pro řízení elektrárny apod.). Zdrojový program musí být také srozumitelný, jinak mu nikdo jiný než jeho autor nebude rozumět (a po čase mu nebude rozumět ani jeho autor). Logika nás učí přesnosti i srozumitelnosti.

Pokročilejší partie logiky jsou základem důležitých oblastí informatiky, pro příklad jmenujme logické programování, umělou inteligenci, expertní systémy, analýzu dat.

Vztah logiky a informatiky III

Vztah logiky a informatiky je velmi těsný. Logika je důležitá v informatice (formální metody specifikace, verifikace a analýzy dat) a elektrotechnice (logika el. obvodů). Naopak, výsledky informatických disciplín (teorie informace, teorie jazyků) jsou nepostradatelné v logice. Logika se zabývá například algoritmickými aspekty usuzování a konstrukcí automatických dokazovacích systémů – jde o speciální algoritmy, které jsou schopny, byť omezeně, mechanicky odvozovat tvrzení z jiných.

Logické paradoxy

Formalizace usuzování je podstatná i vzhledem k (logickým) paradoxům:

- **paradox lháře**
  „V tomto okamžiku lžu.“
- **Grellingův paradox**
  Adjektiva dělíme na autologická (mají vlastnost, kterou vyjadřují, například „čtyřslabičný“, „český“) a heterologická (nemají vlastnost, kterou vyjadřují, například „jednoslabičný“, „anglický“). Každé adjektivum patří právě do jedné třídy. Do jaké třídy patří slovo „heterologický“?
- **Russelův paradox**
  Definujme normální množinu jako množinu, která neobsahuje samu sebe (tedy není svým vlastním prvkem). Je množina $\mathscr{M}$ všech normálních množin normální množinou?

Všimněme si, že předchozí paradoxy jsou založeny na tom, že výpověď se vztahuje sama na sebe.

Obsah

VL se zabývá formálním usuzováním o **výrocích**, tedy o tvrzeních, u kterých má smysl uvažovat o jejich pravdivosti. Přirozený jazyk (čeština) se pro formalizaci nehodí – je komplikovaný a nejednoznačný.

Chceme-li zkoumat formy usuzování o výrocích bez ohledu na jejich obsah, bude užitečné označovat výroky pomocí symbolů. Atomické (dále nedělitelné) výroky, například „Sněží.“ budeme označovat spec. symboly, tzv. **výrokovými symboly**. Spojky, kterými se výroky spojují ve složené výroky, budeme označovat **symboly výrokových spojek**. Dovolíme ještě použít **pomocné symboly** – závorky. Tedy například místo výroku „Jestliže sněží a mrzne, pak lze stavět sněhuláky.“ napíšeme $(p \wedge q) \Rightarrow r$.

Jazyk VL

Následuje definice (formálního) jazyka VL.

Definice

Jazyk výrokové logiky se skládá z

- **výrokových symbolů:** $p, q, r, \ldots$, popř. s indexy, $p_1, p_2, \ldots$; předpokládáme, že máme spočetně mnoho výrokových symbolů
- **symbolů výrokových spojek:** $\neg$ (negace), $\Rightarrow$ (implikace)
- **pomocných symbolů:** (,).

Formální jazyk odstraňuje nevýhody přirozeného jazyka.
V jazyku VL například nejsou formulovatelná tvrzení obsahující autoreference (viz paradoxy).

Formule VL

Ze symbolů jazyka sestávají formule VL. (Poznamenejme, že formule jsou přesným zavedením intuitivního pojmu výrok.)

Definice

Nechť je dán jazyk výrokové logiky. **Formule** daného jazyka výrokové logiky je definována následovně

- každý výrokový symbol je formule (tzv. **atomická formule**)
- jsou-li φ a ψ formule, jsou i výrazy $\neg\varphi$, $(\varphi \Rightarrow \psi)$ formule.

Formule jsou tedy jisté konečné posloupnosti symbolů jazyka VL. Například posloupnosti q_3, $\neg\neg\neg p$, $((\neg r \Rightarrow \neg q) \Rightarrow \neg\neg r)$ jsou formule VL, naproti tomu posloupnosti $\neg(p)$, $q \Rightarrow$, $p\neg p$, ((nejsou formule VL.

Víme, že unární booleovská funkce negace a binární booleovská funkce implikace tvoří úplný systém spojek VL (který je bází).

Další spojky VL

Potřeba „umět číst formule“ je nezbytná ve většině informatických a matematických disciplín, ve kterých se formalizované výroky používají k přesnému vyjadřování vztahů v definicích, algoritmech, větách apod.

Nyní zavedeme symboly $\wedge, \vee, \Leftrightarrow$ jako zkratky za jisté posloupnosti symbolů jazyka VL.
Nechť φ a ψ jsou posloupnosti symbolů jazyka VL, pak posloupnosti $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Leftrightarrow \psi)$ jsou zkratky za následující posloupnosti:

$$
\begin{array}{rll}
(\varphi \wedge \psi) & \text{je zkratkou za} & \neg(\varphi \Rightarrow \neg\psi), \\
(\varphi \vee \psi) & \text{je zkratkou za} & (\neg\varphi \Rightarrow \psi), \\
(\varphi \Leftrightarrow \psi) & \text{je zkratkou za} & ((\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)), \\
 & \text{tedy za} & \neg((\varphi \Rightarrow \psi) \Rightarrow \neg(\psi \Rightarrow \varphi)).
\end{array}
$$

Posloupnosti, které jsou zkratkami formulí nejsou samy o sobě formule, pro jednoduchost jim však formule říkat budeme. Tedy řekneme například „formule $(p \wedge \neg q)$“, přestože bychom měli správně říct „formule, jejíž zkratkou je $(p \wedge \neg q)$“.

Konvence o vynechávání vnějších závorek:
Pro zpřehlednění zápisu formulí budeme vynechávat vnější závorky. Budeme psát $(p \Rightarrow q) \Rightarrow r$ místo $((p \Rightarrow q) \Rightarrow r)$, atp.

Někdy se uvažuje následující priorita symbolů výrokových spojek: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$; což umožňuje vynechávat některé závorky. My ji však používat nebudeme.

Důkaz strukturální indukcí pro formule VL

Formule byly definovány tzv. induktivním (nebo rekurzivním) způsobem. Takový způsob nám umožnil konečným způsobem definovat nekonečnou množinu (množina formulí je nekonečná). Navíc můžeme elegantně dokazovat tvrzení tvaru „Každá formule má vlastnost $\mathscr{V}$“. Platí totiž následující:

Věta – důkaz strukturální indukcí pro formule VL

Nechť $\mathscr{V}$ je vlastnost formulí VL. Nechť platí, že

- každý výrokový symbol má vlastnost $\mathscr{V}$
- mají-li formule φ a ψ vlastnost $\mathscr{V}$, pak vlastnost $\mathscr{V}$ mají i formule $\neg\varphi$ a $(\varphi \Rightarrow \psi)$.

Pak vlastnost $\mathscr{V}$ má každá formule VL.

Ukázka použití důkazu strukturální indukcí

Příklad

Chceme dokázat, že počet levých závorek je v každé formuli VL roven počtu pravých závorek. Vlastnost $\mathscr{V}$ je „mít stejný počet levých a pravých závorek“.
Zřejmě každý výrokový symbol má vlastnost $\mathscr{V}$ (neboť každý výrokový symbol má 0 levých a 0 pravých závorek). Mají-li formule φ a ψ vlastnost $\mathscr{V}$, pak vlastnost $\mathscr{V}$ mají i formule $\neg\varphi$ a $(\varphi \Rightarrow \psi)$ (neboť v obou dvou případech přibude stejný počet levých a pravých závorek, konkrétně pro $\neg\varphi$ nula a pro $(\varphi \Rightarrow \psi)$ jedna), což spolu s indukčním předpokladem dokazuje tvrzení.

Příklad

Podobně jednoduše se dá strukturální indukcí dokázat, že nahradíme-li ve formuli VL výrokové symboly formulemi, dostaneme opět formuli VL.

Obsah

Sémantika VL

Zatím jsme se věnovali jen tzv. syntaktické stránce výrokové logiky. Řekli jsme si, co je to jazyk VL a co jsou formule VL. Zatím však nevíme, co to je pravdivá formule apod. Formule jsou jisté posloupnosti symbolů jazyka, samy o sobě však nemají žádný význam. Přiřazení významu syntaktickým objektům je záležitostí tzv. **sémantiky**. Právě sémantice výrokové logiky se budeme nyní věnovat.

(Pravdivostní) ohodnocení

Definice

(Pravdivostní) ohodnocení je libovolné zobrazení e výrokových symbolů daného jazyka výrokové logiky do množiny $\{0,1\}$, tedy ohodnocení e přiřazuje každému výrokovému symbolu p hodnotu 0 nebo 1.

0 a 1 reprezentují pravdivostní hodnoty nepravda a pravda. Hodnotu přiřazenou ohodnocením e symbolu p označujeme $e(p)$. Je tedy $e(p) = 0$ nebo $e(p) = 1$. Je-li dáno ohodnocení e, můžeme říci, co je to pravdivostní hodnota formule.

Pravdivostní hodnota formule VL

Pravdivostní hodnota libovolné formule je pravdivostním ohodnocením jednoznačně určena a je definována takto:

Definice

Nechť je dáno ohodnocení e. **Pravdivostní hodnota formule φ při ohodnocení e**, označujeme ji $\| \varphi \|_e$, je definována následovně:

- Je-li φ výrokovým symbolem p, pak $\| p \|_e = e(p)$.
- Je-li φ složená formule, tedy má tvar $\neg\psi$ nebo $\psi \Rightarrow \theta$, pak
  $\| \neg\psi \|_e = 1$, pokud $\| \psi \|_e = 0$;
  $\| \neg\psi \|_e = 0$, pokud $\| \psi \|_e = 1$ a

  $\| \psi \Rightarrow \theta \|_e = 1$, pokud $\| \psi \|_e = 0$ nebo $\| \theta \|_e = 1$;
  $\| \psi \Rightarrow \theta \|_e = 0$ jinak.

Je-li $\| \varphi \|_e = 1$ ($\| \varphi \|_e = 0$), říkáme, že **formule φ je při ohodnocení e pravdivá** (**nepravdivá**).

Poznámka: Uvědomme si, že nemá smysl říci „formule φ je pravdivá“ nebo „nepravdivá“ (musíme říci při jakém ohodnocení!). Neboli pravdivost formule chápeme vždy vzhledem k nějakému ohodnocení.

Poznámka: Alternativně lze zadat pravdivostní funkci (operaci) logických spojek tabulkami:

a	$\rightharpoonup a$
0	1
1	0

$\rightarrow$	0	1
0	1	1
1	0	1

Pak pravdivostní hodnotu složených formulí $\neg\psi$, $\psi \Rightarrow \theta$, při ohodnocení e, lze definovat takto:

$$\| \neg\psi \|_e = \rightharpoonup \| \psi \|_e;$$
$$\| \psi \Rightarrow \theta \|_e = \| \psi \|_e \rightarrow \| \theta \|_e.$$

Tautologie, splnitelné formule a kontradikce

Definice

Formule VL se nazývá

- **tautologie**, je-li při každém ohodnocení pravdivá,
- **kontradikce**, je-li při každém ohodnocení nepravdivá,
- **splnitelná**, je-li pravdivá při alespoň jednom ohodnocení.

Zřejmě splnitelné formule jsou právě ty, které nejsou kontradikcemi. Fakt, že formule φ je tautologie, zapisujeme $\models \varphi$, popřípadě $\| \varphi \| = 1$.

Označení: Je-li φ formule VL, pak píšeme $\varphi(p_1, \ldots, p_n)$, chceme-li zdůraznit, že všechny výrokové symboly vyskytující se ve φ jsou mezi $p_1, \ldots, p_n$ (tedy žádný jiný výrokový symbol než některý z $p_1, \ldots, p_n$ se ve φ nevyskytuje).

Lemma

Platí-li pro ohodnocení e a e', že $e(p_1) = e'(p_1), \ldots, e(p_n) = e'(p_n)$, pak pro každou formuli $\varphi(p_1, \ldots, p_n)$ platí $\| \varphi \|_e = \| \varphi \|_{e'}$.

Důkaz: Jednoduchý – strukturální indukcí pro formule VL. Vlastnost $\mathscr{V}$ je tvrzení Lemmy.

Jinak řečeno, pravdivostní hodnota formule VL závisí jen na tom, jaké hodnoty přiřazuje dané ohodnocení výrokovým symbolům, které se ve formuli vyskytují.

Tabulková metoda

Pro n výrokových symbolů $p_1, \ldots, p_n$ existuje právě 2^n různých ohodnocení symbolů $p_1, \ldots, p_n$ (každému výrokovému symbolu se přiřazuje 0 nebo 1). Tyto úvahy jsou základem tzv. tabulkové metody pro zjištění pravdivostních hodnot formule.

Tabulková metoda slouží k vypsání (tabelaci) hodnot zadaných formulí $\varphi_1, \varphi_2, \ldots, \varphi_m$ v tabulce. Tabulka má (pod záhlavím) 2^n řádků a $n+m$ sloupců, kde n je počet všech výrokových symbolů, které se vyskytují ve formulích $\varphi_1, \varphi_2, \ldots, \varphi_m$. Do řádků píšeme všechna možná ohodnocení těchto symbolů a hodnoty formulí $\varphi_1, \varphi_2, \ldots, \varphi_m$.
Formule φ_i je tautologií (kontradikcí, splnitelnou), právě když jí odpovídající sloupec pravdivostních hodnot obsahuje ve všech řádcích samé 1 (samé 0, aspoň jednu 1).

Poznamenejme, že s využitím tabulkové metody lze formule algoritmicky převádět do tzv. úplné konjunktivní (respektive disjunktivní) normální formy.

Sémantické vyplývání

Dále si definujeme pojem sémantického vyplývání, který formalizuje intuitivní pojem „vyplývání“ z množin formulí.

Definice

Formule ψ **sémanticky plyne z formule** φ, značíme $\varphi \models \psi$, jestliže ψ je pravdivá při každém ohodnocení, při kterém je pravdivá φ. Pokud ψ sémanticky plyne z φ a naopak, říkáme, že φ a ψ jsou **sémanticky ekvivalentní**.
Obecněji, formule ψ **sémanticky plyne z množiny formulí** T, značíme $T \models \psi$, je-li ψ pravdivá při každém ohodnocení, při kterém je pravdivá každá formule z T.

Pro ověření sémantického vyplývání je možné použít tabelaci (tabulkovou metodu).

Příklad na sémantické vyplývání

Zjistěte zda z množiny $T = \{p \Rightarrow \neg q, q, \neg(((p \Rightarrow \neg q) \vee r) \Leftrightarrow (r \wedge \neg q))\}$ sémanticky plynou následující tři formule: $r \wedge \neg q$; r; $(p \Rightarrow \neg q) \vee r$.

K řešení použijeme tabulkovou metodu pomocí které zjistíme, při kterých ohodnoceních nabývají formule z T současně pravdivostní hodnotu 1 (viz šedě podbarvené řádky). Nyní se stačí podívat, zda při těchto (dvou) ohodnoceních jsou jednotlivé formule ze zadání (modré) také pravdivé (v obou případech). Pokud ano, pak sémanticky vyplývají z T. Jinak sémanticky nevyplývají z T.

p	q	r	$p \Rightarrow \neg q$	$(p \Rightarrow \neg q) \vee r$	$r \wedge \neg q$	$\neg(((p \Rightarrow \neg q) \vee r) \Leftrightarrow (r \wedge \neg q))$
1	1	1	0	1	0	1
1	1	0	0	0	0	0
1	0	1	1	1	1	0
1	0	0	1	1	0	1
0	1	1	1	1	0	1
0	1	0	1	1	0	1
0	0	1	1	1	1	0
0	0	0	1	1	0	1

Zřejmě tedy $p \Rightarrow \neg q, q, \neg(((p \Rightarrow \neg q) \vee r) \Leftrightarrow (r \wedge \neg q)) \models (p \Rightarrow \neg q) \vee r$.

Poznámka: Zřejmě formule φ, ψ jsou sémanticky ekvivalentní, pokud $\| \varphi \|_e = \| \psi \|_e$ pro každé ohodnocení e.

Poznámka: Formule φ, ψ jsou sémanticky ekvivalentní, právě když je formule $\varphi \Leftrightarrow \psi$ tautologie. Tedy sémanticky ekvivalentní formule od sebe nelze rozlišit pravdivostí.

Některé tautologie povyšujeme na tzv. **zákony VL**.

Některé zákony VL, kde φ, ψ, χ jsou libovolné formule VL:

1. $\varphi \vee \neg\varphi$ (zákon vyloučeného třetího)
2. $\neg(\varphi \wedge \neg\varphi)$ (zákon sporu)
3. $\neg\neg\varphi \Leftrightarrow \varphi$ (zákon dvojí negace)
4. $(\varphi \wedge \psi) \Leftrightarrow (\psi \wedge \varphi)$ (komutativní zákon pro $\wedge$)
5. $(\varphi \vee \psi) \Leftrightarrow (\psi \vee \varphi)$ (komutativní zákon pro $\vee$)
6. $(\varphi \wedge (\psi \wedge \chi)) \Leftrightarrow ((\varphi \wedge \psi) \wedge \chi)$ (asociativní zákon pro $\wedge$)
7. $(\varphi \vee (\psi \vee \chi)) \Leftrightarrow ((\varphi \vee \psi) \vee \chi)$ (asociativní zákon pro $\vee$)
8. $(\varphi \wedge (\psi \vee \chi)) \Leftrightarrow ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$ (distributivní zákon)
9. $(\varphi \vee (\psi \wedge \chi)) \Leftrightarrow ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$ (distributivní zákon)
10. $\neg(\varphi \wedge \psi) \Leftrightarrow (\neg\varphi \vee \neg\psi)$ (de Morganův zákon)
11. $\neg(\varphi \vee \psi) \Leftrightarrow (\neg\varphi \wedge \neg\psi)$ (de Morganův zákon)
12. $(\varphi \Rightarrow \psi) \Leftrightarrow (\neg\varphi \vee \psi)$ (náhrada implikace)
13. $\neg(\varphi \Rightarrow \psi) \Leftrightarrow (\varphi \wedge \neg\psi)$ (náhrada negace implikace)
14. $(\varphi \Rightarrow \psi) \Leftrightarrow (\neg\psi \Rightarrow \neg\varphi)$ (zákon kontrapozice)
15. $(\varphi \Leftrightarrow \psi) \Leftrightarrow ((\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi))$ (náhrada ekvivalence)
16. $((\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \chi)) \Rightarrow (\varphi \Rightarrow \chi)$ (tranzitivita implikace).

Je užitečné si uvědomit ještě další tautologie:

a) $(\varphi \wedge \varphi) \Leftrightarrow \varphi$, $(\varphi \vee \varphi) \Leftrightarrow \varphi$ (idempotentnost $\vee, \wedge$)

b) $\varphi \Rightarrow (\psi \Rightarrow \varphi)$

c) $\varphi \Rightarrow (\psi \vee \varphi)$

d) $(\varphi \wedge \psi) \Rightarrow \varphi$.

I zde jsou φ a ψ libovolné formule výrokové logiky.

Znovu sémantické vyplývání

Už víme, že VL má svou **syntaxi** a **sémantiku**. Syntaxe VL definuje pojmy jako je jazyk a formule, ale formulemi (i ostatními syntaktickými pojmy) se zabývá čistě z pohledu jejich tvaru. Sémantika VL zavádí pojem pravd. ohodnocení a pravdivost formule při daném ohodnocení. Sémantika přiřazuje význam syntaktickým pojmům.

Pojem vyplývání má v logice ústřední význam (zopakujme jej):

Definice

Mějme formule $\psi_1, \ldots, \psi_n$ ($n \geq 0$). Formule φ **sémanticky plyne** z formulí $\psi_1, \ldots, \psi_n$ (značíme $\psi_1, \ldots, \psi_n \models \varphi$), jestliže $\| \varphi \|_e = 1$ pro každé ohodnocení e takové, že $\| \psi_1 \|_e = 1, \ldots, \| \psi_n \|_e = 1$.

Formule $\psi_1, \ldots, \psi_n$ nazýváme **předpoklady**, formuli φ **sémantický důsledek** formulí $\psi_1, \ldots, \psi_n$.

Příklad

Dokažte, že je-li $\psi \models \varphi$ a $\varphi \models \chi$, pak $\psi \models \chi$.

Řešení: Máme ukázat, že $\psi \models \chi$. Nechť e je ohodnocení, při kterém je ψ pravdivá. Dle předpokladu $\psi \models \varphi$ je při e pravdivá také φ a tedy dle předpokladu $\varphi \models \chi$ je při e pravdivá také χ, což jsme měli ukázat.

Sémantická podoba věty o dedukci I

Věta

Nechť $\chi_1, \ldots, \chi_n, \varphi, \psi$ jsou formule VL. Pak platí:
$\chi_1, \ldots, \chi_n \models \varphi \Rightarrow \psi$, právě když $\chi_1, \ldots, \chi_n, \varphi \models \psi$.

Důkaz:

$(\Rightarrow)$

Nejprve předpokládejme $\chi_1, \ldots, \chi_n \models \varphi \Rightarrow \psi$ a dokažme $\chi_1, \ldots, \chi_n, \varphi \models \psi$. Stačí ověřit, že pro každé ohodnocení e, při kterém jsou všechny formule z $\chi_1, \ldots, \chi_n, \varphi$ pravdivé, máme $\| \psi \|_e = 1$. Jsou-li ale $\chi_1, \ldots, \chi_n, \varphi$ při ohodnocení e pravdivé, pak dostáváme $\| \varphi \Rightarrow \psi \|_e = 1$ dle předpokladu. Rovněž platí $\| \varphi \|_e = 1$. To jest $\| \varphi \Rightarrow \psi \|_e = \| \varphi \|_e \rightarrow \| \psi \|_e = 1 \rightarrow \| \psi \|_e = 1$. Z vlastností $\rightarrow$ pak plyne, že $\| \psi \|_e = 1$. To jest $\chi_1, \ldots, \chi_n, \varphi \models \psi$.

$(\Leftarrow)$ následující slajd

Sémantická podoba věty o dedukci II

Věta

Nechť $\chi_1, \ldots, \chi_n, \varphi, \psi$ jsou formule VL. Pak platí:
$\chi_1, \ldots, \chi_n \models \varphi \Rightarrow \psi$, právě když $\chi_1, \ldots, \chi_n, \varphi \models \psi$.

Důkaz:

($\Rightarrow$) předchozí slajd

($\Leftarrow$)

Naopak předpokládejme $\chi_1, \ldots, \chi_n, \varphi \models \psi$. Stačí ověřit, že pro každé ohodnocení e, při kterém jsou všechny formule $\chi_1, \ldots \chi_n$ pravdivé, je $\| \varphi \Rightarrow \psi \|_e = 1$. Mohou nastat dva případy:

1) $\| \varphi \|_e = 0$, odkud $\| \varphi \Rightarrow \psi \|_e = 0 \rightarrow \| \psi \|_e = 1$.
2) $\| \varphi \|_e = 1$, to jest při ohodnocení e jsou pravdivé všechny formule z $\chi_1, \ldots, \chi_n, \varphi$ a tedy $\| \psi \|_e = 1$ dle předpokladu. Odtud $\| \varphi \Rightarrow \psi \|_e = 1 \rightarrow 1 = 1$, v důsledku čehož $\chi_1, \ldots, \chi_n \models \varphi \Rightarrow \psi$.

Aplikace sémantické podoby věty o dedukci

Příklady aplikace sémantické podoby věty o dedukci:

- Můžeme okamžitě tvrdit, že $\models \varphi \Rightarrow \varphi$, protože φ sémanticky plyne z φ triviálně.
- Dvojnásobnou aplikací VoD na zřejmý fakt $\varphi, \psi \models \varphi \wedge \psi$ dostáváme $\models \varphi \Rightarrow (\psi \Rightarrow (\varphi \wedge \psi))$.
- Dále, k tomu abychom ověřili, že φ sémanticky plyne z formulí $\chi_1, \ldots, \chi_n$ stačí ověřit, že formule $\chi_1 \Rightarrow (\chi_2 \Rightarrow (\ldots(\chi_n \Rightarrow \varphi)\ldots))$ je tautologie.

Obsah

Motivace: Tabelace je neúnosná při velkém množství výrokových symbolů. Nabízí se tedy otázka, zda-li není možné o sémantickém vyplývání rozhodnout jinak než tabelací . . .

Odvozovací pravidlo modus ponens

Nejprve si zavedeme nový pojem vyplývání, který nebude založen na pojmu pravdivostní ohodnocení, ale pouze na manipulaci s formulemi na úrovni jejich tvaru. Základní pojem, na kterém je tento typ vyplývání založen je **odvozovací pravidlo** – předpis pomocí nějž ze vstupních formulí odvozujeme další formule. Odvozovací pravidla formalizují elementární úsudky. Nám bude ve VL postačovat pouze jediné odvozovací pravidlo, tzv. **pravidlo odloučení** neboli **modus ponens** (MP), které lze schématicky vyjádřit

$$\text{MP:} \qquad \frac{\varphi, \varphi \Rightarrow \psi}{\psi}$$

a jehož význam je: „z formulí φ a $\varphi \Rightarrow \psi$ odvodíme formuli ψ“. Formulím $\varphi, \varphi \Rightarrow \psi$ někdy říkáme **předpoklady**.

Například formule $\neg q$ vzniká použitím modus ponens z formulí $p \Rightarrow r$ a $(p \Rightarrow r) \Rightarrow \neg q$.

Axiomy VL

Při odvozování formulí budeme dále používat **axiomy**, což jsou formule, které automaticky přijímáme jako „platné“. Axiomy popisují vlastnosti logických spojek a jejich vzájemný vztah. Axiomy VL si definujeme pomocí tří **axiomových schémat**:

(A1) $\varphi \Rightarrow (\psi \Rightarrow \varphi)$,

(A2) $(\varphi \Rightarrow (\psi \Rightarrow \chi)) \Rightarrow ((\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \chi))$,

(A3) $(\neg\psi \Rightarrow \neg\varphi) \Rightarrow (\varphi \Rightarrow \psi)$.

Jakákoli formule, která je ve tvaru jednoho ze schémat (A1) – (A3) se nazývá **axiom** VL.

Axiomová schémata jsou „předpisy“, kterými definujeme všechny axiomy. Ačkoli budeme používat pouze tři axiomová schémata, axiomů jako takových je nekonečně mnoho. Například formule $(\neg(p \Rightarrow q) \Rightarrow \neg\neg p) \Rightarrow (\neg p \Rightarrow (p \Rightarrow q))$ je axiom, který je instancí schéma (A3). Dále například $p \Rightarrow (q \Rightarrow r)$ není axiom.

Množinu axiomů a odvozovacích pravidel, která používáme, souhrnně nazýváme **axiomatický systém**.

Důkaz

Pod pojmem „důkaz“ je intuitivně myšlen záznam odvozování, provedený tak, že za sebe napíšeme tvrzení, ke kterým se postupně dobíráme tak, že začneme předpoklady a pokračujeme tvrzeními, která z předchozích tvrzení plynou pomocí elementárních úsudkových kroků.

Nyní zavedeme přesný pojem důkazu v našem axiomatickém systému – neformální pojem důkazu tak převedeme z úrovně intuice na přesnou formální úroveň.

Důkaz, syntaktické vyplývání

Definice

Důkaz formule φ z množiny formulí T je lib. posloupnost formulí $\varphi_1, \ldots, \varphi_n$ taková, že $\varphi_n = \varphi$ a každá φ_i $(i = 1, \ldots, n)$

- je axiomem,
- nebo náleží do T,
- nebo vzniká z předchozích formulí důkazu pomocí odvozovacího pravidla MP, tedy existují indexy $j, k < i$ tak, že φ_k je formule ve tvaru $\varphi_j \Rightarrow \varphi_i$.

Formule φ je dokazatelná z T (zapisujeme $T \vdash \varphi$), pokud existuje důkaz formule φ z T. Pokud $\vdash \varphi$, pak říkáme, že φ je dokazatelná (z prázdného systému předpokladů).

Dokazatelnosti budeme také říkat **syntaktické vyplývání**, abychom tím zdůraznili, že jde o protějšek sémantického vyplývání. Fakt $T \vdash \varphi$ lze tedy číst „φ syntakticky plyne z T“, případně „φ je syntaktickým důsledkem T“.

Zřejmě každý axiom je dokazatelný, neboť $\vdash \varphi$ platí pro každý axiom φ, protože jednoprvková posloupnost φ je důkazem φ z prázdného systému předpokladů.

Poznámka: Máme dva pojmy vyplývání formule z množiny formulí: sémantické vyplývání ($T \models \varphi$) a syntaktické vyplývání ($T \vdash \varphi$). Jak spolu souvisí uvidíme později (ve větě o korektnosti a větě o úplnosti).
Speciálně máme dva pojmy platnosti formule: $\models \varphi$ označuje platnost φ v sémantickém smyslu (pravdivost), $\vdash \varphi$ označuje platnost φ v syntaktickém smyslu (dokazatelnost).

Tvrzení

Pro každou množinu formulí T a formule φ, ψ platí, že z $T \vdash \varphi \Rightarrow \psi$ a $T \vdash \varphi$ plyne $T \vdash \psi$.

Důkaz: Máme tedy dokázat, že jsou-li z T dokazatelné formule $\varphi \Rightarrow \psi$ a φ, pak je z T dokazatelná i formule ψ. Jsou-li však z T dokazatelné formule $\varphi \Rightarrow \psi$ a φ, znamená to, že existuje důkaz $\chi_1, \ldots, \chi_n$ formule $\varphi \Rightarrow \psi$ z T (tedy χ_n je formulí $\varphi \Rightarrow \psi$) a že existuje důkaz $\theta_1, \ldots, \theta_m$ formule φ z T (tedy θ_m je formulí φ). Nyní však stačí vzít posloupnost $\chi_1, \ldots, \chi_n, \theta_1, \ldots, \theta_m, \psi$ – ta je již důkazem ψ z T. Abychom se o tom přesvědčili, stačí ověřit podmínky z definice pojmu důkaz (pro každou formuli uvažované posloupnosti). Zřejmě každá formule χ_i je buď axiomem nebo je formulí z T nebo plyne z nějakých předchozích χ_k, χ_l pomocí MP. Podobně uvažujeme pro libovolnou formuli θ_j. Dále, formule ψ plyne z formulí χ_n (což je $\varphi \Rightarrow \psi$) a θ_m (což je φ) pomocí MP. Vidíme tedy, že posloupnost $\chi_1, \ldots, \chi_n, \theta_1, \ldots, \theta_m, \psi$ je důkazem ψ z T, tedy $T \vdash \psi$.

Věta

Pro každou formuli φ platí $\vdash \varphi \Rightarrow \varphi$ (tedy formule $\varphi \Rightarrow \varphi$ je dokazatelná v našem axiomatickém systému).

Důkaz: Máme ukázat, že existuje důkaz (z prázdné množiny předpokladů), jehož posledním prvkem je $\varphi \Rightarrow \varphi$. Důkazem formule $\varphi \Rightarrow \varphi$ je například posloupnost formulí

$\alpha_1 : \ \varphi \Rightarrow ((\varphi \Rightarrow \varphi) \Rightarrow \varphi)$

$\alpha_2 : \ (\varphi \Rightarrow ((\varphi \Rightarrow \varphi) \Rightarrow \varphi)) \Rightarrow ((\varphi \Rightarrow (\varphi \Rightarrow \varphi)) \Rightarrow (\varphi \Rightarrow \varphi))$

$\alpha_3 : \ (\varphi \Rightarrow (\varphi \Rightarrow \varphi)) \Rightarrow (\varphi \Rightarrow \varphi)$

$\alpha_4 : \ \varphi \Rightarrow (\varphi \Rightarrow \varphi)$

$\alpha_5 : \ \varphi \Rightarrow \varphi$

Důkazem $\vdash \varphi \Rightarrow \varphi$ by byla i posloupnost: $\alpha_4, \alpha_2, \alpha_1, \alpha_3, \alpha_5$.

Fakt, že $\vdash \varphi \Rightarrow \varphi$ budeme dále používat.

Lemma – monotonie dokazatelnosti (MD)

Nechť T a S jsou množiny formulí a φ, ψ jsou formule. Pak platí: pokud $T \vdash \varphi$ a pro každou $\psi \in T$ máme $S \vdash \psi$, pak $S \vdash \varphi$.

Důkaz: Předpokládejme, že platí $T \vdash \varphi$. To jest existuje důkaz $\chi_1, \ldots, \chi_n$ z T, kde $\chi_n = \varphi$. Uvažujme posloupnost $\vartheta_1, \ldots \vartheta_m$, kterou vytvoříme z posloupnosti $\chi_1, \ldots, \chi_n$ tak, že každý člen χ_i, pro který máme $\chi_i \in T$, nahradíme některým jeho důkazem ze systému S (důkaz vždy existuje, jelikož $S \vdash \chi_i$), jinými slovy, formuli χ_i „vyjmeme“ z posloupnosti $\chi_1, \ldots, \chi_n$ a na její místo „vložíme důkaz“ formule χ_i z S, což je opět konečná posloupnost formulí. Vzniklá posloupnost $\vartheta_1, \ldots \vartheta_m$ je evidentně důkazem z S a ϑ_m je formule φ. Dostáváme tedy $S \vdash \varphi$.

Poznámka: MD budeme často používat v následující situaci. Z platnosti $\vdash \varphi$ odvodíme jednoduše platnost $T \vdash \varphi$.

Věta o dedukci

Věta o dedukci (VoD)

Pro každou množinu formulí T a formule φ, ψ platí: $T \vdash \varphi \Rightarrow \psi$, právě když $T, \varphi \vdash \psi$.

Důkaz:

„$\Rightarrow$“ Předpokládáme-li $T \vdash \varphi \Rightarrow \psi$, je tím spíše (dle MD) $T, \varphi \vdash \varphi \Rightarrow \psi$. Použitím MP okamžitě dostáváme $T, \varphi \vdash \psi$.

„$\Leftarrow$“ Nechť $T, \varphi \vdash \psi$, tedy existuje důkaz $\psi_1, \ldots, \psi_n$ formule ψ z T, φ (ψ_n je ψ). Indukcí dokážeme, že $T \vdash \varphi \Rightarrow \psi_i$ platí pro $i = 1, \ldots, n$, z čehož dostaneme požadovaný vztah jako speciální případ pro $i = n$. Vezměmě tedy $i \in \{1, \ldots, n\}$ a předpokládejme, že pro každé $j < i$ platí $T \vdash \varphi \Rightarrow \psi_j$ (indukční předpoklad). Dokážeme, že $T \vdash \varphi \Rightarrow \psi_i$. Podle definice důkazu mohou nastat pouze následující tři případy:

(A) ψ_i je axiom nebo formule z T. Pak je posloupnost formulí
$\psi_i \Rightarrow (\varphi \Rightarrow \psi_i)$,
ψ_i,
$\varphi \Rightarrow \psi_i$
důkazem formule $\varphi \Rightarrow \psi_i$ z T.

(B) ψ_i je formulí φ. Pak $T \vdash \varphi \Rightarrow \psi_i$ plyne z předchozí Věty.

(C) ψ_i plyne z předchozích formulí $\psi_j, \psi_k = \psi_j \Rightarrow \psi_i$ $(j,k < i)$ pomocí MP. Dle indukčního předpokladu existuje důkaz $\alpha, \ldots, \varphi \Rightarrow \psi_j$ z T a důkaz $\beta, \ldots, \varphi \Rightarrow (\psi_j \Rightarrow \psi_i)$ z T.
Přidáme-li k posloupnosti
$\alpha, \ldots, \varphi \Rightarrow \psi_j, \beta, \ldots, \varphi \Rightarrow (\psi_j \Rightarrow \psi_i)$ formule
$(\varphi \Rightarrow (\psi_j \Rightarrow \psi_i)) \Rightarrow ((\varphi \Rightarrow \psi_j) \Rightarrow (\varphi \Rightarrow \psi_i))$,
$(\varphi \Rightarrow \psi_j) \Rightarrow (\varphi \Rightarrow \psi_i)$,
$\varphi \Rightarrow \psi_i$,
dostaneme důkaz formule $\varphi \Rightarrow \psi_i$ z T.

Důkaz je hotov.

Věta o dedukci umožňuje mimo jiné zkracovat důkazy.

Příklad

Ukažme, že jestliže $T \vdash \varphi \Rightarrow \psi$ a $T \vdash \psi \Rightarrow \chi$, pak $T \vdash \varphi \Rightarrow \chi$ (tzv. **princip tranzitivity implikace**). Skutečně, máme $T, \varphi \vdash \psi$ (dle VoD aplikované na $T \vdash \varphi \Rightarrow \psi$), dále $T, \varphi \vdash \chi$ (použitím MP a MD) a konečně $T \vdash \varphi \Rightarrow \chi$ (VoD použitá na $T, \varphi \vdash \chi$).

Věta

Pro formule φ, ψ platí

($a_{\vdash}$) $\vdash \neg\varphi \Rightarrow (\varphi \Rightarrow \psi)$,

($b_{\vdash}$) $\vdash \neg\neg\varphi \Rightarrow \varphi$,

($c_{\vdash}$) $\vdash \varphi \Rightarrow \neg\neg\varphi$,

($d_{\vdash}$) $\vdash (\varphi \Rightarrow \psi) \Rightarrow (\neg\psi \Rightarrow \neg\varphi)$,

($e_{\vdash}$) $\vdash \varphi \Rightarrow (\neg\psi \Rightarrow \neg(\varphi \Rightarrow \psi))$.

Důkaz: na cvičení.

Poznámka: Vztahy ($a_{\vdash}$) – ($e_{\vdash}$) mají dobrý intuitivní význam. Vztah ($a_{\vdash}$) vyjadřuje, že pokud je φ neplatná, pak z vlastnosti φ plyne lib. formule. Vztahy ($b_{\vdash}$) a ($c_{\vdash}$) popisují vlastnosti dvojí negace – popisují právě to, co na sémantické úrovni vyjadřuje fakt, že φ a $\neg\neg\varphi$ jsou sémanticky ekvivalentní. Vztah ($d_{\vdash}$) je duálním vztahem k axiomovému schématu (A3) a spolu s (A3) popisuje to, co na sémantické úrovni vyjadřuje fakt, že $\varphi \Rightarrow \psi$ a $\neg\psi \Rightarrow \neg\varphi$ jsou sémanticky ekvivalentní. Vztah ($e_{\vdash}$) je modifikací vztahu: „z platnosti φ a z platnosti ψ plyne platnost $\varphi \wedge \psi$“.

Spornost a bezespornost

Definice

Množina formulí T se nazývá **sporná** (nekonzistentní), jestliže je z ní dokazatelná jakákoliv formule. Není-li T sporná (tedy existuje formule, která není z T dokazatelná), nazývá se **bezesporná** (konzistentní).

Lemma

Následující tvrzení jsou ekvivalentní:

(i) Množina formulí T je sporná;

(ii) $T \vdash \varphi$ a $T \vdash \neg\varphi$ pro nějakou formuli φ;

(iii) $T \vdash \neg(\vartheta \Rightarrow \vartheta)$.

Důkaz: „(i) $\Rightarrow$ (ii)“: Pokud je T sporný systém formulí, pak je z něj dokazatelná jakákoliv formule, tedy i formule φ a $\neg\varphi$.

„(ii) $\Rightarrow$ (iii)“: Nechť $T \vdash \varphi$ a $T \vdash \neg\varphi$. Dle $(a_{\vdash})$ máme $\vdash \neg\varphi \Rightarrow (\varphi \Rightarrow \neg(\vartheta \Rightarrow \vartheta))$, z MD $T \vdash \neg\varphi \Rightarrow (\varphi \Rightarrow \neg(\vartheta \Rightarrow \vartheta))$. Dvojnásobným použitím MP dostaneme $T \vdash \neg(\vartheta \Rightarrow \vartheta)$.

„(iii) $\Rightarrow$ (i)“: Nechť φ je libovolná formule. Platí $\vdash \neg(\vartheta \Rightarrow \vartheta) \Rightarrow ((\vartheta \Rightarrow \vartheta) \Rightarrow \varphi)$ opět dle $(a_{\vdash})$. Z MD $T \vdash \neg(\vartheta \Rightarrow \vartheta) \Rightarrow ((\vartheta \Rightarrow \vartheta) \Rightarrow \varphi)$. Dále platí, že $T \vdash \vartheta \Rightarrow \vartheta$; z předpokladu $T \vdash \neg(\vartheta \Rightarrow \vartheta)$ tedy dvojnásobným použitím MP máme $T \vdash \varphi$.

O důkazu sporem

Důkaz sporem je populární dokazovací princip v informatice a matematice. Sporem se snadno dokazuje například tvrzení: „prvočísel je nekonečně mnoho“ nebo „$\sqrt{2} \notin \mathbb{Q}$“ atd. Při dokazování postupujeme tak, že předpokládáme neplatnost tvrzení a dojdeme ke sporu, čímž dokážeme platnost daného tvrzení.
Následující věta ukazuje, že intuitivní důkaz sporem má ve VL svou formalizaci.

Důkaz sporem

Věta o důkazu sporem

Nechť T je množina formulí, nechť φ je libovolná formule. Pak platí: $T \vdash \varphi$, právě když $T, \neg\varphi$ je sporná množina.

Důkaz: Nechť $T \vdash \varphi$. Pak zřejmě $T, \neg\varphi \vdash \varphi$ a triviálně též $T, \neg\varphi \vdash \neg\varphi$, což dle (ii) předchozí Lemmy znamená, že $T, \neg\varphi$ je sporná množina.
Naopak, předpokládáme-li, že $T, \neg\varphi$ je sporná množina, pak je z $T, \neg\varphi$ dokazatelná formule $\neg(\vartheta \Rightarrow \vartheta)$ dle (iii) předchozí Lemmy. Užitím VoD máme $T \vdash \neg\varphi \Rightarrow \neg(\vartheta \Rightarrow \vartheta)$. Jelikož $(\neg\varphi \Rightarrow \neg(\vartheta \Rightarrow \vartheta)) \Rightarrow ((\vartheta \Rightarrow \vartheta) \Rightarrow \varphi)$ je axiom dle (A3), pak z MD a užitím MP dostáváme $T \vdash (\vartheta \Rightarrow \vartheta) \Rightarrow \varphi$. Dále $\vdash \vartheta \Rightarrow \vartheta$, odkud opětovným užitím MD a MP máme $T \vdash \varphi$.

Věta o neutrální formuli (VoNF)

Věta o důkazu rozborem případů

Pro množinu formulí T a formule φ, ψ, χ platí $T, \varphi \vee \psi \vdash \chi$, právě když $T, \varphi \vdash \chi$ a $T, \psi \vdash \chi$.

Důkaz: Vynecháme.

Věta o neutrální formuli (VoNF)

Pro množinu formulí T a formule φ a ψ platí $T \vdash \psi$, právě když $T, \varphi \vdash \psi$ a $T, \neg\varphi \vdash \psi$.

Důkaz: Dle předchozí věty je $T, \varphi \vee \neg\varphi \vdash \psi$, právě když $T, \varphi \vdash \psi$ a $T, \neg\varphi \vdash \psi$. Dále však platí, že $T, \varphi \vee \neg\varphi \vdash \psi$, právě když $T \vdash \psi$ (neboť $\varphi \vee \neg\varphi$ je zkratkou za $\neg\varphi \Rightarrow \neg\varphi$, což (jak víme) je dokazatelná formule; pro dokazatelnou formuli α je vždy $T, \alpha \vdash \beta$, právě když $T \vdash \beta$), a tím je důkaz hotov.

Viděli jsme formule, které jsou v našem axiomatickém systému dokazatelné. Brzy se lehce přesvědčíme, že každá dokazatelná formule je tautologií.
Nabízí se otázka, zda také naopak je každá tautologie dokazatelná. Uvidíme, že ano (a uvidíme i více). Jinými slovy, naše axiomy a odvozovací pravidlo jsou zvoleny tak vhodně, že umožňují dokázat všechny tautologie, ale žádné další formule (tedy formule, které jsou někdy nepravdivé).

Poznámka: Pokud bychom označili Fml množinu všech formulí jazyka VL, ve kterém pracujeme, pak potenční množina 2^{Fml} je vlastně množinou všech systémů formulí ($T \in 2^{\mathrm{Fml}}$ potom znamená, že T je systém formulí). Syntaktické vyplývání je tedy relace $\vdash \subseteq 2^{\mathrm{Fml}} \times \mathrm{Fml}$, přitom $T \in 2^{\mathrm{Fml}}$ je v relaci $\vdash$ s $\varphi \in \mathrm{Fml}$, právě když je φ dokazatelná z T. Stejně tak sémantické vyplývání lze chápat jako relaci $\models \subseteq 2^{\mathrm{Fml}} \times \mathrm{Fml}$, kde $T \in 2^{\mathrm{Fml}}$ je v relaci $\models$ s $\varphi \in \mathrm{Fml}$, právě když φ sémanticky plyne z T.
Následující tvrzení ukazuje, že $\vdash \subseteq \models$.

Věta o korektnosti

Pro libovolnou množinu formulí T a formuli φ platí, že je-li $T \vdash \varphi$, pak $T \models \varphi$. Speciálně tedy, každá dokazatelná formule je tautologií.

Důkaz: Nejprve pro $T = \emptyset$. Každý axiom je tautologie (o čemž se lze snadno přesvědčit tabelací). Dále zřejmě platí, že jsou-li φ a $\varphi \Rightarrow \psi$ tautologie, je i ψ tautologie. Indukcí tedy dostáváme, že každý člen důkazu je tautologie. Tedy každá dokazatelná formule je tautologie.
Je-li $T \neq \emptyset$, pak z $T \vdash \varphi$ plyne, že pro nějaké $\psi_1, \ldots, \psi_n \in T$ je $\psi_1, \ldots, \psi_n \vdash \varphi$. Opakovaným ($n$-násobným) použitím VoD odtud dostaneme $\vdash \psi_1 \Rightarrow (\psi_2 \Rightarrow (\ldots(\psi_n \Rightarrow \varphi)\ldots))$, z čehož dle výše dokázaného plyne $\models \psi_1 \Rightarrow (\psi_2 \Rightarrow (\ldots(\psi_n \Rightarrow \varphi)\ldots))$. Nyní n-násobně použijeme „sémantické verze“ VoD a dostaneme $\psi_1, \ldots, \psi_n \models \varphi$, z čehož plyne $T \models \varphi$.

Důsledek

Sporný systém formulí není splnitelný.

Důkaz: Pokud je T sporný systém, pak $T \vdash \neg(\vartheta \Rightarrow \vartheta)$, tedy (dle VoK) $T \models \neg(\vartheta \Rightarrow \vartheta)$. Odtud dostáváme, že $\neg(\vartheta \Rightarrow \vartheta)$ musí být pravdivá při každém ohodnocení, při kterém jsou pravdivé všechny formule z T. Ale $\neg(\vartheta \Rightarrow \vartheta)$ je kontradikce, tedy neexistuje žádné ohodnocení e, při kterém by byly všechny formule z T pravdivé. Tím jsme prokázali, že sporný systém formulí není splnitelný.

Poznámka: Korektnost lze využít k prokázání faktu, že některá formule není dokazatelná z jistého systému předpokladů. Reformulací korektnosti totiž dostáváme, že pokud φ sémanticky neplyne z T, pak φ není ze systému T ani dokazatelná. K tomu, abychom prokázali, že $T \nvdash \varphi$ tedy stačí ukázat $T \not\models \varphi$, což je výrazně jednodušší než prokázat „neexistenci důkazu“, protože důkazů, jakožto konečných posloupností formulí, je obecně nekonečně mnoho.

Příklad

Prokážeme, že $p \Rightarrow q \nvdash \neg p \Rightarrow q$. Z Věty o korektnosti VL stačí ukázat, že $p \Rightarrow q \not\models \neg p \Rightarrow q$. To jest zbývá najít pravdivostní ohodnocení e takové, že $\| p \Rightarrow q \|_e = 1$, ale $\| \neg p \Rightarrow q \|_e = 0$. S využitím vlastností logické operace $\rightarrow$ zřejmě stačí vzít pravdivostní ohodnocení e, kde $e(p) = 0$ a $e(q) = 0$. Tím je důkaz hotov.

Shrneme-li předchozí poznatky, zavedli jsme dva druhy vyplývání: $\models, \vdash$ a již víme, že každá formule dokazatelná z prázdného systému je tautologie a obecněji $T \vdash \varphi$ implikuje $T \models \varphi$, neboli: „to co je dokazatelné z nějakého systému, z tohoto systému rovněž sémanticky plyne“.
Ukážeme, že to platí i obráceně.

Před důkazem věty o úplnosti zavedeme následující značení.
Pro formuli φ a ohodnocení e je

$$\varphi^e = \begin{cases} \varphi, & \text{pokud } \| \varphi \|_e = 1 \\ \neg\varphi, & \text{pokud } \| \varphi \|_e = 0. \end{cases}$$

Churchovo lemma (ChL)

Pro libovolnou formuli $\varphi(p_1, \ldots, p_n)$ platí $p_1^e, \ldots, p_n^e \vdash \varphi^e$.

Důkaz: Tvrzení dokážeme strukturální indukcí přes složitost formule φ.

I. Nechť φ je výrokový symbol p. Pak je tvrzení zřejmé $(p^e \vdash p^e)$.

II. Nechť tvrzení platí pro φ. Ukažme, že pak platí i pro $\neg\varphi$, tedy, že $p_1^e, \ldots, p_n^e \vdash (\neg\varphi)^e$. Rozlišme dva případy, $\| \varphi \|_e = 0$ a $\| \varphi \|_e = 1$. Pro $\| \varphi \|_e = 0$ je $\varphi^e = \neg\varphi$ a $(\neg\varphi)^e = \neg\varphi$. Požadované tvrzení $p_1^e, \ldots, p_n^e \vdash (\neg\varphi)^e$ tedy přímo plyne z předpokladu. Pro $\| \varphi \|_e = 1$ je $\varphi^e = \varphi$ a $(\neg\varphi)^e = \neg\neg\varphi$. Máme tedy dokázat, že $p_1^e, \ldots, p_n^e \vdash \neg\neg\varphi$. To však plyne z předpokladu: $p_1^e, \ldots, p_n^e \vdash \varphi$ a z $(c_\vdash)$: $\vdash \varphi \Rightarrow \neg\neg\varphi$ pomocí MP.

III. Nechť tvrzení platí pro $\varphi(p_1, \ldots, p_n)$ a $\psi(q_1, \ldots, q_m)$. Ukažme, že pak platí i pro $\varphi \Rightarrow \psi$, tedy, že $p_1^e, \ldots, p_n^e, q_1^e, \ldots, q_m^e \vdash (\varphi \Rightarrow \psi)^e$. Mohou nastat následující případy:

- $\| \varphi \|_e = 0$: Pak je $\| \varphi \Rightarrow \psi \|_e = 1$, tedy $(\varphi \Rightarrow \psi)^e = \varphi \Rightarrow \psi$. Podle předpokladu máme $p_1^e, \ldots, p_n^e \vdash \neg\varphi$. Dle ($a_\vdash$) je $\vdash \neg\varphi \Rightarrow (\varphi \Rightarrow \psi)$, odkud pomocí MP a MD dostaneme požadované $p_1^e, \ldots, p_n^e, q_1^e, \ldots, q_m^e \vdash \varphi \Rightarrow \psi$.
- $\| \psi \|_e = 1$: Pak je $\| \varphi \Rightarrow \psi \|_e = 1$, tedy opět $(\varphi \Rightarrow \psi)^e = \varphi \Rightarrow \psi$. Dle předpokladu máme $q_1^e, \ldots, q_m^e \vdash \psi$. Z (A1): $\psi \Rightarrow (\varphi \Rightarrow \psi)$, MP a MD dostaneme požadované $p_1^e, \ldots, p_n^e, q_1^e, \ldots, q_m^e \vdash \varphi \Rightarrow \psi$.
- $\| \varphi \|_e = 1$ a $\| \psi \|_e = 0$: Pak $\| \varphi \Rightarrow \psi \|_e = 0$, tedy $(\varphi \Rightarrow \psi)^e = \neg(\varphi \Rightarrow \psi)$. Podle předpokladu je $p_1^e, \ldots, p_n^e \vdash \varphi$ a $q_1^e, \ldots, q_m^e \vdash \neg\psi$. Použitím ($e_\vdash$): $\vdash \varphi \Rightarrow (\neg\psi \Rightarrow \neg(\varphi \Rightarrow \psi))$, MD a dvojnásobným použitím MP dostaneme požadované $p_1^e, \ldots, p_n^e, q_1^e, \ldots, q_m^e \vdash \neg(\varphi \Rightarrow \psi)$.

Věta o úplnosti, slabá verze

Pro libovolnou **konečnou** množinu T formulí a formuli φ platí, že z $T \models \varphi$ plyne $T \vdash \varphi$. Speciálně, každá pravdivá formule je dokazatelná.

Důkaz: Tvrzení dokážeme nejprve pro případ $T = \emptyset$. Nechť tedy $\models \varphi$. Pro každé ohodnocení e tedy platí $\varphi^e = \varphi$ (protože podle předpokladu je $\| \varphi \|_e = 1$). Jsou-li $p_1, \ldots, p_n$ všechny výrokové symboly z φ, je dle ChL

$$p_1^e, p_2^e, \ldots, p_n^e \vdash \varphi.$$

Uvažujme nyní ohodnocení e', které se od e liší právě v hodnotě, kterou přiřazuje symbolu p_1. Předpokládejme, že $e(p_1) = 1$ a $e'(p_1) = 0$ (případ $e(p_1) = 0$ a $e'(p_1) = 1$ se ošetří symetricky). Dle ChL je opět

$$p_1^{e'}, p_2^{e'}, \ldots, p_n^{e'} \vdash \varphi.$$

Protože je však podle předpokladu $p_2^e = p_2^{e'}, \ldots, p_n^e = p_n^{e'}$, $p_1^e = p_1$ a $p_1^{e'} = \neg p_1$, dostáváme

$$p_1, p_2^e, \ldots, p_n^e \vdash \varphi \qquad \text{a} \qquad \neg p_1, p_2^e, \ldots, p_n^e \vdash \varphi,$$

odkud dle VoNF máme

$$p_2^e, \ldots, p_n^e \vdash \varphi.$$

Opakovaným použitím právě provedené úvahy postupně dostaneme

$$p_3^e, \ldots, p_n^e \vdash \varphi$$

až po

$$p_n^e \vdash \varphi$$

a nakonec

$$\vdash \varphi.$$

Nechť nyní $T = \{\varphi_1, \ldots, \varphi_n\}$. Podle sémantické verze VoD dostaneme z $T \models \varphi$, že $\models \varphi_1 \Rightarrow (\ldots(\varphi_n \Rightarrow \varphi))$. Odtud podle právě dokázaného plyne $\vdash \varphi_1 \Rightarrow (\ldots(\varphi_n \Rightarrow \varphi))$, odkud pomocí VoD dostáváme $\varphi_1, \ldots, \varphi_n \vdash \varphi$, tedy požadované $T \vdash \varphi$. Tím je důkaz hotov.

Poznámka: věta o kompaktnosti

Pro důkaz tzv. silné verze věty o úplnosti (ta se neomezuje na konečné T) potřebujeme následující větu:

Věta o kompaktnosti

(1) Množina T formulí je splnitelná, právě když je splnitelná každá konečná podmnožina množiny T.

(2) Pro každou formuli φ je $T \models \varphi$, právě když existuje konečná $S \subseteq T$ tak, že $S \models \varphi$.

Důkaz: Vynecháme.

Poznámka: věta o úplnosti, silná verze

S použitím věty o kompaktnosti již snadno dokážeme silnou verzi věty o úplnosti.

Věta o úplnosti, silná verze

Pro libovolnou množinu T formulí a formuli φ platí, že z $T \models \varphi$ plyne $T \vdash \varphi$.

Důkaz: Je-li $T \models \varphi$, pak dle věty o kompaktnosti (2) existuje konečná $S \subseteq T$ tak, že $S \models \varphi$. Dle slabé verze věty o úplnosti je $S \vdash \varphi$, odkud z MD $T \vdash \varphi$.

Poznámka: Věta o úplnosti bývá často formulována jako ekvivalence, tedy tak, že v sobě zahrnuje i větu o korektnosti.

Uvědomme si, že věta o úplnosti je velmi netriviální tvrzení: Z toho, že nějaká formule má při všech (intuitivně zcela přirozeně definovaných) možných ohodnoceních pravdivostní hodnotu 1 plyne, že je dokazatelná pomocí tří (jednoduchých a intuitivně přijatelných) axiomů a jednoho (jednoduchého a intuitivně přijatelného) odvozovacího pravidla. Pojem pravdivého tvrzení, tak jak je formalizován v rámci VL, je tedy plně syntakticky charakterizovatelný (a navíc velmi jednoduchým způsobem).

Následující věta ukazuje další vztah dvojice pojmů, jednoho sémantického (splnitelnost) a druhého syntaktického (bezespornost), které spolu na první pohled nesouvisí.

Věta

Množina T formulí je splnitelná, právě když je bezesporná.

Důkaz: Nechť je T splnitelná. Pak existuje ohodnocení e, ve kterém jsou pravdivé všechny formule z T. Kdyby byla T sporná, pak by pro libovolnou formuli φ bylo $T \vdash \varphi$ a $T \vdash \neg\varphi$, a tedy dle VoK $T \models \varphi$ a $T \models \neg\varphi$. To znamená, že při každém ohodnocení, při kterém jsou pravdivé všechny formule z T (jedním z nich je e), je pravdivá jak formule φ, tak formule $\neg\varphi$. To je ale pochopitelně nemožné. Proto musí být T bezesporná.

Nechť T je bezesporná. Pak existuje formule φ, pro kterou neplatí $T \vdash \varphi$, tedy (podle věty o úplnosti) neplatí $T \models \varphi$. To ale znamená, že existuje ohodnocení, ve kterém není pravdivá φ, a přitom jsou pravdivé všechny formule z T. Tedy T je splnitelná.

Poznámka na okraj (1/2)

Kritika slov je soubor esejí a fejetonů Karla Čapka. Poprvé byl vydán v roce 1920.

V Kritice slov Karel Čapek napadá logiku: „O logickém důkazu je jediná pravda, že se nic nedá logicky dokazovat; což vám dokážu logicky. Buď dokazuji své tvrzení samými evidentními soudy; ale kdyby mé tvrzení plynulo evidentně z evidentních vět, bylo by samo evidentní, a tu by ovšem naprosto nepotřebovalo být dokazováno. Nebo dokazuji své tvrzení větami neevidentními, ale pak bych musel logicky dokazovat všechny tyto věty „usque ad infinitum“, jak říkají řeckořímští zápasníci, z čehož logicky plyne, že logický důkaz je nemožný; a není-li tento logický důkaz naprosto přesvědčující, vidíte z toho, že logické dokazování opravdu za nic nestojí.“

Poznámka na okraj (2/2)[1]

Karel Čapek mylně zpochybňuje důkaz (ústřední logický pojem) a spolu s ním i samotné logické vyvozování. Neuvědomuje si však, že pomocí velmi mnoha triviálních kroků lze logicky dokázat i netriviální tvrzení (například větu o úplnosti VL).

Rozložení důkazu na elementární kroky je důležité, umožňuje totiž ověření jeho správnosti.

I obyčejný člověk může pomocí velmi mnoha kroků urazit velkou vzdálenost. Přitom je pravdou, že po několika málo krocích se nikdo moc daleko od výchozího bodu nedostane.

[1] Poznámka o Karlovi Čapkovi byla čerpána z knihy A. Sochora: Logika pro všechny ochotné myslet, UK Praha, 2011.

Obsah

Formulemi VL jsme formalizovali intuitivní pojem výrok a dovedli jsme jimi popsat skládání složitějších výroků z jednodušších pomocí logických spojek. Výroky, které byly dále nedělitelné, jsme označovali výrokovými symboly a vnitřní strukturou těchto výroků jsme se nezabývali. Naproti tomu predikátová logika (PL) formalizuje vztahy mezi individui neboli objekty, například jejich funkční závislosti, vlastnosti a vzájemné vztahy. Oproti VL se tedy na tvrzení díváme daleko jemnějším pohledem a formule PL to musí pochopitelně zohledňovat. Nyní si na příkladu ukážeme, co máme konkrétně na mysli pod pojmem „vztahy mezi individui“.

Například tvrzení

„Pokud je x sudé číslo, pak je $x+1$ liché.“

je z pohledu VL ve tvaru imlikace dvou výroků a je tudíž formalizovatelné výrokovou formulí $p \Rightarrow q$. Z pohledu PL se ale ve tvrzení vyskytují individua (čísla), jejich vlastnosti (být sudé, být liché) a funkční závislosti ($x+1$ je následníkem x, nebo podrobněji 1 označuje individuum a „$+$“ označuje funkční závislost dvou individuí, v našem případě individuí označených x a 1).

Dalším typickým rysem je vytváření výroků kvantifikací, kterou ve slovním popisu vyjadřujeme obraty „každý“, „nějaký“, „právě jeden“ a podobně. Například ve tvrzení

Každý člověk má otce.

se vyskytuje kvantifikátor „každý“. Kdybychom si toto tvrzení reformulovali poněkud kostrbatěji, mohlo by znít: „Pro každého člověka platí, že má otce.“ Vazbu „mít otce“ bychom mohli chápat hned několika způsoby, například jako vlastnost (člověk *A* má otce), nebo třeba jako vztah dvou individuí (člověk *B* je otcem člověka *A*). Ve druhém případě je navíc ve tvrzení skryt další kvantifikátor: „ke každému člověku *A* existuje jeho otec *B*“.

Přirozený jazyk je tedy základním prostředkem, pomocí kterého formulujeme a zaznamenáváme své usuzování. Již základní rozbor vět přirozeného jazyka odhalí některé jeho významné jednotky/součásti. Pro nás to budou: proměnné, relační symboly (symboly pro označování relací), funkční symboly (symboly pro označování funkcí) včetně symbolů pro označování konstant, symboly pro logické spojky, symboly pro kvantifikátory a pomocné symboly.

Ve větách „Každý slon je savec.“, „Pro každé dva body existuje bod, který mezi nimi neleží“, „Petr je mladší než Pavel nebo věk Jiřího je větší než součet věků Petra a Pavla.“, „Součet druhých mocnin nenulových čísel je větší než nula.“ se mluví o jednomístných vztazích „být slonem“, „být savcem“, trojmístném vztahu „neležet mezi dvěma body“, dvojmístném vztahu „být větší“, o funkci přiřazující člověku jeho věk, o funkci sčítání, o funkci mocnění, o (konstantních) objektech Petr, Pavel, Jiří, nula, implicitně se zde objevují proměnné (například první tvrzení, formulováno přesneji, říká „pro každé x platí, že je-li x slonem, je x savcem“), logické spojky („nebo“) a kvantifikátory („pro každé“, „existuje“).

Stejně jako ve VL se budeme v PL soustředit na formu usuzovaného a budeme abstrahovat od obsahu sdělení.

Definice

Jazyk $\mathscr{J}$ **PL** obsahuje (a je tím určen)

- (**předmětové**) **proměnné** $x, y, z, \ldots, x_1, x_2, \ldots$
- **relační symboly** $p, q, r, \ldots, p_1, p_2 \ldots$, ke každému relačnímu symbolu r je dáno nezáporné celé číslo $\sigma(r)$ nazývané arita symbolu r; musí existovat alespoň jeden relační symbol
- **funkční symboly** $f, g, h, \ldots, f_1, f_2, \ldots$ ke každému funkčnímu symbolu f je dáno nezáporné celé číslo $\sigma(f)$ nazývané arita symbolu f
- **symboly pro logické spojky** $\neg$ (negace) a $\Rightarrow$ (implikace)
- **symbol pro univerzální kvantifikátor** $\forall$
- **pomocné symboly** – různé typy závorek a čárka.

Místo předmětové proměnné často říkáme jen proměnné. Předpokládáme, že proměnných je spočetně mnoho.

Množina všech relačních (někdy se říká predikátových) symbolů jazyka $\mathscr{J}$ se značí R; množina všech funkčních (někdy se říká operačních) symbolů jazyka $\mathscr{J}$ se značí F. Je-li $r \in R$ a $\sigma(r) = n$, pak říkáme, že r je **n-ární**. Podobně pro $f \in F$. Je-li $f \in F$ 0-ární, nazývá se f **symbol konstanty** (neboť funkce, která má 0 argumentů, musí přiřazovat vždy stejnou hodnotu, tedy je konstantní).

Je zřejmé, že jazyk je jednoznačně určen svými relačními symboly, funkčními symboly a jejich aritami (vše ostatní mají všechny jazyky stejné). Trojici $\langle R, F, \sigma \rangle$ proto nazýváme **typ jazyka**. (Pochopitelně předpokládáme, že $R \cap F = \emptyset$.)

Je-li mezi relačními symboly symbol $\approx$, nazýváme ho **symbol pro rovnost** a celý jazyk pak **jazyk s rovností**. (Symbol pro rovnost má specifické postavení, jak uvidíme dále.)

Termy PL

Základní syntaktické jednotky vybudované ze symbolů jazyka PL jsou termy a formule. Termy jsou výrazy reprezentující funkci aplikovanou na své operandy; formule reprezentují tvrzení o prvcích univerza.

Definice

Term jazyka typu $\langle R, F, \sigma \rangle$ je induktivně definován takto:

(i) každá proměnná x je term

(ii) je-li $f \in F$ n-ární a jsou-li $t_1, \ldots, t_n$ termy, pak $f(t_1, \ldots, t_n)$ je term.

Termy jsou tedy jisté konečné posloupnosti prvků daného jazyka. Je-li $f \in F$ binární, používáme také tzv. infixovou notaci a píšeme $x\, f\, y$ nebo $(x\, f\, y)$ místo $f(x, y)$, například $2+3$ místo $+(2,3)$; ve složených termech používáme závorky, například $(2+3)\cdot 5$.

Definice

Formule jazyka typu $\langle R, F, \sigma \rangle$ je induktivně definována takto:

(i) je-li $r \in R$ n-ární a jsou-li $t_1, \ldots, t_n$ termy, pak $r(t_1, \ldots, t_n)$ je formule

(ii) jsou-li φ a ψ formule, pak $\neg\varphi$, $(\varphi \Rightarrow \psi)$ jsou také formule

(iii) je-li φ formule a x proměnná, pak $(\forall x)\varphi$ je formule.

Formule vytvořené dle (i) se nazývají **atomické**. Je-li $r \in R$ binární, píšeme také $t_1\, r\, t_2$ nebo $(t_1\, r\, t_2)$ místo $r(t_1, t_2)$, tedy například $x \leq y$ místo $\leq (x, y)$. Obzvláště píšeme $t_1 \approx t_2$ místo $\approx (t_1, t_2)$.

Budeme používat obvyklé konvence, zjednodušující čitelnost formulí: budeme vynechávat vnější závorky a budeme psát $(\forall x_1, \ldots, x_n)$ místo $(\forall x_1) \ldots (\forall x_n)$.

Stejně jako ve VL, symboly pro ostatní logické spojky ($\vee, \wedge, \Leftrightarrow$) nepatří do jazyka PL. Podobně existenční kvantifikátor ($\exists$) nepatří do jazyka PL. Abychom tyto symboly měli k dispozici, chápeme

posloupnost $\varphi \wedge \psi$ jako zkratku za $\neg(\varphi \Rightarrow \neg\psi)$,

posloupnost $\varphi \vee \psi$ jako zkratku za $\neg\varphi \Rightarrow \psi$,

posloupnost $\varphi \Leftrightarrow \psi$ jako zkratku za $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$,

posloupnost $(\exists x)\varphi$ jako zkratku za $\neg(\forall x)\neg\varphi$.

Posloupnosti obsahující symboly $\wedge, \vee, \Leftrightarrow$ a $\exists$ tedy nejsou formulemi. Pro jednoduchost však vědomi si toho, že a jakým způsobem se dopouštíme nepřesnosti, budeme těmto posloupnostem také říkat formule.

Příklad

Uvažujme jazyk $\mathscr{J}$ typu $\langle R, F, \sigma \rangle$, kde $R = \{p, d, b, m\}$, $F = \emptyset$, relační symboly p, d, b jsou unární, m binární. Je to jazyk bez funkčních symbolů, a tedy jedinými termy jsou proměnné. Atomickými formulemi jsou $p(x), p(y), d(y), b(x)$ atd. Dalšími formulemi (ne atomickými) jsou například výrazy $p(x) \vee p(x)$, $p(x) \vee \neg p(x)$, $(\forall x)((p(x) \wedge \neg d(x)) \Rightarrow b(x))$, $(\exists x) b(x) \wedge \neg p(x)$, $(\forall x, y)(b(x) \Rightarrow m(x, y)) \Rightarrow b(y)$. Výrazy $((x \Rightarrow$, $p(x, x)$, $p(d(x))$, $(\forall x)(m(x, y) \Rightarrow\Rightarrow p(x))$ formulemi nejsou.

Příklad

Uvažujme jazyk $\mathscr{J}$ typu $\langle R, F, \sigma \rangle$, kde $R = \{p, \leq\}$, $F = \{c, \circ\}$, c je nulární (tedy symbol konstanty), p je unární, $\leq, \circ$ jsou binární. Pak termy jsou například výrazy $c, x, c \circ c, c \circ (x \circ y)$. Výrazy $cc, c \circ p(x), p(x), c \circ \circ x$ termy nejsou. Formulemi jsou například $c \leq x$, $p(x) \Rightarrow (c \leq x)$, $(\forall x)(x \leq x \circ x)$, $(\forall x, y)(x \circ y \leq y \circ x)$.

Poznámka: Jazyk PL obsahuje symboly různých typů (relační symboly, funkční symboly, symboly spojek). Z nich se dají vytvářet termy a formule, které jsou v určitém smyslu „rozumnými řetězci“. Rozumné proto, že dáme-li relačním a funkčním symbolům smysl, dají se „rozumně“ číst. (Přesný smysl dostanou relační a funkční symboly, a také termy a formule, až vybudujeme sémantiku.) Tak například uvažujme jazyk z 1. příkladu předchozího slajdu. Nechť p, d, b, m označují po řadě „mít dostatečný příjem“, „mít velké dluhy“, „být bonitní“, „být manželé“, tedy $p(x)$ znamená „objekt označený x má dostatečný příjem“ atd. Formule $(\forall x)(p(x) \wedge \neg d(x) \Rightarrow b(x))$ pak „říká“: „pro každé x platí, že má-li x dostatečný příjem a nemá-li velké dluhy, pak je x bonitní“. Formule $(\forall x, y)(b(x) \Rightarrow (m(x, y) \Rightarrow b(y)))$ „říká“: „pro každé x a y platí, že je-li x bonitní a jsou-li x a y manželé, je i y bonitní“.

Můžeme také postupovat obráceně: K dané větě přirozeného jazyka navrhneme jazyk PL a napíšeme formuli, která odpovídá danému tvrzení.

Podobně jako ve VL můžeme i v PL provádět důkazy strukturální indukcí.

Věta – důkaz strukturální indukcí pro termy

Nechť $\mathscr{V}$ je vlastnost termů. Nechť platí, že

- každá proměnná má vlastnost $\mathscr{V}$
- mají-li termy $t_1, \ldots, t_n$ vlastnost $\mathscr{V}$ a je-li $f \in F$ n-ární, pak vlastnost $\mathscr{V}$ má i term $f(t_1, \ldots, t_n)$.

Pak vlastnost $\mathscr{V}$ má každý term.

Věta – důkaz strukturální indukcí pro formule

Nechť $\mathscr{V}$ je vlastnost formulí. Nechť platí, že

- každá atomická formule má vlastnost $\mathscr{V}$
- mají-li formule φ a ψ vlastnost $\mathscr{V}$, pak vlastnost $\mathscr{V}$ mají i formule $\neg\varphi$ a $(\varphi \Rightarrow \psi)$
- má-li formule φ vlastnost $\mathscr{V}$, pak vlastnost $\mathscr{V}$ má i formule $(\forall x)\varphi$ s proměnnou x.

Pak vlastnost $\mathscr{V}$ má každá formule PL.

Poznámka: Chceme-li napsat formuli nebo term, musíme nejdříve popsat jazyk, jehož formuli nebo term chceme napsat. To je však často pracné a zbytečné. Proto zavedeme pojem indukovaného jazyka. **Jazyk indukovaný** množinami $\mathscr{F}$ a $\mathscr{T}$ řetězců je nejmenší jazyk $\mathscr{J}$ typu $\langle R, F, \sigma \rangle$, v němž řetězce z $\mathscr{F}$ jsou formulemi a řetězce z $\mathscr{T}$ jsou termy (tedy je-li $\mathscr{J}'$ jiný takový jazyk typu $\langle R', F', \sigma' \rangle$, pak $R \subseteq R'$, $F \subseteq F'$). Místo jazyk indukovaný množinami $\mathscr{F}$ a $\mathscr{T}$ řetězců také říkáme jazyk určený formulemi z $\mathscr{F}$ a termy z $\mathscr{T}$.

Příklad

Uvažujme jazyk určený termy $t_1 = x_1 + ((c + x_1) + x_2)$,
$t_2 = (c + c) + c$ a formulemi
$\varphi_1 = (\forall x, y)((r(x) \wedge (x \leq y)) \Rightarrow r(y))$,
$\varphi_2 = (c \leq x) \wedge (\forall x)(\exists y)(x + x \leq x + y)$.
Pak zřejmě $R = \{r, \leq\}$, $F = \{c, +\}$, kde $\sigma(r) = 1$, $\sigma(\leq) = 2$, $\sigma(c) = 0$ a $\sigma(+) = 2$.

Obsah

Jazyk PL je určen svými relačními a funkčními symboly spolu s definicí jejich arity. Z těchto symbolů spolu se symboly proměnných, logických spojek, kvantifikátorů a pomocných symbolů se skládají termy a formule daného jazyka. Samotné termy a formule jsou syntaktické pojmy a nemají žádný význam (tedy term sám o sobě nemá žádnou hodnotu, formule sama o sobě nemá žádnou pravdivostní hodnotu). To je například dobře patrné u termu $x + 0$: abychom mohli uvažovat hodnotu tohoto termu, musí mít přiřazenu nějakou hodnotu proměnná x a dále musíme interpretovat symboly $+$ a 0.

Interpretací funkčních a relačních symbolů se zabývá sémantika PL (ta přiřazuje význam funkčním a relačním symbolům). Poznamenejme ještě, že interpretaci proměnných definuje tzv. ohodnocení proměnných (viz dále).

Struktura pro jazyk

Intuitivní pojem interpretace jazyka nyní přesně zavedeme:

Definice

Struktura pro jazyk typu $\langle R, F, \sigma \rangle$ je trojice $\mathbf{M} = \langle M, R^{\mathbf{M}}, F^{\mathbf{M}} \rangle$, která sestává z neprázdné množiny M a dále z množin

$$R^{\mathbf{M}} = \{ r^{\mathbf{M}} \subseteq M^n \mid r \in R, \sigma(r) = n \},$$

$$F^{\mathbf{M}} = \{ f^{\mathbf{M}} : M^n \to M \mid f \in F, \sigma(f) = n \}.$$

Pokud $\approx \in R$, pak $\approx$ interpretujeme vždy relací identity, tedy $\approx^{\mathbf{M}} = \omega_M = \{ \langle u, u \rangle \mid u \in M \}$.

Jinými slovy, struktura $\mathbf{M}$ pro jazyk typu $\langle R, F, \sigma \rangle$ je systém relací a funkcí na jisté množině M, přitom ke každému n-árnímu relačnímu symbolu $r \in R$ je ve struktuře $\mathbf{M}$ odpovídající n-ární relace $r^{\mathbf{M}} \in R^{\mathbf{M}}$ na M a ke každému n-árnímu funkčnímu symbolu $f \in F$ je ve struktuře $\mathbf{M}$ odpovídající n-ární funkce $f^{\mathbf{M}} \in F^{\mathbf{M}}$ v M. Nehrozí-li nebezpečí nedorozumění, budeme někdy vynechávat horní indexy a místo $r^{\mathbf{M}}$ a $f^{\mathbf{M}}$ budeme psát jen r a f.

Příklad

Uvažujme jazyk typu $\langle R, F, \sigma \rangle$, kde $R = \{p, \leq\}$, $F = \{c, \circ\}$, c je nulární, p je unární, $\leq$ a $\circ$ jsou binární. Nechť $M = \mathbb{Z}$.
Definujme relace $p^{\mathbf{M}}$ (unární, tedy podmnožina M), $\leq^{\mathbf{M}}$ a funkce $c^{\mathbf{M}}$ (nulární, tedy vybraný prvek z M) a $\circ^{\mathbf{M}}$ následovně:

$$p^{\mathbf{M}} = \{m \in M \mid m \text{ je větší nebo rovno nule}\},$$
$$\leq^{\mathbf{M}} = \{\langle m_1, m_2 \rangle \in M \times M \mid m_1 \text{ je menší nebo rovno } m_2\},$$
$$c^{\mathbf{M}} = 0, \quad m_1 \circ^{\mathbf{M}} m_2 = m_1 + m_2,$$

tedy $c^{\mathbf{M}}$ je číslo nula a $\circ^{\mathbf{M}}$ je operace sčítání celých čísel.

Jinou strukturu pro stejný jazyk dostaneme, pokud změníme výše uvedenou strukturu tak, že $c^{\mathbf{M}} = 1$, případně ještě definujeme $m_1 \circ^{\mathbf{M}} m_2 = m_1 \cdot m_2$ (násobení celých čísel).

Další strukturou (opět) pro stejný jazyk je struktura s nosičem $M = \{a, b\}$, $p^{\mathbf{M}} = \{a, b\}$, $\leq^{\mathbf{M}} = \{\langle a, a \rangle, \langle b, b \rangle, \langle b, a \rangle\}$, $c^{\mathbf{M}} = b$ a s operací $\circ^{\mathbf{M}}$ definovanou tabulkou:

$\circ^{\mathbf{M}}$	a	b
a	a	b
b	a	a

Jak tedy vidíme, k danému jazyku PL existuje nekonečně mnoho struktur. Variabilita je dána jednak nosičem M, který může mít libovolný (nenulový) počet prvků, dále každý relační symbol r může být interpretován libovolnou relací $r^{\mathbf{M}}$ příslušné arity a konečně každý funkční symbol f může být interpretován libovolnou funkcí $f^{\mathbf{M}}$ příslušné arity.

Nechť **M** je struktura pro jazyk typu $\langle R, F, \sigma \rangle$. **M-ohodnocení proměnných** (krátce jen **M-ohodnocení**, popř. jen **ohodnocení**) je zobrazení v přiřazující každé proměnné x prvek $v(x) \in M$. Jsou-li v a v' ohodnocení a x je proměnná, píšeme $v =_x v'$ pokud pro každou proměnnou $y \neq x$ je $v(y) = v'(y)$, tedy v a v' se liší nejvýše v tom, jakou hodnotu přiřazují proměnné x.

Definice

Nechť v je **M**-ohodnocení. **Hodnota $\| \mathbf{t} \|_{\mathbf{M},\mathbf{v}}$ termu t v M při v** je definována

$$\| t \|_{\mathbf{M},v} = \begin{cases} v(x), & \text{je-li } t \text{ proměnná } x \\ f^{\mathbf{M}}(\| t_1 \|_{\mathbf{M},v}, \ldots, \| t_k \|_{\mathbf{M},v}), & \text{je-li } t \text{ tvaru } f(t_1, \ldots, t_k). \end{cases}$$

Uvědomme si, že při dané struktuře **M** a při daném **M**-ohodnocení v je každému termu t přiřazena právě jedna hodnota $\| t \|_{\mathbf{M},v}$ z univerza M. Dále je patrné, že hodnota $\| t \|_{\mathbf{M},v}$ nezávisí na hodnotách přiřazených ohodnocením v těm proměnným, které se v t nevyskytují (což lze dokázat jednoduše strukturální indukcí).

Příklad

Uvažujme jazyk typu $\langle\{p,\leq\},\{c,\circ\},\sigma\rangle$, kde $\sigma(c)=0$, $\sigma(p)=1$, $\sigma(\leq)=\sigma(\circ)=2$. Nechť $M=\mathbb{Z}$. Definujme relace $p^{\mathbf{M}}$, $\leq^{\mathbf{M}}$ a funkce $c^{\mathbf{M}}$ a $\circ^{\mathbf{M}}$ následovně:

$$p^{\mathbf{M}}=\{m\in M \mid m \text{ je větší nebo rovno nule}\},$$
$$\leq^{\mathbf{M}}=\{\langle m_1,m_2\rangle\in M\times M \mid m_1 \text{ je menší nebo rovno } m_2\},$$
$$c^{\mathbf{M}}=0,\quad m_1\circ^{\mathbf{M}} m_2=m_1+m_2.$$

Vezmeme-li v této struktuře term $(x\circ(c\circ y))\circ x$, pak při ohodnocení v, kde $v(x)=10$, $v(y)=50$ máme

$$\| (x\circ(c\circ y))\circ x \|_{\mathbf{M},v}=\| x\circ(c\circ y) \|_{\mathbf{M},v}+\| x \|_{\mathbf{M},v}=$$
$$=(\| x \|_{\mathbf{M},v}+\| c\circ y \|_{\mathbf{M},v})+\| x \|_{\mathbf{M},v}=$$
$$=(\| x \|_{\mathbf{M},v}+(\| c \|_{\mathbf{M},v}+\| y \|_{\mathbf{M},v}))+\| x \|_{\mathbf{M},v}=$$
$$=(v(x)+(c^{\mathbf{M}}+v(y)))+v(x)=(10+(0+50))+10=70.$$

Dále definujeme pravdivostní hodnotu formule ve struktuře při daném ohodnocení.

Definice

Pravdivostní hodnota $\| \varphi \|_{\mathbf{M},\mathbf{v}}$ **formule** φ **při M-ohodnocení v** je definována následovně:

(i) pro atomické formule

$$\| r(t_1,\ldots,t_n) \|_{\mathbf{M},v} = \begin{cases} 1, & \text{je-li } \langle \| t_1 \|_{\mathbf{M},v},\ldots,\| t_n \|_{\mathbf{M},v} \rangle \in r^{\mathbf{M}} \\ 0, & \text{jinak} \end{cases}$$

(ii) pro formule φ ve tvaru $\neg\alpha$ a $\alpha \Rightarrow \beta$

$$\| \neg\alpha \|_{\mathbf{M},v} = \begin{cases} 1, & \text{pokud } \| \alpha \|_{\mathbf{M},v} = 0 \\ 0, & \text{pokud } \| \alpha \|_{\mathbf{M},v} = 1 \end{cases}$$

$$\| \alpha \Rightarrow \beta \|_{\mathbf{M},v} = \begin{cases} 1, & \text{pokud } \| \alpha \|_{\mathbf{M},v} = 0 \text{ nebo} \\ & \| \beta \|_{\mathbf{M},v} = 1 \\ 0, & \text{jinak} \end{cases}$$

(iii) pro kvantifikovanou formuli φ

$$\| (\forall x)\varphi \|_{\mathbf{M},v} = \begin{cases} 1, & \text{pokud pro každé } v' \text{ takové, že} \\ & v' =_x v \text{ je } \| \varphi \|_{\mathbf{M},v'} = 1 \\ 0, & \text{jinak.} \end{cases}$$

Je-li $\| \varphi \|_{\mathbf{M},v} = 1$ ($\| \varphi \|_{\mathbf{M},v} = 0$), říkáme, že formule φ je **pravdivá** (**nepravdivá**) **ve struktuře M při ohodnocení v**.

Stejně jako u ohodnocení termů je (při daných $\mathbf{M}$ a v) každé formuli φ přiřazena právě jedna hodnota $\| \varphi \|_{\mathbf{M},v}$.

Strukturální indukcí lze jednoduše dokázat, že hodnota $\| \varphi \|_{\mathbf{M},v}$ nezávisí na hodnotách přiřazených ohodnocením v proměnným, které se ve φ nevyskytují.

Uvědomme si, že říct: „formule φ je pravdivá“ nemá smysl, protože pravdivost φ vztahujeme vždy k nějaké struktuře při některém ohodnocení proměnných.
Běžně sice říkáme například „formule $(\forall x)(\forall y)x \leq x + \mathrm{abs}(y)$ je pravdivá“, ale to je způsobeno tím, že implicitně nějakou strukturu předpokládáme dle kontextu, ve kterém formuli uvažujeme. Třeba v matematické analýze jde většinou o číselné struktury, například reálná čísla s běžnými relacemi („menší nebo rovno“) a operacemi („sčítání reálných čísel“, „absolutní hodnota“).

Nyní budeme zkoumat platnost formulí ve struktuře „přes všechna ohodnocení“ a platnost formulí přes všechny struktury.

Definice

Formule φ se nazývá **tautologie ve struktuře** (**pravdivá ve struktuře**) **M**, jestliže $\| \varphi \|_{\mathbf{M},v} = 1$ pro každé **M**-ohodnocení v. Formule φ se nazývá **tautologie**, jestliže je φ tautologie v každé struktuře **M**.

Formule φ je tedy tautologie, jestliže pro libovolnou strukturu **M** a libovolné ohodnocení v je $\| \varphi \|_{\mathbf{M},v} = 1$.

Definice

Teorie v jazyku PL typu $\langle R, F, \sigma \rangle$ je libovolná množina T formulí jazyka tohoto typu. Struktura **M** jazyka typu $\langle R, F, \sigma \rangle$ se nazývá **model teorie** T (píšeme $\mathbf{M} \models T$, popř. $\| T \|_{\mathbf{M}} = 1$), jestliže každá formule z T je pravdivá v **M**.

Poznámka: Teorie formalizuje soubor předpokladů.
Pojem teorie je zcela přirozený. Běžně se říká „S tvojí teorií nesouhlasím.“ apod. Přitom teorií rozumíme soubor tvrzení, které daná osoba zastává. Soubor tvrzení v PL představuje množina formulí. Též pojem model je přirozený a vyskytuje se v běžné komunikaci. Například obratem „Představme si modelovou situaci, kdy . . . “ chceme vyjádřit, abychom se soustředili na nějaký konkrétní model jisté teorie.

Příklad

Uvažujme jazyk $\mathscr{J}$ typu $\langle R, F, \sigma \rangle$, kde $R = \{r\}$, $F = \emptyset$ a $\sigma(r) = 2$. Struktury pro $\mathscr{J}$ jsou $\mathbf{M} = \langle M, \{r^{\mathbf{M}}\}, \emptyset \rangle$, kde $r^{\mathbf{M}}$ je binární relace na M (tedy struktury jsou vlastně binární relace na M). Struktura $\mathbf{M}$ je modelem teorie $T = \{(\forall x) r(x,x), (\forall x,y,z)((r(x,y) \wedge r(y,z)) \Rightarrow r(x,z))\}$, právě když je relace $r^{\mathbf{M}}$ reflexivní a tranzitivní.

Poznámka: Některé teorie nemají model.

Příklad

Mějme jazyk typu $\langle R, F, \sigma \rangle$, kde $R = \{r\}$, $F = \emptyset$ a r je unární. Teorie $T = \{(\forall x)r(x), (\exists x)\neg r(x)\}$ zřejmě nemá žádný model.

Nyní zavedeme sémantické vyplývání v PL.

Definice

Množina S formulí **sémanticky plyne** z množiny T formulí (píšeme $T \models S$; píšeme také $T \models \varphi$, jestliže $S = \{\varphi\}$, podobně když $T = \{\psi\}$), jestliže každý model T je modelem S.

Tedy $T \models S$, právě když v každé struktuře, ve které jsou pravdivé všechny formule z T, jsou také pravdivé všechny formule z S.

Všimněme si, že pojem sémantického vyplývání je zaveden analogicky jako v případě VL, jen místo „pravdivostních ohodnocení“ používáme z pochopitelných důvodů pojem model.
Dále platí, že φ je tautologie, právě když $\models \varphi$.

Vidíme i jak prokázat, že daná formule φ sémanticky neplyne z teorie T. Stačí najít jediný model $\mathbf{M} \models T$ a $\mathbf{M}$-ohodnocení v takové, že $\| \varphi \|_{\mathbf{M},v} = 0$ (což nemusí být vůbec jednoduché).
Podotkněme ještě, že daleko větším problémem je ověření, zda-li φ z T sémanticky plyne.

Příklad

Formule $\varphi = (\forall x, y, z, w)((r(x,y) \wedge r(y,z) \wedge r(z,w)) \Rightarrow r(x,w))$ sémanticky plyne z formule
$\psi = (\forall x, y, z)((r(x,y) \wedge r(y,z)) \Rightarrow r(x,z))$, tedy $\psi \models \varphi$.
Obrácené vyplývání, tedy $\varphi \models \psi$, neplatí.

Příklad

Následující tautologie vyjadřují záměnu pořadí kvantifikátorů:

$$\models (\forall x)(\forall y)\varphi \Leftrightarrow (\forall y)(\forall x)\varphi,$$
$$\models (\exists x)(\exists y)\varphi \Leftrightarrow (\exists y)(\exists x)\varphi,$$
$$\models (\exists x)(\forall y)\varphi \Rightarrow (\forall y)(\exists x)\varphi.$$

Ukažme, že implikaci ve 3. tautologii nelze obrátit. Uvažujme jazyk typu $\langle R, \emptyset, \sigma \rangle$, kde $R = \{r\}$, $\sigma(r) = 2$ a strukturu tohoto jazyka $\mathbf{M} = \langle M, \{r^{\mathbf{M}}\}, \emptyset \rangle$, kde $M = \{a, b\}$ a relace $r^{\mathbf{M}}$ je definována $r^{\mathbf{M}} = \{\langle a, b \rangle, \langle b, a \rangle\}$. Ve struktuře $\mathbf{M}$ máme $\| (\forall y)(\exists x) r(x, y) \|_{\mathbf{M},v} = 1$ při libovolném ohodnocení v. Na druhou stranu však $\| (\exists x)(\forall y) r(x, y) \|_{\mathbf{M},v} = 0$, tedy máme model, ve kterém není $(\forall y)(\exists x)\varphi \Rightarrow (\exists x)(\forall y)\varphi$ pravdivá.

Poznámka k větě o korektnosti a větě o úplnosti PL

Podobně jako ve VL lze i v PL vybudovat axiomatický systém, definovat pojem důkaz z množiny formulí a dokázat větu o korektnosti i větu o úplnosti PL. Platí tedy, že pro každou teorii T a každou formuli φ: $T \vdash \varphi$, právě když $T \models \varphi$.

Poznámky k omezením PL

K omezením PL, které nelze jednoduše (resp. vůbec) vyřešit jejím rozšířením, ať už o nové spojky nebo o další pravdivostní hodnoty, patří její **nerozhodnutelnost**. Neformálně řečeno, neexistuje žádný algoritmus, který by o vstupní teorii T a formuli φ dokázal po konečném počtu kroků říct, zda-li φ je sémantickým důsledkem T.

Dále nelze axiomaticky vymezit vlastnost „být konečný“. Jinými slovy, neexistuje žádná teorie T taková, že $\mathbf{M}$ je modelem T, právě když je $\mathbf{M}$ struktura s konečným nosičem. Také nelze axiomaticky vymezit, aby byl symbol rovnosti $\approx$ interpretován relací identity.

Zmiňme ještě jedno omezení: PL má jen dvě základní pravdivostní hodnoty. Studiem více pravdivostních hodnot a studiem vyplývání v prostředí vágnosti se zabývá fuzzy logika.

Obsah

Logické programování je jedním z paradigmat programování. Následují základní rysy, kterými se logické programování zásadně odlišuje od ostatních programovacích stylů:

(1) programátor specifikuje, co se má vypočítat, a ne jak se to má vypočítat a kam uložit mezivýsledky

(2) nejsou potřeba příkazy pro řízení běhu výpočtu ani pro řízení toku dat, nejsou potřeba příkazy cyklů, větvení, ani přiřazovací příkaz

(3) neexistuje rozdělení proměnných na vstupní a výstupní

(4) nerozlišuje se mezi daty a programem.

PROLOG je logický programovací jazyk. Vznikl ve Francii (v Marseille) začátkem 70. let minulého století. Byl vytvořen Alainem Colmeranerem ve spolupráci s Philippem Rousselem, na základě procedurálního výkladu Hornovy klauzule od Roberta Kowalskiho. Jméno „PROLOG“ vzniklo jako zkratka z „PROgrammation à LOGic“ (francouzsky „programování v logice“).

1. program v PROLOGu

Víme, že se dá z Prahy letět přímo do Paříže a do Lyonu. Dále víme, že se dá přímo letět z Paříže do Marseille a z Marseille přímo do Vídně. Letecká spojení mezi městy jsou buď přímá nebo přes nějaká další města. Uvedené informace přepíšeme tak, aby syntaxe odpovídala PROLOGu. Dostaneme tzv. formalizovanou znalostní bázi (jednoduchý logický program):

```
direct(praha,pariz).
direct(praha,lyon).
direct(pariz,marseille).
direct(marseille,viden).
connection(X,Y) :- direct(X,Y).
connection(X,Z) :- direct(X,Y), connection(Y,Z).
```

Podrobný komentář na přednášce.

PROLOG je interpretační (neprocedurální) jazyk. Patří mezi deklarativní programovací jazyky – potlačuje imperativní[2] složku. PROLOG je využíván především v oboru umělé inteligence a v počítačové lingvistice (obzvláště pro zpracování přirozeného jazyka, pro nějž byl původně navržen).

PROLOG je založen na PL (prvního řádu); konkrétně se omezuje na Hornovy klauzule[3]. Základními využívanými přístupy jsou unifikace (speciální substituce), rekurze (využívá principu sebeopakování, viz dále) a backtracking (metoda prohledávání do hloubky).

---

[2] Imperativní paradigma popisuje, jak vyřešit problém. Deklarativní paradigma popisuje, co je problém.

[3] Hornova klauzule je formule (ve tvaru disjunkce atomických formulí), která obsahuje nejvýše jednu nenegovanou atomickou formuli (ostatní jsou právě jednou negované).

Program a výpočet v logickém programování

Základní koncepci logického programování vyjadřuje následující dvojice „rovností“:

$$\begin{aligned} \text{program} &= \text{množina axiomů} \\ \text{výpočet} &= \text{konstruktivní důkaz uživatelem zadaného cíle,} \end{aligned}$$

nebo volněji: program je souborem tvrzení, kterými programátor (uživatel, expert) popisuje určitou část okolního světa. Výpočet nad daným programem, který je iniciován zadáním dotazu, je hledání důkazu dotazu z daného souboru tvrzení[4].

[4] Přesněji řečeno, znegovaný dotaz se přidá ke všem faktům a pravidlům a s využitím rezolučního odvozovacího pravidla se snaží dojít ke sporu.

Shrňme a doplňme základní rysy logického programování:

- logický program je konečná množina formulí speciálního tvaru (v PROLOGu to jsou Hornovy klauzule)
- výpočet je zahájen zadáním dotazu, což je logická formule, kterou zadá uživatel
- cílem výpočtu je najít důkaz potvrzující, že dotaz logicky vyplývá (je dokazatelný) z logického programu (konstruktivnost)
- pokud je takto zjištěno, že dotaz z programu vyplývá, výpočet končí a uživateli je oznámeno „Yes“ s hodnotami případných proměnných, které se v dotazu vyskytují
- pokud není zjištěno, že dotaz z programu vyplývá, výpočet končí a uživateli je oznámeno „No“
- může se stát, že výpočet neskončí (například uvázne v nekonečném cyklu).

Základem PROLOGu je databáze klauzulí, které lze dále rozdělit na **fakta** a **pravidla**, nad kterými je možno klást **dotazy** formou tvrzení, u kterých PROLOG zhodnocuje jejich pravdivost (dokazatelnost z údajů obsažených v databázi).

Nejjednoduššími klauzulemi jsou fakta, která pouze vypovídají o vlastnostech objektu nebo vztazích mezi objekty. Složitějšími klauzulemi jsou pravidla, která umožňují (pomocí implikace) odvozovat nová fakta. Zapisují se ve tvaru

```
hlava :- tělo.,
```

přičemž `hlava` definuje odvozovaný fakt a `tělo` podmínky, za nichž je pravdivý. Tělo pravidla obsahuje jeden či více cílů. Pokud se interpretu podaří odvodit, že je tělo pravdivé, ověří tím pravdivost hlavy.

2. program v PROLOGu

```
rodic(pavla,robert).  rodic(tom,robert).
rodic(tom,liza).  rodic(robert,anna).
rodic(robert,patricie).  rodic(patricie,jan).
zena(pavla).  zena(liza).
zena(anna).  zena(patricie).
muz(tom).  muz(robert).  muz(jan).
potomek(Y,X) :- rodic(X,Y).
matka(X,Y) :- rodic(X,Y), zena(X).
prarodic(X,Z) :- rodic(X,Y), rodic(Y,Z).
predek(X,Z) :- rodic(X,Z).
predek(X,Z) :- rodic(X,Y), predek(Y,Z).
sestra(X,Y) :-
        rodic(Z,X), rodic(Z,Y), zena(X), not(X=Y).
```

Podrobný komentář na přednášce.

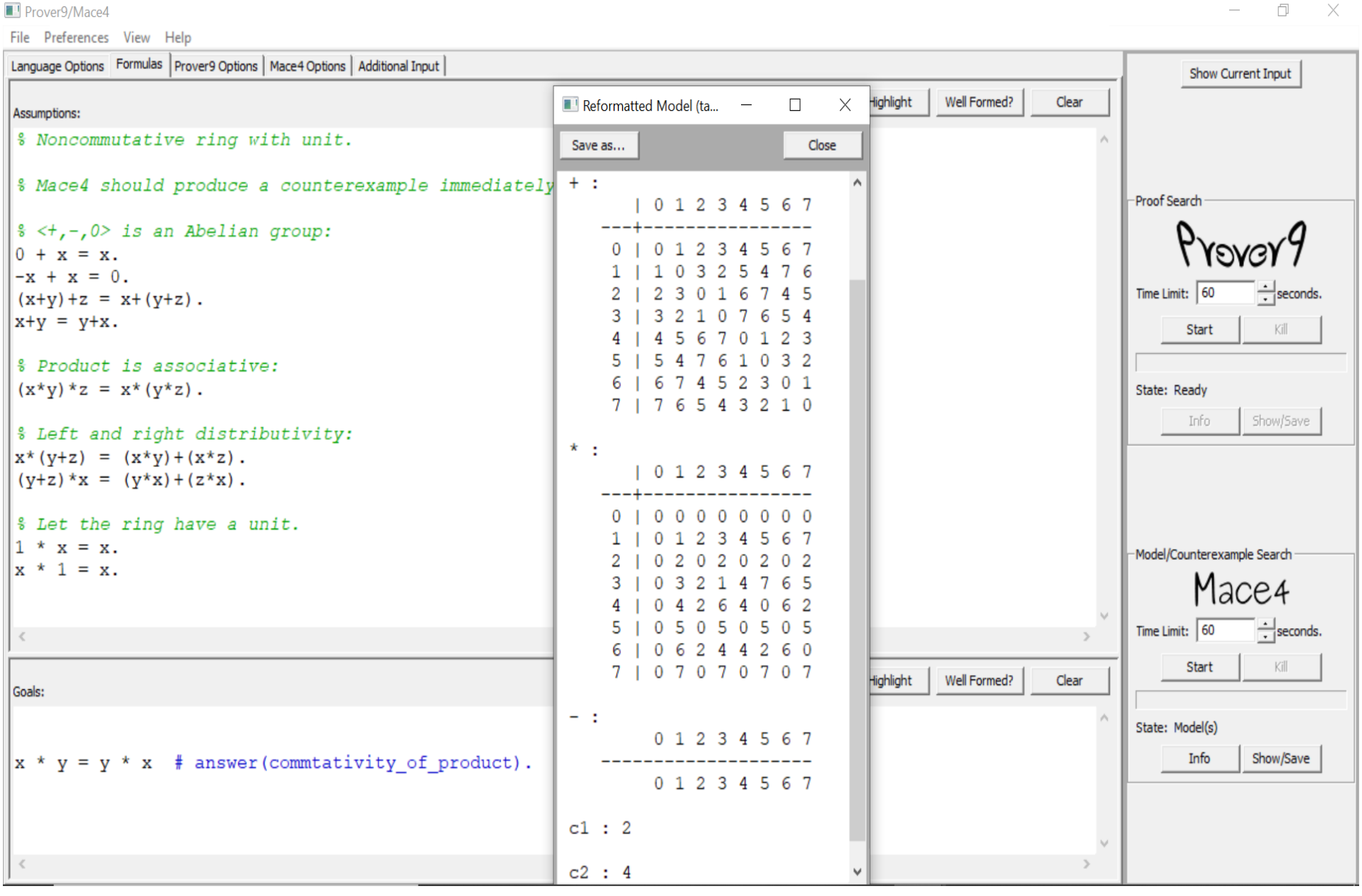
Na závěr k PROLOGu

Programátor není zbaven zodpovědnosti za to, jak bude výpočet probíhat. Výpočet (tedy logické dokazování) je řízen PROLOGovským překladačem a programátor musí znát pravidla (alespoň chce-li psát efektivní programy), kterými se výpočet řídí a v souladu s nimi program v PROLOGu vytvářet.

PROLOG je obzvláště vhodný pro problémy, které zahrnují objekty (zejména strukturované) a vztahy mezi nimi. Silným nástrojem pro vytváření programů v PROLOGu je rekurze. Díky těmto vlastnostem je PROLOG výkonným jazykem pro umělou inteligenci a nečíselné programování obecně.

Programy v jazyku PROLOG se skládají z výrazů, které tvoří fakta a pravidla. Zvláštní výrazy, které nejsou přímou součástí programů, jsou dotazy, někdy nazývané cíle. Programy v PROLOGu slouží k vyjádření (popisu) naší znalostní báze. Programy píšeme v roli „programátorů“, pomocí cílů aktivujeme výpočet, přičemž cíle zadáváme v roli „uživatelů“ vytvořeného programu.

Ukázka programu Prover9&Mace4; informace na přednášce



Obsah

Základní motivace

Klasická logika (VL a PL) nestačí při modelování tzv. vágních tvrzení, například „Petr je silný.“, „Teplota je vysoká.“, „Zákazník je spokojený.“. Uvedená tvrzení často (intuitivně) nepovažujeme za ani úplně nepravdivá, ani úplně pravdivá, tedy za tvrzení, jejichž pravdivostní hodnota leží mezi 0 a 1, třeba je to $0,9$ (skoro pravda), $0,5$ (napůl pravda), $0,1$ (skoro nepravda).
S vágními tvrzeními se setkáváme téměř při každém popisu reálného světa. Jedná se tedy o netriviální a širokou oblast.

Jako první se uvedenou problematikou z pohledu možných aplikací zabýval Lofti A. Zadeh v práci *Fuzzy sets. Information and Control (1965)*.
Fuzzy logika (FL) našla významné uplatnění v praxi a je v současné době intenzívně zkoumána. FL je bohatě rozvinuta jak po stránce komerčně úspěšných aplikací (zejména fuzzy regulátory), tak po stránce teoretických základů.

Struktura pravdivostních hodnot ve FL – základní požadavky

Množinu pravdivostních hodnot budeme značit L. Požadujeme, aby $0,1 \in L$ a aby L byla částečně uspořádána relací $\leq$.
Například tedy $L = [0,1]$; $L = \{0,1\}$ (klasická logika);
$L = \{0,1\} \times \{0,1\}$ (nelineární logika).

Musí existovat operace na L modelující logické spojky (zejména $\otimes$ pro konjunkci a $\rightarrow$ pro implikaci). Tyto operace by měly mít přirozené vlastnosti odpovídající vlastnostem požadovaným po logických spojkách (například komutativita $\otimes$, monotónnost $\otimes$ apod.).
Dále, aby ve FL „dobře fungovalo“ pravidlo MP, je potřeba, aby:
$a \otimes b \leq c \Leftrightarrow a \leq b \rightarrow c$.

Výše uvedené požadavky na strukturu pravdivostních hodnot (a některé neuvedené) vedou k jedné ze základních struktur pravdivostních hodnot ve FL, k tzv. reziduovaným svazům – viz následující definici.

Úplný reziduovaný svaz

Definice

Úplný reziduovaný svaz je struktura $\mathscr{L} = (L; \wedge, \vee, \otimes, \to, 0, 1)$, kde

(1) $(L; \wedge, \vee, 0, 1)$ je úplný svaz (s nejmenším prvkem 0 a největším prvkem 1)

(2) $(L; \otimes, 1)$ je komutativní monoid (tedy $\otimes$ je binární operace na L, která je komutativní, asociativní a platí $a \otimes 1 = a$)

(3) $\otimes, \to$ jsou binární operace na L (nazývané „**násobení**“ a „**reziduum**“) splňující tzv. **podmínku adjunkce**:

$$a \otimes b \leq c \quad \text{právě když} \quad a \leq b \to c.$$

Mezi nejčastěji používané struktury pravdivostních hodnot patří ty, které mají za nosič reálný interval $[0,1]$ s přirozeným uspořádáním, tedy $a \wedge b = \min(a,b)$, $a \vee b = \max(a,b)$. Na nich se používají tři páry adjungovaných operací $\otimes$ a $\to$:

(I) **Łukasiewiczovy operace:**
$a \otimes b = \max(a+b-1,0), \qquad a \to b = \min(1-a+b,1)$

(II) **Gödelovy operace:**
$$a \otimes b = \min(a,b), \qquad a \to b = \begin{cases} 1, & \text{pro} \quad a \leq b, \\ b, & \text{jinak} \end{cases}$$

(III) **součinové operace:**
$$a \otimes b = a \cdot b, \qquad a \to b = \begin{cases} 1, & \text{pro} \quad a \leq b, \\ b/a, & \text{jinak.} \end{cases}$$

Poznámka: V reziduovaném svazu definujeme některé odvozené operace. Mezi nejdůležitější patří **negace** ($\neg$) a tzv. **bireziduum** ($\leftrightarrow$) definované následovně: $\neg a = a \to 0$, $a \leftrightarrow b = (a \to b) \wedge (b \to a)$.

Poznámka: Je-li $L = \{0, 1\}$, $\otimes$ je operace klasické konjunkce a $\rightarrow$ je operace klasické implikace, pak příslušný reziduovaný svaz (ve kterém je uspořádání dáno vztahem $0 \leq 1$) je svazem pravdivostních hodnot klasické logiky.

Poznámka: Reziduované svazy jsou bohaté struktury, které kromě Booleových algeber zahrnují také například Heytingovy algebry, MV-algebry, algebry lineární logiky a další významné algebraické struktury.

Nejúspěšnějšími aplikacemi FL jsou tzv. **fuzzy regulátory** a tzv. **pravidlové fuzzy systémy**. Ty nalezly zejména v Japonsku (začátkem 90. let) rozsáhlé komerční uplatnění.[5]

Vybrané aplikace pravidlových fuzzy systémů a fuzzy regulátorů:

- spotřební elektronika (fuzzy pračka, fuzzy myčka, fuzzy vysavač, fuzzy kamera apod.)
- řízení metra v japonských městech (fuzzy regulátor zajišťuje plynulé rozjíždění a brzdění, lépe než člověk; nižší spotřeba energie)
- řízení velké průmyslové helikoptéry ovládané hlasem (tuto úlohu se klasickými metodami nepodařilo vyřešit)
- řízení velkých průmyslových systémů (například pece)
- fotoaparát s automatickým vyhledáváním centrálního bodu pro zaostření (Minolta)

[5] Důvodem úspěchu nebyla blízkost fuzzy přístupu k východnímu myšlení, ale povaha japonského trhu, tedy vstřícnost k novinkám, nízká cena senzorů a koncepční jednoduchost.

Použití fuzzy technologie (pokračování):

- ABS, řízení motoru, volnoběhu a klimatizace (Honda, Nissan, Subaru)
- řízení výtahů (Mitsubishi)
- korekce chyb ve slévárenských zařízeních na plastické výrobky (Omron)
- palmtop Kanji určený pro rozpoznávání ručně psaných textů
- rozpoznávání řeči
- fuzzy SQL (Omron)
- pomoc při hledání identifikačních a profilových systémů pachatele (velký, ne příliš těžký, víceméně starý, . . .)
- analýza portfolia při investování na kapitálovém trhu
- efekty ve filmech (například Lord of the Rings)
- jazykové filtry na zprávy s nechtěným obsahem textu (třeba v chatovacích místnostech)
- fuzzy technologie se používají i v některých mikroprocesorech.

Poznamenejme, že hlavní aplikace FL tvoří fuzzy relační modelování, tedy modelování pomocí fuzzy relací. Kromě zmíněných pravidlových fuzzy systémů se jedná o

- rozhodování
- vyhledávání
- shlukování, rozpoznávání.

Další zajímavou oblastí aplikací FL je fuzzy logické programování. Jde o rozšíření klasického logického programování o principy fuzzy modelování (zejména: databáze faktů může obsahovat fakt s nějakým stupněm pravdivosti, například fakt JEUSPORNY(OCTAVIA19TDI) se stupněm 0,9, fakt JEUSPORNY(OCTAVIA14MPI) se stupněm 0,3). Problematika se rozvíjí.

Poznámka: paradox hromady

Paradox hromady

- 1 000 000 zrnek písku tvoří hromadu.
- Odebráním jednoho zrnka písku hromada nepřestane být hromadou.
- Tedy 999 999 zrnek písku tvoří hromadu.
- Tedy 999 998 zrnek písku tvoří hromadu.

  ...
- Tedy 0 zrnek písku tvoří hromadu.

Klasická logika tento paradox[6] uspokojivě nevyřeší. Má jen dvě pravdivostní hodnoty: 0 a 1. Klasická logika tak vynucuje „ostrý skok“: libovolné množství zrnek písku buď hromadou je, nebo není. Ale proč by třeba 50 000 zrnek písku mělo být hromadou a 49 999 už ne?

[6] Paradox sóritės (z řeckého sóros = hromada) je připisován Eubúlidovi z Milétu. Byl současníkem Aristotela a patřil do tzv. megarské školy.

Poznámka: řešení paradoxu hromady

FL umožňuje pozvolné snižování stupňů pravdivosti.
Více o tom na přednášce.

Ve FL lze definovat to, že odebrání jednoho zrnka písku z hromady nepatrně sníží „stupeň hromadovitosti“ (například o 0,000 001). Při použití tohoto přístupu, v souladu s praxí, nula zrnek písku netvoří hromadu.

Varianty paradoxu hromady:

- Paradox holohlavého s předpokladem, že vypadnutím jednoho vlasu se vlasatý člověk nestane plešatým.
- Které malé přirozené číslo je největší?
- Kde přesně na obrázku je žlutá barva?

Obsah

Výchozí motivace k zavedení ML

Klasická logika nemá prostředky k formalizaci tvrzení obsahujících modality, například „je možné, že . . . “, „je nutné, že . . . “. Rozšíření klasické logiky, kde toto je možné se nazývá **modální logika**. ML našla uplatnění například ve formalizaci znalostních systémů a systémů, které pracují s časem.

Základy syntaxe a sémantiky ML

Zaměříme se na výrokovou ML. Oproti jazyku klasické VL obsahuje jazyk ML navíc unární spojky $\Box$ a $\Diamond$ ($\Box\varphi$ se čte „je nutné, že φ“ a $\Diamond\varphi$ se čte „je možné, že φ“). Definice formule se příslušným způsobem rozšíří, tedy přidáme pravidlo „je-li φ formule, jsou i $\Box\varphi$ a $\Diamond\varphi$ formule“. Poznamenejme, že konkrétní význam $\Box\varphi$ může být „je známo, že φ“, „věří se, že φ“, „vždy v budoucnosti bude platit φ“ apod.
Sémantika ML je založena na pojmu **možný svět**. Možný svět je obecná kategorie (v jednom možném světě může v daný okamžik pršet, ve druhém ne, apod.), která má řadu interpretací. Možné světy mohou být časové okamžiky; mohou reprezentovat názory jednotlivých expertů (co možný svět, to expert) apod. Speciální interpretací světů získáme **logiku času** (**temporální logika**), což je logika zabývající se tvrzeními, jejichž pravdivostní hodnota závisí na čase. **Logika znalostí** (**epistemická logika**) se zabývá spojkami „ví se, že . . . “, „věří se, že . . . “ apod.

Příklad s karetní hrou Taroky

První hráč zná své karty a vidí následující zdvih. Co z toho může usoudit, pokud v pravidlech je, že hráč musí ctít barvu a pokud danou barvu nemá, musí zahrát trumf (tarok)?

- **Nutně** první hráč má dva taroky.
- **Nutně** pouze první hráč má ještě srdce.
- **Nutně** třetí hráč nemá taroky.
- **Možná** druhý hráč má alespoň jeden tarok.
- **Možná** čtvrtý hráč má alespoň jeden tarok.
- **Možná** třetí hráč má piky.
- ...

Definice

Kripkeho struktura pro výrokovou ML je trojice $\mathbf{K} = \langle W, e, r \rangle$, kde $W \neq \emptyset$ je množina možných světů, e je zobrazení přiřazující každému $w \in W$ a každému výrokovému symbolu p pravdivostní hodnotu $e(w, p)$ (p je/není pravdivé ve w), $r \subseteq W \times W$ je relace dosažitelnosti ($\langle w, w' \rangle \in r$ znamená, že z w je možné se dostat do w').

Pro $r = W \times W$ (každý svět je dosažitelný z každého) se příslušná logika nazývá **logika znalostí.** Platí-li pro nějaké $W' \subseteq W$, že $r = W \times W'$, nazývá se příslušná logika **logika domění**.

Definujme nyní pravdivostní hodnotu $\| \varphi \|_{\mathbf{K},w}$ formule φ v $\mathbf{K}$ v možném světě w takto: $\| p \|_{\mathbf{K},w} = e(w,p)$ (pro výrokový symbol p); $\| \varphi \wedge \psi \|_{\mathbf{K},w} = 1$, právě když $\| \varphi \|_{\mathbf{K},w} = 1$ a $\| \psi \|_{\mathbf{K},w} = 1$ (tedy jako v klasické logice) a podobně pro ostatní výrokové spojky; $\| \Box\varphi \|_{\mathbf{K},w} = 1$, právě když $\| \varphi \|_{\mathbf{K},v} = 1$ pro každý $v \in W$, pro který $\langle w, v \rangle \in r$ (tedy „je nutné, že φ“ znamená, že φ je pravdivá v každém možném světě dosažitelném z w); $\| \Diamond\varphi \|_{\mathbf{K},w} = 1$, právě když $\| \varphi \|_{\mathbf{K},v} = 1$ pro nějaký $v \in W$, pro který $\langle w, v \rangle \in r$ (tedy „je možné, že φ“ znamená, že φ je pravdivá v nějakém možném světě dosažitelném z w). Poznamenejme ještě, že $\Box\varphi$ a $\neg\Diamond\neg\varphi$ (a $\Diamond\varphi$ a $\neg\Box\neg\varphi$) mají stejné pravdivostní hodnoty. Stačí proto zavést jen jednu spojku a druhou brát jako odvozenou.

Poznámka: Je-li r reflexivní, tranzitivní a platí-li, že pro každé $v, w \in W$ je $\langle v, w \rangle \in r$ nebo $\langle w, v \rangle \in r$, pak se odpovídající logika nazývá logika času (temporální logika) a $w \in W$ se chápou jako časové okamžiky. (Existují i jiné systémy logiky času.)
Pak tedy $\| \Box\varphi \|_{\mathbf{K},w} = 1$ znamená, že φ je pravdivá ve všech časových okamžicích počínaje w. Pro r^{-1} pak $\| \Box\varphi \|_{\mathbf{K},w} = 1$ znamená, že φ je pravdivá ve všech časových okamžicích až po w. Podobně pro $\| \Diamond\varphi \|_{\mathbf{K},w} = 1$.

Obsah

Přirozená čísla

Jsou to čísla $1, 2, 3, 4, 5, 6, \ldots$.
Množinu všech přirozených čísel označujeme $\mathbb{N}$.

Celá čísla

Jsou to čísla $0, 1, -1, 2, -2, 3, -3, 4, -4, 5, -5, \ldots$.
Množinu všech celých čísel značíme $\mathbb{Z}$.

Racionální čísla

Jsou to čísla, která lze vyjádřit ve tvaru zlomku $\frac{m}{n}$, kde m je celé číslo a n je přirozené číslo. Množinu všech racionálních čísel označujeme $\mathbb{Q}$. Racionální čísla jsou tedy například $\frac{1}{3}$, $\frac{-2}{5}$, $\frac{21}{37}$, $\frac{-6}{12}$ atd. Poznamenejme, že $\frac{1}{3}$, $\frac{2}{6}$, $\frac{6}{18}$ jsou různé zápisy téhož racionálního čísla. Racionální čísla zapisujeme také pomocí tzv. desetinného rozvoje. Třeba číslo $\frac{3}{2}$ zapisujeme jako $1,5$. Číslo $\frac{1}{3}$ má tzv. nekonečný desetinný rozvoj a je jím $0,3333\ldots$, což také zapisujeme jako $0,\overline{3}$.

Věta

$\sqrt{2} \notin \mathbb{Q}$.

Důkaz: Sporem, na cvičení.

Poznámka: Víme, že $\mathbb{N}$, $\mathbb{Z}$ a $\mathbb{Q}$ jsou spočetné množiny.

Reálná čísla

Jsou to všechna čísla, která se nacházejí na číselné (reálné) ose. (Každý bod reálné osy odpovídá právě jednomu reálnému číslu.) Množinu všech reálných čísel označujeme $\mathbb{R}$. Kromě racionálních čísel zahrnují reálná čísla i tzv. **čísla iracionální**. To jsou reálná čísla, která nejsou racionální. Příkladem iracionálních čísel jsou $\sqrt{2}$, $\sqrt{3}$, π, e. Každé iracionální číslo má nekonečný neperiodický desetinný rozvoj. Množinu všech iracionálních čísel označujeme $\mathbb{I}$.

Zřejmě: $\mathbb{Q} \cup \mathbb{I} = \mathbb{R}$ a $\mathbb{Q} \cap \mathbb{I} = \emptyset$.

Poznámka: Víme jak dokázat, že $\mathbb{I}$ a $\mathbb{R}$ jsou nespočetné množiny.

Komplexní čísla

Množinu všech uspořádaných dvojic reálných čísel $\langle a, b\rangle$ zapisovaných obvykle ve tvaru $a+bi$, kde symbolem i označujeme tzv. imaginární jednotku, pro niž platí $i^2 = -1$, nazýváme komplexní čísla; značíme $\mathbb{C}$. Zásluhou K. F. Gausse se od roku 1831 znázorňují komplexní čísla v rovině.
Každé komplexní číslo $z = a+bi$ můžeme vyjádřit i v tzv. goniometrickém tvaru $z = r(\cos\varphi + i\sin\varphi)$, kde číslo $r = \sqrt{a^2+b^2}$ je tzv. absolutní hodnota a úhel φ argument komplexního čísla.

Poznámka: Zřejmě $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$.

Obsah

Tvrzení

Pro každé $n \in \mathbb{N}$, $n \geq 5$ platí: $\ln n < n < n \cdot \ln n < n^2 < 2^n < n!$.

Důkaz: Tvrzení dokážeme indukcí. Pro $n = 5$ dostáváme: $\ln 5 \doteq 1,61 < 5 < 5 \cdot \ln 5 \doteq 8,05 < 25 < 32 < 120$ a tedy základní krok indukce platí. Předpokládejme dále, platnost indukčního předpokladu: pro libovolné $n \in \mathbb{N}$, $n \geq 5$ platí: $\ln n < n < n \cdot \ln n < n^2 < 2^n < n!$. Ukažme (komentář na přednášce), že pak bude platit

$$\ln(n+1) < n+1 < (n+1) \cdot \ln(n+1) < (n+1)^2 < 2^{n+1} < (n+1)!.$$

To ale platí, protože $\ln(n+1) < \ln(en) = \ln n + 1 < n+1 < (n+1) \cdot \ln(n+1) < (n+1)^2 < 2 \cdot n^2 < 2 \cdot 2^n = 2^{n+1} = 2 \cdot 2^n < (n+1) \cdot 2^n < (n+1) \cdot n! = (n+1)!$.

Poznámka. Analogicky bychom mohli dokázat podobná tvrzení, například, že pro každé $n \in \mathbb{N}$, $n \geq 25$ platí:

$$\log n < n < n \cdot \log n < n^{10} < 10^n < n!.$$

Tedy i v tomto případě platí pro nekonečně mnoho přirozených čísel, že (dekadický) logaritmus „roste pomaleji“ než funkce lineární, která „roste pomaleji“ než funkce lineárně logaritmická, která „roste pomaleji“ než funkce polynomická (n^{10}) a ta „roste pomaleji“ než funkce exponenciální (se základem 10) a ta „roste pomaleji“ než faktoriál. K rychlostem růstu se ještě vrátíme a význam pojmu „roste pomaleji“ definujeme přesně.

Dá se dokázat, že od jistého konkrétního přirozeného čísla n_0 bude pro všechna přirozená čísla $n \geq n_0$ (tedy pro nekonečně mnoho přirozených čísel) platit, že

$$k < \log_a n < n < n \cdot \log_b n < n^c < d^n < n!,$$

přičemž $a, b, c, d \in \mathbb{N} \setminus \{1\}$ a konstanta $k \in \mathbb{Z}$.

Obsah

Dělitelnost na množině celých čísel

Definice

Pro $m, n \in \mathbb{Z}$ říkáme, že m **dělí**[a] n, píšeme $m \mid n$, právě když $\exists k \in \mathbb{Z}$ tak, že $m \cdot k = n$. Fakt, že m **nedělí** n zapisujeme $m \nmid n$.

[a] Když $m \mid n$, říkáme také, že m **je dělitelem** n nebo n **je dělitelné** m, nebo n **je násobek** m.

Příklad

Zřejmě $5 \mid 15$ a $-6 \mid -54$. Dále třeba $2 \nmid 3$ a $3 \nmid 2$.

Příklad

Dokažte indukcí, že pro $\forall n \in \mathbb{N}$ platí: $7 \mid (2^{n+2} + 3^{2n+1})$.

Řešení: Zkuste sami.

Vlastnosti binární relace dělitelnosti

Věta

Pro $a, b, c \in \mathbb{Z}$ platí:

a) $a \mid a, \quad 1 \mid b, \quad c \mid 0.$

b) Jestliže $a \mid b$ a $b \mid c$, pak $a \mid c$.

c) Jestliže $a \mid b$ a $a \mid c$, pak pro každé $x, y \in \mathbb{Z}$ platí $a \mid (bx + cy)$.

Důkaz: a), b) viz přednáška, c) na cvičení.

Věta

Relace dělitelnosti $\mid$ je uspořádání na množině $\mathbb{N}$.

Důkaz: To, že je $\mid$ na $\mathbb{N}$ reflexivní a tranzitivní plyne z předchozí věty. Zbývá ověřit antisymetrii na $\mathbb{N}$.
Závěr důkazu na přednášce.

Příklad s relací dělitelnosti na konečných podmnožinách $\mathbb{N}$

Příklad

Pomocí Hasseova diagramu znázorněte množinu všech přirozených dělitelů čísel 30 a 24.

Řešení: 24 viz přednáška, 30 na cvičení.

Prvočísla

Definice

Přirozené číslo p se nazývá **prvočíslo**, jestliže $p \neq 1$ a jestliže p je dělitelné jen a pouze čísly 1 a p. **Složené číslo** je každé přirozené číslo $n \geq 2$, které není prvočíslem.[a]

[a] Číslo 1 není ani prvočíslo, ani číslo složené.

Příklad

Vypište seznam 31 nejmenších prvočísel.

Řešení: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127.

Věta

Existuje nekonečně mnoho prvočísel.

Důkaz: Sporem, na přednášce.

Základní věta aritmetiky

Každé přirozené číslo větší než jedna lze vyjádřit jednoznačně až na pořadí činitelů jako součin prvočísel.[a]

[a] V tomto případě je třeba u prvočísel položit součin jednoho čísla roven tomuto číslu.

Poznámka: Prvočísla tak lze považovat za „základní kameny“, ze kterých lze „vystavět“ libovolné složené[7] číslo pomocí operace násobení. ZVA říká, že každé přirozené číslo větší než 1 je buď prvočíslem, nebo je možné jej zapsat jako součin dvou či více prvočísel. Takový rozklad je přitom jednoznačně dán až na pořadí činitelů.

Pro libovolné číslo $2 \leq n \in \mathbb{N}$ je $n = p_1^{k_1} \cdot p_2^{k_2} \cdot p_3^{k_3} \cdot \cdots \cdot p_m^{k_m}$, kde $p_1 < p_2 < p_3 < \cdots < p_m$ jsou prvočísla a $k_1, k_2, k_3, \ldots, k_m \in \mathbb{N}$.

[7] Je-li n složené číslo, pak existuje prvočíslo $p \leq \sqrt{n}$ takové, že $p \mid n$.

Příklad

Na prvočinitele rozložte čísla 2100, 19 a 1519344902621.

Řešení: Daná čísla vyjádříme ve tvaru součinu prvočísel. Máme:

- $2100 = 2^2 \cdot 3^1 \cdot 5^2 \cdot 7^1$
- $19 = 19^1$
- $1519344902621 = 7^2 \cdot 13^1 \cdot 23^1 \cdot 31^3 \cdot 59^2$.

Důkaz ZVA je pro prvočíslo triviální (není co dokazovat). Pro složená čísla dokážeme nejprve existenci prvočíselného rozkladu (s využitím silné matematické indukce) a poté dokážeme jednoznačnost tohoto rozkladu (s využitím důkazu sporem).

K důkazu základní věty aritmetiky

K důkazu ZVA:

Důkaz se provádí matematickou indukcí.

- Existence. Tvrzení zřejmě platí pro každé prvočíslo p, tedy i pro číslo 2, čímž je ověřena platnost základního kroku indukce. Indukčním předpokladem je, že tvrzení platí pro všechna přirozená čísla od 2 až do $n-1$. Pak n je buď prvočíslo (a tvrzení platí) nebo je n složené číslo. Pak ale musí existovat přirozená čísla u, v taková, že $n = u \cdot v$, kde $1 < u < n$, $1 < v < n$. Na základě indukčního předpokladu mají čísla u a v prvočíselný rozklad. Odtud již jednoduše vyplývá existence prvočíselného rozkladu i pro jejich součin, pro číslo n.
- Jednoznačnost. Dokažme sporem, že rozklad složeného čísla n je jednoznačný (pro prvočíslo je to zřejmé). Pokud by pro n existovaly dva různé rozklady, pak by musely existovat dva různé rozklady také pro menší (výše uvedená) čísla u a v, což by bylo ve sporu s indukčním předpokladem.

Obsah

V této části se budeme zabývat hledáním největšího společného dělitele (NSD) a nejmenšího společného násobku (NSN) dvou přirozených čísel. Dostaneme se k algoritmickému hledání největšího společného dělitele dvou přirozených čísel – k tzv. Euklidovu algoritmu (EA).

Poznámka: Hledání NSD lze realizovat i na jiných číselných oborech, například na množině celých čísel. Také lze celkem snadno zobecnit hledání NSD a NSN pro více než dvě čísla. My to ale potřebovat nebudeme.

Poznámka: Euklidův algoritmus je tradičním prototypem algoritmického postupu.[8] Má velmi bohatou historii. Objevuje se nejen při hledání NSD, ale take při řešení neurčitých rovnic (kde se využívá Bézoutova identita) a třeba také ve Sturmově metodě při hledání počtu reálných kořenů algebraických rovnic. Velice úzce souvisí s řetězovými zlomky.

---

[8] Donald Knuth v 1. díle The Art of Computer Programming uvedl následující: „Počínaje rokem 1950 je slovo algoritmus spojeno s EA“.

NSD dvou přirozených čísel

Definice

Nechť $x, y \in \mathbb{N}$. Číslo $d \in \mathbb{N}$ nazveme **společný dělitel** čísel x a y, jestliže $d \mid x$ a $d \mid y$. Největší číslo z množiny všech společných dělitelů čísel x a y označíme $\mathrm{NSD}(x, y)$ a nazveme jej **největší společný dělitel** x a y.

Poznámka. Pokud je $\mathrm{NSD}(x, y) = d$ a d' je libovolný společný dělitel přirozených čísel x a y, pak $d' \mid d$.

Definice

Čísla $x, y \in \mathbb{N}$ nazveme **nesoudělná**, jestliže $\mathrm{NSD}(x, y) = 1$.
V případě, že $\mathrm{NSD}(x, y) \neq 1$ řekneme, že jsou x a y **soudělná**.

Příklad

- Čísla 12 a 77 jsou nesoudělná, neboť $\mathrm{NSD}(12, 77) = 1$.
- Čísla 24 a 84 jsou soudělná. K jejich společným dělitelům patří čísla $1, 2, 3, 4, 6, 12$, přičemž zřejmě $\mathrm{NSD}(24, 84) = 12$.

NSN dvou přirozených čísel

Následuje duální definice k pojmům společný dělitel a NSD.

Definice

Nechť $x, y \in \mathbb{N}$. Číslo $n \in \mathbb{N}$ nazveme **společný násobek** čísel x, y, jestliže $x \mid n$ a $y \mid n$. Nejmenší číslo z množiny všech společných násobků čísel x a y označíme $\mathrm{NSN}(x, y)$ a nazveme jej **nejmenší společný násobek** x a y.

Poznámka. Pokud je $\mathrm{NSN}(x, y) = n$ a n' je libovolný společný násobek přirozených čísel x a y, pak $n \mid n'$.

Příklad

Čísla 10 a 15 mají nekonečně mnoho společných násobků, například: $30, 60, 90, 360, 3\,870, 936\,333\,120$, přičemž zřejmě $\mathrm{NSN}(10, 15) = 30$.

Souvislost mezi NSD a NSN

Následující věta dává do souvislosti NSD a NSN. Vyplývá z ní, jak spočítat nejmenší společný násobek dvou přirozených čísel, známe-li jejich největší společný dělitel.

Věta

Nechť $x, y \in \mathbb{N}$. Pak $\mathrm{NSD}(x,y) \cdot \mathrm{NSN}(x,y) = x \cdot y$.

Důkaz: Vynecháme.

Příklad

Čísla 100 a 40 mají největší společný dělitel roven 20. Podle poslední věty je zřejmě $\mathrm{NSN}(100,40) = \frac{100 \cdot 40}{20} = 200$.

Věta o jednoznačnosti dělení se zbytkem

Pro $a, b \in \mathbb{Z}$, $b \neq 0$, existují jednoznačně určená $q, r \in \mathbb{Z}$ tak, že $a = bq + r$ a $0 \leq r < b$.

Číslo r se nazývá **zbytek** po celočíselném dělení čísla a číslem b. Číslu q říkáme **částečný podíl**. Píšeme $\mathbf{a} \bmod \mathbf{b} = \mathbf{r}$.

Důkaz: Vynecháme. (Provádí se ve dvou krocích: ověří se existence čísel q a r a pak jejich jednoznačnost.)

Příklad

Zřejmě $20 \bmod 6 = 2$ a třeba $-20 \bmod 6 = 4$.

Poznámka k převodům mezi číselnými soustavami

Věta o jednoznačnosti zápisu přirozeného čísla v soustavě o základu b

Nechť $b > 1$ je přirozené číslo. Pro každé $x \in \mathbb{N}$ existují jednoznačně určená čísla $a_n, a_{n-1} \ldots, a_1, a_0$, přičemž $0 \leq a_i < b$, $a_n \neq 0$ tak, že $x = a_n b^n + a_{n-1} b^{n-1} + \cdots + a_1 b + a_0$.

Poznámka: Pro zápis čísel ve dvojkové soustavě ($b = 2$) používáme symboly 0 a 1. Pro zápis čísel v šestnáctkové soustavě používáme symboly $0, 1, \ldots, 9, A, B, C, D, E, F$.

Příklad

Převeďte z desítkové soustavy číslo 28 do soustavy trojkové a patnáctkové.[a]

Řešení: $(1001)_3 = (28)_{10} = (1D)_{15}$

[a] Znáte algoritmus, využívající dělení se zbytkem, který efektivně hledá zápis vybraného přirozeného čísla v soustavě o základu b.

Poznámky k vybraným aplikacím počítání modulo *n*

(A) Od roku 1954 jsou **rodná čísla** v ČR uváděna jako deseticiferná čísla ve tvaru *rrmmddxxxy*. Z cifer *rr* lze poznat rok, z cifer *mm* měsíc (u žen je přičtena hodnota 50) a z cifer *dd* den narození dané osoby. Cifry *xxx* souvisí s pořadovým číslem narození dítěte v dané oblasti v konkrétní den. Poslední cifra *y* slouží jako kontrolní, jde o zbytek (modulo) po celočíselném dělení devítimístného čísla jedenácti. Výjimečně bylo toto pravidlo porušeno, v takovém případě je poslední cifrou vždy nula.

(B) Podobně desetimístný **ISBN kód** je navržen tak, aby byl snadno detekovatelný překlep v jedné cifře. Prvních devět cifer identifikuje jazyk, nakladatele a číslo knihy dle katalogu nakladatele. Poslední cifra dává zbytek po dělení prvních devíti cifer jedenácti. Vyjde-li zbytek 10 umístí se na poslední pozici symbol X.

(C) **Hašovací funkce** se aplikují pro rychlejší prohledávání tabulek a pro porovnávání dat. V kryptografii jsou používány zejména pro vytváření a ověřování digitálního podpisu, zajištění integrity dat a ochranu uložených hesel. Funkce $f(x) = x \bmod n$ je příkladem jednoduché hašovací funkce převádějící číselná data na relativně (vzhledem k velikosti n) malé číslo. Výstupem hašovací funkce je tzv. hash (haš).

(D) Rychlé generátory pseudonáhodných čísel efektivně generují posloupnost čísel, obtížně rozeznatelnou od posloupnosti náhodné. Používají se v moderní kryptografii i pro počítačové hry. Tzv. **lineární kongruentní generátor** je definován vztahem: $x_{i+1} = (ax_i + c) \bmod n$, kde a, c, n jsou vhodně zvolené konstanty; například $a = 1\,664\,525$, $c = 1\,013\,904\,223$, $n = 2^{32}$. Výchozí hodnotě x_0 se říká náhodné semínko. Správně nastavený generátor generuje pseudonáhodně všechna celá čísla s rovnoměrným rozložením v rozsahu $0 \leq x_i < n$.

Tvrzení, o které se opírá Euklidův algoritmus

Tvrzení

Nechť $a, b \in \mathbb{N}$ a nechť $r = a \bmod b$. Pak NSD(a, b) =NSD(b, r).

Důkaz: Stačí ukázat, že množina společných dělitelů a, b se rovná množině společných dělitelů b, r. Jsou-li totiž tyto množiny stejné, musí se rovnat i jejich největší prvky, což potřebujeme dokázat.

Dokažme tedy následující ekvivalenci: $d \in \mathbb{N}$ je společný dělitel a, b, právě když je to společný dělitel b, r.

Z rovnosti $r = a \bmod b$ víme, že $a = bq + r$, kde $q, r \in \mathbb{Z}$, přičemž $0 \leq r < b$.

$(\Rightarrow)$: Z toho, že $d \mid a$, $d \mid b$ máme dokázat, že $d \mid b$ a $d \mid r$. To je snadné, viz přednáška.

$(\Leftarrow)$: Z toho, že $d \mid b$, $d \mid r$ máme dokázat, že $d \mid a$ a $d \mid b$. To je také snadné, viz přednáška.

Předchozí tvrzení je pro fungování EA klíčové. Namísto hledání NSD pro čísla $a, b \in \mathbb{N}$, $a > b$[9] lze hledat NSD pro odpovídající čísla $b > r_1$, kde $r_1 = a \bmod b$. Tvrzení však můžeme aplikovat znovu, tentokrát s výchozími přirozenými čísly b, r_1, $b > r_1$. Bude platit, že $\mathrm{NSD}(b, r_1) = \mathrm{NSD}(r_1, r_2)$, kde $r_2 = b \bmod r_1$. Takto lze pokračovat dále. Uvedený postup nám postupně výchozí úlohu zjednodušuje, neboť se počítá se stále menšími čísly: $a > b > r_1 > r_2 > \cdots > r_n > r_{n+1} = 0$. Konečnost tohoto postupu je zaručena, protože se po každé aplikaci tvrzení obě čísla zmenší, přičemž ale nemohou být záporná. Postup se tak vždy zastaví (a to právě ve chvíli, kdy se menší z obou čísel bude rovnat nule). NSD tak bude roven poslednímu nenulovému zbytku po dělení. V našem značení tedy $\mathrm{NSD}(a, b) = \mathrm{NSD}(b, r_1) = \mathrm{NSD}(r_1, r_2) = \cdots = \mathrm{NSD}(r_n, 0) = r_n$. Je třeba ale dodefinovat, že $\mathrm{NSD}(x, 0) = x$ pro každé $x \in \mathbb{N}$.[10]

[9] Zřejmě $\mathrm{NSD}(a, a) = a$. Pokud by $b > a$, tak čísla a, b „prohodíme“. „Prohození“ se opírá o fakt, že $\mathrm{NSD}(a, b) = \mathrm{NSD}(b, a)$.

[10] Toto rozšíření je v dobrém souladu s naší definicí NSD pro dvě přirozená čísla. Vskutku x je zřejmě společný dělitel čísel x a 0 na $\mathbb{N}_0$, neboť $x \mid x$ a $x \mid 0$. Zároveň je i největším číslem, které současně dělí x a nulu.

Euklidův algoritmus

Euklidův algoritmus je využíván k nalezení NSD. Je považován za jeden z nejstarších algoritmů. Byl zapsán řeckým matematikem Euklidem v jeho knize Základy.

Věta – Euklidův algoritmus na $\mathbb{N}$

Nechť $a, b \in \mathbb{N}$, $b \leq a$. Pak v $\mathbb{N}$ existují prvky $q_0, q_1, \ldots q_n, r_1, r_2, \ldots, r_n$, přičemž $r_n = \text{NSD}(a, b)$ a

$$
\begin{aligned}
a &= bq_0 + r_1, \quad r_1 = a \bmod b \\
b &= r_1 q_1 + r_2, \quad r_2 = b \bmod r_1 \\
&\vdots \\
r_{i-1} &= r_i q_i + r_{i+1}, \quad r_{i+1} = r_{i-1} \bmod r_i \\
&\vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n, \quad r_n = r_{n-2} \bmod r_{n-1} \\
r_{n-1} &= r_n q_n.
\end{aligned}
$$

Příklad na výpočet NSD a NSN

Příklad na EA

Určete NSD čísel 924 a 8190.

Řešení: Podle Euklidova algoritmu postupně obdržíme následující rovnosti:

$$\begin{aligned} 8190 \bmod 924 &= 798 \\ 924 \bmod 798 &= 126 \\ 798 \bmod 126 &= 42 \\ 126 \bmod 42 &= 0. \end{aligned}$$

Posledním nenulovým zbytkem je číslo 42. Z provedených výpočtů plyne, že $\mathrm{NSD}(924, 8190) = 42$.

Poznámka: Snadno dopočítáme, že $\mathrm{NSN}(924, 8190) = \frac{924 \cdot 8190}{42} = 180\,180$.

Euklidův algoritmus – zdrojový kód v Ruby[11]

Euklidův algoritmus pro nalezení NSD přirozených čísel a, b:

verze bez rekurze:

```
def nsd(a, b)
  while (b != 0) do
    c = b
    b = a % b
    a = c
  end
  return a
end
```

rekurzivní verze:

```
def nsd(a, b)
  return a if (b == 0)
  return nsd(b, a % b)
end
```

[11] Ruby je objektově orientovaný interpretovaný skriptovací programovací jazyk s jednoduchou syntaxí. Autorem je Jukihiro Macumoto. První verze byla uveřejněna v roce 1995.

Poznámky k výpočtu EA

Poznámka: Podle Lamého věty, jsou-li $a, b \in \mathbb{N}$, $a > b$, vyžaduje Euklidův algoritmus pro nalezení NSD(a, b) nanejvýš tolik kroků, kolik je pětkrát počet cifer v b.

Poznámka: V jednotlivých krocích EA je třeba vypočítat částečný podíl q a zbytek r. Ve více jak v polovině případů je $q \in \{1, 2\}$.

Poznámka: Nechť $a, b \in \mathbb{N}$, $a > b$. Pak platí následující:

- jsou-li a, b sudá, pak NSD$(a, b) = 2 \cdot$NSD$(\frac{a}{2}, \frac{b}{2})$
- je-li a sudé a b liché, pak NSD$(a, b) =$NSD$(\frac{a}{2}, b)$
- je-li a liché a b sudé, pak NSD$(a, b) =$NSD$(a, \frac{b}{2})$
- jsou-li a, b lichá, pak NSD$(a, b) =$NSD$(a - b, b)$.

Bézoutova rovnost (Bézoutova věta)

Věta – Bézoutova rovnost

Nechť $a, b \in \mathbb{N}$. Pak existují[a] čísla $x, y \in \mathbb{Z}$ taková, že $\mathrm{NSD}(a, b) = ax + by$.

---

[a] Je-li řešením Bézoutovy rovnosti dvojice koeficientů $\langle x, y \rangle$, pak existuje nekonečně mnoho dalších dvojic koeficientů (řešících Bézoutovu rovnost) ve tvaru $\langle x + \frac{kb}{d}, y - \frac{ka}{d} \rangle$, kde $k \in \mathbb{Z}$ a $d = \mathrm{NSD}(a, b)$.

Bézoutova rovnost říká, že NSD dvou přirozených čísel a, b lze zapsat jako lineární kombinaci těchto dvou čísel, přičemž koeficienty jsou celá čísla. Tyto tzv. **Bézoutovy koeficienty** lze určit rozšířeným Eukleidovým algoritmem. Jde o klasický EA, ve kterém se navíc uchovávají Bézoutovy koeficienty, které se počítají z částečných podílů vzniklých při celočíselném dělení (v jednotlivých krocích EA).

Příklad na rozšířený EA a Bézoutovu rovnost

Určete Bézoutovy koeficienty $x, y \in \mathbb{Z}$ tak, aby $\mathrm{NSD}(4829, 1373) = 4829x + 1373y$.

Řešení: Komentář na přednášce.

a, b	**q**	**x**	**y**
4829		1	0
1373	3	0	1
710	1	1	−3
663	1	−1	4
47	14	2	−7
5	9	−29	102
2	2	263	−925
1	2	−555	1952
0			

Tedy $\mathrm{NSD}(4829, 1373) = 1 = 4829 \cdot (-555) + 1373 \cdot 1952$.

Rozšířený Euklidův algoritmus – zdrojový kód v Ruby

Rozšířený Euklidův algoritmus pro nalezení Bézoutovy rovnosti:

```
def nsdBezout(a, b)
  x, xx, y, yy = 1, 0, 0, 1
  while (b != 0) do
    q = a / b
    r = a - b * q
    ra = x - q * xx
    rb = y - q * yy
    a = b
    b = r
    x = xx
    xx = ra
    y = yy
    yy = rb
  end
  return a, x, y
end
```

Obsah

Kongruence modulo n

Definice

Nechť $a, b \in \mathbb{Z}$, $n \in \mathbb{N}$. Řekneme, že daná celá čísla a, b jsou **kongruentní modulo n**, jestliže $n \mid (a - b)$.

Zavedené značení: $\mathbf{a} \equiv \mathbf{b} \pmod{\mathbf{n}}$.

Příklad

- Zřejmě $33 \equiv 5 \pmod{7}$, neboť v souladu s definicí $7 \mid (33 - 5)$.
- Podobně $-20 \equiv 10 \pmod{6}$, protože $6 \mid (-20 - 10)$.
- Dále třeba 13 není kongruentní s 27 modulo 12, jelikož $12 \nmid (13 - 27)$.

Vlastnosti kongruence modulo n

Věta

Nechť $a, b \in \mathbb{Z}$, $n \in \mathbb{N}$. Pak jsou následující podmínky ekvivalentní:

(i) $a \equiv b \pmod{n}$

(ii) existuje $k \in \mathbb{Z}$ takové, že $b = a - nk$

(iii) $a \bmod n = b \bmod n$.

Důkazu: (i) $\Rightarrow$ (ii): Víme, že $a \equiv b \pmod{n}$, právě když $n \mid (a-b)$. To dle definice relace dělitelnosti znamená, že existuje $k \in \mathbb{Z}$ takové, že $nk = a - b$. Odtud plyne (ii).
(ii) $\Rightarrow$ (iii): Předpokládejme platnost (ii) a nechť dále platí $a \bmod n = r$, to jest: $\exists q, r \in \mathbb{Z}$ tak, že $a = nq + r$, kde $0 \leq r < n$. Nyní ze (ii) vyjádříme a a dosazením do poslední rovnosti obdržíme: $b + nk = nq + r$. Odtud $b = (q-k)n + r$, přičemž $q - k \in \mathbb{Z}$ a $0 \leq r < n$, tedy $b \bmod n = r$ a (iii) platí.
(iii) $\Rightarrow$ (i): Na přednášce.

Vlastnosti kongruence modulo n

Věta

Pro $n \in \mathbb{N}$ je relace kongruence modulo n ekvivalencí na $\mathbb{Z}$.

Důkaz: Je třeba ověřit, že relace kongruence modulo n je reflexivní, symetrická a tranzitivní. Stačí tedy dokázat, že:

- pro každé $x \in \mathbb{Z}$ je $x \equiv x \pmod{n}$
- pro každé $x, y \in \mathbb{Z}$: $x \equiv y \pmod{n}$ implikuje $y \equiv x \pmod{n}$
- pro každé $x, y, z \in \mathbb{Z}$: $x \equiv y \pmod{n}$ a $y \equiv z \pmod{n}$ společně implikují, že $x \equiv z \pmod{n}$.

Komentář k důkazu na přednášce.

Vlastnosti kongruence modulo n

Věta

Buď $a, b, c, d \in \mathbb{Z}$, $n \in \mathbb{N}$, $a \equiv b \pmod{n}$, $c \equiv d \pmod{n}$. Pak:

(a) $a + c \equiv b + d \pmod{n}$

(b) $a - c \equiv b - d \pmod{n}$

(c) $a \cdot c \equiv b \cdot d \pmod{n}$

(d) $a^k \equiv b^k \pmod{n}$, kde k je libovolné přirozené číslo.

K důkazu: (a) a (b) jsou důsledkem snadno ověřitelného faktu: pokud $n \mid A$ a $n \mid B$, pak $n \mid (A \pm B)$.[12]
Pro důkaz (c) přepíšeme výchozí předpoklady na (jak víme ekvivalentní) tvar: $b = a - nk$, $d = c - nl$, kde $k, l \in \mathbb{Z}$. Po vynásobení a úpravě dostaneme:
$bd = (a - nk) \cdot (c - nl) = ac - n \cdot K$, kde $K = al + ck - nkl \in \mathbb{Z}$, což je (jak víme ekvivalentním) přepisem (c).
Důkaz (d) vyplývá z předpokladu po k-násobné aplikací (c).

[12] V důkazu pak stačí položit $A = a - b$ a $B = c - d$.

Příklad

Snadno ověříme, že

$$\begin{aligned} 22 &\equiv 8 \pmod 7 \\ 12 &\equiv 5 \pmod 7. \end{aligned}$$

Pak dle předchozí věty platí:

(a) $22+12 \equiv 8+5 \pmod 7$, tedy $34 \equiv 13 \pmod 7$

(b) $22-12 \equiv 8-5 \pmod 7$, tedy $10 \equiv 3 \pmod 7$

(c) $22 \cdot 12 \equiv 8 \cdot 5 \pmod 7$, tedy $264 \equiv 40 \pmod 7$

(d_1) $22^2 \equiv 8^2 \pmod 7$, tedy $484 \equiv 64 \pmod 7$

(d_2) $22^{10} \equiv 8^{10} \pmod 7$, tedy
$26\,559\,922\,791\,424 \equiv 1\,073\,741\,824 \pmod 7$.

Vlastnosti kongruence modulo n

Poznámka. Umíme dokázat, že binární relace „mít stejný zbytek po dělení přirozeným číslem n“ je ekvivalencí na $\mathbb{Z}$. Poslední věta prokazuje, že tato ekvivalence může být „povýšena“ na tzv. kongruenci, neboť je na $\mathbb{Z}$ kompatibilní — „hezky se chová“ — vzhledem k základním algebraickým operacím: $+$, $-$, $\cdot$, ^{k}. Na základě platnosti (a) – (d) z poslední věty říkáme, že $\equiv$ je relace kongruence vzhledem ke sčítání, odčítání, násobení a umocňování.

Víme, že každá ekvivalence jednoznačným způsobem indukuje rozklad na dané množině. Podívejme se dále, jak bude tento rozklad vypadat u relace kongruence modulo n (pro konkrétní $n \in \mathbb{N}$).

Příklad

Uvažujme na $\mathbb{Z}$ binární relaci „mít stejný zbytek po dělení přirozeným číslem $n = 5$“. Označme si ji $\equiv_5$. Jak vypadá rozklad $\Pi_{\equiv_5}$ množiny $\mathbb{Z}$ příslušný této ekvivalenci?

Řešení. Podívejme se nejprve, jak vypadají třídy ekvivalence určené prvky $0, 1, 2, 3, 4$. Máme:

$$\begin{aligned}
\overline{0} &= [0]_{\equiv_5} = \{\ldots, -15, -10, -5, 0, 5, 10, 15, \ldots\} \\
\overline{1} &= [1]_{\equiv_5} = \{\ldots, -14, -9, -4, 1, 6, 11, 16, \ldots\} \\
\overline{2} &= [2]_{\equiv_5} = \{\ldots, -13, -8, -3, 2, 7, 12, 17, \ldots\} \\
\overline{3} &= [3]_{\equiv_5} = \{\ldots, -12, -7, -2, 3, 8, 13, 18, \ldots\} \\
\overline{4} &= [4]_{\equiv_5} = \{\ldots, -11, -6, -1, 4, 9, 14, 19, \ldots\}.
\end{aligned}$$

Příslušným rozkladem množiny $\mathbb{Z}$ bude zřejmě systém množin $\Pi_{\equiv_5} = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}\}$. Je to tzv. množina zbytkových tříd modulo 5; značí se $\mathbb{Z}_5$.

Poznámka. Pro jiná $n \in \mathbb{N}$ by vyšly rozklady analogicky.

Zbytkové třídy modulo n

Zbytkovou třídou modulo n rozumíme množinu všech celých čísel, která při dělení přirozeným číslem n dávají stejný zbytek (odtud název) po celočíselném dělení. Tedy třída ekvivalence $\equiv_n$ je zbytková třída modulo n.

Třídu obsahující číslo $i \in \mathbb{Z}$ značíme $[i]_{\equiv_n}$ nebo zkráceně $\bar{i}$. Platí, že $\bar{i} = \{j \mid j = i + kn, k \in \mathbb{Z}\}$. Tuto množinu lze chápat jako jeden celek a celá čísla, která obsahuje nadále nerozlišovat. Množinu všech zbytkových tříd modulo n značíme $\mathbb{Z}_n$. Snadno nahlédneme, že $\mathbb{Z}_n = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$.

Na množině zbytkových tříd $\mathbb{Z}_n$ definujeme operaci sčítání $\oplus$ a násobení $\odot$ a podíváme se na jejich významné vlastnosti.

Definice a vybrané vlastnosti $\oplus$ a $\odot$

Na množině $\mathbb{Z}_n$ definujeme dvě binární operace $\oplus$ a $\odot$:

$$\begin{aligned} \bar{i} \oplus \bar{j} &= \overline{i+j} \\ \bar{i} \odot \bar{j} &= \overline{i \cdot j}. \end{aligned}$$

Operace $\oplus$ a $\odot$ jsou obě komutativní a asociativní a splňují distributivní zákon ($\odot$ vzhledem k $\oplus$). Očividně, pro libovolnou třídu $\bar{i}$ platí: $\bar{i} \oplus \bar{0} = \bar{i}$, $\bar{i} \oplus \overline{-i} = \bar{0}$ a $\bar{i} \odot \bar{1} = \bar{i}$, $\bar{i} \odot \bar{0} = \bar{0}$.

Podrobnosti a příklad na přednášce.

Malá Fermatova věta

Pro každé prvočíslo p a každé $a \in \mathbb{N}$ platí: $a^p \equiv a \pmod{p}$.
Pokud navíc NSD$(a,p) = 1$, pak $a^{p-1} \equiv 1 \pmod{p}$.

Příklad

Podle malé Fermatovy věty platí: $59^{113} \equiv 59 \pmod{113}$ a $18^{36} \equiv 1 \pmod{37}$.

Příklad

U čísla 175^{146} určete zbytek po celočíselném dělení 17.

Řešení: Vhodné použití malé Fermatovy věty, ve tvaru $5^{16} \equiv 1 \pmod{17}$, výpočet značně zjednodušší. Máme: $175^{146} \equiv 5^{146} = (5^{16})^9 \cdot 5^2 \equiv 1^9 \cdot 25 = 25 \equiv 8 \pmod{17}$.

Eulerova funkce

Definice

Pro přirozené číslo n je **Eulerova funkce** $\varphi(n)$ rovna počtu přirozených čísel menších než n a nesoudělných s n.

Následující věta se dokazuje na základě ZVA a vztahu $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$, kde $m, n \in \mathbb{N}$ a $\mathrm{NSD}(m, n) = 1$.

Věta

$$\text{Pro } n \in \mathbb{N}, p \in \mathbb{P} \text{ je: } \varphi(n) = n \cdot \prod_{p|n} \left(1 - \frac{1}{p}\right).$$

Násobí se přes různá prvočísla p dělící n.

Příklady

- $\varphi(10) = 4$, neboť s číslem deset nesoudělná přirozená čísla menší než 10 jsou čtyři: $1, 3, 7, 9$
- $\varphi(42) = 42 \cdot \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{3}\right) \cdot \left(1 - \frac{1}{7}\right) = 42 \cdot \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{6}{7} = 12$
- Pro libovolné prvočíslo $p \in \mathbb{P}$ je $\varphi(p) = p - 1$.

Eulerova věta

Následující věta využívá Eulerovu funkci $\varphi(n)$.

Eulerova věta

Nechť $a, n \in \mathbb{N}$. Pokud $\text{NSD}(a, n) = 1$, pak $a^{\varphi(n)} \equiv 1 \pmod{n}$.

Rozmyslete si, že Malá Fermatova věta je přímým důsledkem Eulerovy věty.

Příklad

Podle Eulerovy věty platí: $25^{12} \equiv 1 \pmod{42}$, neboť $\varphi(42) = 12$ a $\text{NSD}(25, 42) = 1$.

RSA[13] je příkladem dodnes používaného kryptografického (šifrovacího) systému stojícího na principech asymetrické kryptografie. Kryptografický systém RSA lze využít i pro elektronický podpis (viz dále).

Poznámka. S RSA se na Internetu setkáváme každodenně, aniž by si to většina uživatelů plně uvědomovala.

[13] Název šifry **RSA** je zkratkou prvních písmen z příjmení tří autorů: Ron **R**ivest, Adi **S**hamir a Leonard **A**dleman, kteří o něm v roce 1977 napsali článek do odborného časopisu. Ještě před nimi (v roce 1973) popsal ekvivalentní systém anglický matematik Clifford Cocks. Jeho myšlenky však zůstaly až do roku 1993 uchovány s podtitulem přísně tajné.

Asymetrická kryptografie a jednosměrné funkce

V **asymetrické kryptografii** (kryptografii s veřejným klíčem) se pro šifrování a dešifrování používají odlišné klíče, čímž se zásadně odlišuje od symetrické kryptografie, která používá k šifrování i dešifrování jediný klíč. Asymetrická kryptografie je založena na tzv. jednosměrných funkcích.

Jednosměrná funkce (jednocestná funkce), je taková funkce, kterou lze snadno vyčíslit, ale je velmi obtížné z výsledku funkce odvodit její vstup. Jsou to tedy funkce, které lze snadno provádět jedním směrem (vynásobit spolu dvě velká prvočísla; smíchat dvě různé barvy), přičemž postup opačným směrem je výpočetně velmi náročný (rozložit velké číslo na součin dvou prvočísel; rozdělit smíchanou barvu na dvě původní barvy).[14]

[14] Podobnými problémy jsou výpočet diskrétního logaritmu či problém batohu.

K asymetrické kryptografii a k RSA

Šifrovací klíč pro asymetrickou kryptografii sestává ze dvou částí: jedna část se používá pro šifrování zpráv, druhá pro dešifrování. Hlavní výhodou je, že nemusí dojít ke vzájemné výměně klíčů. Nejčastěji je v asymetrické kryptografii využíván tzv. veřejný a soukromý klíč. Veřejný klíč je komukoli k dispozici a pomocí něj se zprávy šifrují. Soukromý (privátní) klíč je přísně tajný. Prakticky jen jeho vlastník může dešifrovat zprávy zašifrované veřejným klíčem.
Veřejný a soukromý klíč jsou spolu provázány matematicky a to tak, aby se z veřejného nedal (v rozumném čase) odvodit klíč soukromý. V případě RSA se používá jednosměrná funkce založená na výše zmíněném násobení, respektive faktorizaci.

Rozložit velmi velké číslo na součin prvočísel (provést jeho faktorizaci) je velmi obtížná úloha. Z čísla $n = p \cdot q$ (n by mělo být alespoň 2048 bitů dlouhé) je v rozumném čase prakticky nemožné zjistit činitele p a q, a proto je kryptografický systém RSA považován za bezpečný.

RSA: tvorba veřejného a soukromého klíče

Alice a Bob chtějí komunikovat prostřednictvím otevřeného (nezabezpečeného) kanálu a Bob by chtěl Alici poslat soukromou zprávu. Proto si v následujících krocích Alice nejprve vytvoří veřejný a soukromý klíč:

- Zvolí dvě různá velká náhodná prvočísla p a q a vypočítá hodnotu jejich součinu: $n = p \cdot q$.
- Spočítá hodnotu Eulerovy funkce $\varphi(n) = (p-1) \cdot (q-1)$.
- Zvolí číslo $e \in \mathbb{N}$ tak, aby $e < \varphi(n)$ a $\mathrm{NSD}(e, \varphi(n)) = 1$.
- Z kongruenční rovnice $d \cdot e \equiv 1 \pmod{\varphi(n)}$ vypočítá číslo d. Číslo d určí pomocí rozšířeného EA aplikovaného na výpočet $\mathrm{NSD}(e, \varphi(n))$. Číslo d bude jedním z Bézoutových koeficientů.

Veřejným klíčem je dvojice $\langle n, e \rangle$, soukromým klíčem je trojice $\langle p, q, d \rangle$. Veřejný klíč Alice „klidně“ uveřejní (pošle Bobovi veřejným kanálem), soukromý klíč si uchová v přísné tajnosti.

RSA: zašifrování a dešifrování číselné zprávy

Předpokládejme, že Bob chce Alici poslat zprávu z, která je v číselné podobě. Pro její zašifrování využije veřejný klíč Alice, tedy mu dostupnou dvojici čísel $\langle n, e \rangle$. Zašifrovanou zprávu, šifru s, získá z následující kongruenční rovnice:

$$z^e \equiv s \pmod{n}.$$

Šifru s pošle (veřejným kanálem) Alici, která ji dešifruje s pomocí svého (tajného) soukromého klíče $\langle p, q, d \rangle$:

$$s^d \equiv (z^e)^d \equiv z \pmod{n}.$$

Poznámka. To, že Alici vyjde jako výsledek zpráva z je důsledkem Eulerovy věty.

RSA: příklad s nízkými prvočísly

Příklad

Zvolme pro názornost malá prvočísla: $p = 59$, $q = 23$. Pak $n = p \cdot q = 1357$ a $\varphi(n) = (p-1) \cdot (q-1) = 58 \cdot 22 = 1276$. Dále zvolme přirozené číslo e tak, aby $e < \varphi(n)$ a $\mathrm{NSD}(e, \varphi(n)) = 1$. Nechť třeba $e = 15$. Nyní, z rovnice $d \cdot e \equiv 1 \pmod{\varphi(n)}$ vypočítáme, že $d = 2467$. Veřejným klíčem je dvojice $\langle 1357, 15 \rangle$. Soukromým klíčem je trojice $\langle 59, 23, 2467 \rangle$.

Zprávu $z = 1045$ zašifrujeme s veřejným klíčem na číslo s:

$$z^e \equiv s \pmod{n}, \quad \text{tedy} \quad 1045^{15} \equiv s \pmod{1357}.$$

Výpočtem zjistíme, že šifrou s je číslo 1270.

Šifru s lze soukromým klíčem dešifrovat na původní zprávu z:

$$s^d \equiv (z^e)^d \equiv z \pmod{n}, \quad \text{tedy} \quad (1270)^{2467} \equiv z \pmod{1357}.$$

Odtud vypočteme, že $z = 1045$.

Poznámka: RSA lze využít i pro elektronický podpis

RSA lze použít k elektronickému (digitálnímu) podpisu, který v elektronické komunikaci nahrazuje klasický, ručně psaný podpis. Připojení elektronického podpisu k dané zprávě zajišťuje autentičnost[15] (možnost ověření, kdo zprávu podepsal a odeslal) a nepopiratelnost (nemožnost popření, kým byla zpráva podepsána). Nepopiratelnost je možné zajistit pomocí „opačného“ použití RSA (či jiné asymetrické šifry). Principiálně stačí jen vyměnit roli veřejného a soukromého klíče (pro šifrování se použije soukromý klíč, pro dešifrování veřejný klíč). Pokud Alice zašifruje[16] zprávu svým soukromým klíčem (jejímž je jediným vlastníkem), pak nemůže popřít, že jej podepsala. Bob (nebo kdokoliv jiný) může zprávu dešifrovat.

[15] Autentičnost je v elektronické komunikaci realizována použitím tzv. digitálního certifikátu.

[16] Chce-li Alice poslat podepsanou zprávu Bobovi, připojí k ní číslo získané „dešifrováním“ haše své zprávy pomocí svého soukromého klíče. Bob poté (jakoby zpětně) „zašifruje“ tento podpis pomocí veřejného klíče Alice a výsledek porovná s hašem zprávy. Stejná hodnota vyjde, pokud zpráva zůstala nezměněna.

Obsah

Intuitivní vymezení pojmu algoritmus

Algoritmus je přesný návod nebo postup, kterým lze vyřešit daný typ úlohy. Algoritmus obsahuje

1) hodnoty vstupních dat
2) předpis pro řešení
3) požadovaný výsledek, tedy výstupní data.

Pro zpřesnění pojmu algoritmus dodejme: je to předpis (konečný proces), který se skládá z uspořádané sady jednoznačných a proveditelných kroků a který zabezpečí, že na základě vstupních dat jsou poskytnuta požadovaná data výstupní.

Poznámky k vymezení pojmu algoritmus

Poznámka. Slovo „algoritmus“ pochází ze jména významného perského matematika al-Chorézmího (z 9. století našeho letopočtu).

Poznámka. Algoritmus bývá nahlížen také jako konečná posloupnost instrukcí pro vyřešení nějakého problému, které lze vykonávat mechanicky (jeho vykonávání tedy není podmíněno porozuměním tomu, proč a jak algoritmus funguje).

Poznámka. V analogii lze algoritmus přirovnat k „mlýnku na data“. Nasypeme-li do něj správná data a zameleme, obdržíme požadovaný výsledek. Přitom kvalita „mlýnku“ může být různá (o tzv. časové a paměťové náročnosti, viz dále).

Vybrané příklady algoritmů

- Erathostenovo síto (algoritmus na hledání prvních n prvočísel)
- Eukleidův algoritmus (na hledání NSD dvou přirozených čísel a jeho rozšířená verze, hledající navíc Bézoutovy celočíselné koeficienty)
- algoritmus na dělení mnohočlenů
- algoritmus na vyřešení kvadratické rovnice na množině $\mathbb{R}$
- algoritmus binárního vyhledávání
- Dijkstrův algoritmus (na hledání nejkratších cest v grafu)
- Kruskalův algoritmus (pro hledání minimální kostry grafu)
- algoritmy pro setřídění posloupnosti čísel (quicksort, mergesort, . . .)
- algoritmy na kompresi dat (LZ77, LZ78, LZW, Huffmanovo kódování, JPEG, MP3, . . .)
- šifrovací algoritmy (AES, DES, RSA, . . .)

Základní vlastnosti algoritmů

- **konečnost** (finitnost)
  každý algoritmus by měl skončit po konečném počtu kroků (v praxi je pochopitelně chtěno, aby požadovaný výsledek byl poskytnut v „rozumném“ čase, ne za milion let)
- **jednoznačnost** (determinovanost)
  každý krok algoritmu by měl být jednoznačně a přesně definován (v každé situaci by mělo být jasné, co a jak se má provést, jak má provádění algoritmu pokračovat)
- **obecnost** (hromadnost)
  algoritmus řeší celou třídu obdobných problémů (například jak obecně vypočítat součin dvou celých čísel) a neřeší jen jeden konkrétní problém (jak vypočítat kolik je $2 \cdot 6$).

K dalším typickým vlastnostem algoritmů patří:

- **rezultativnost** – algoritmus po zadání vstupních dat vrací nějaký výstup (například chybové hlášení, nebo třeba provede změnu parametrů v nějakém systému). (Algoritmus, který by nevydal žádný výsledek by byl v praxi k ničemu.)
- **korektnost** – je požadována a chtěna správnost algoritmem vydaného výsledku
- **opakovatelnost** – pro stejné vstupní údaje by měl algoritmus vracet stejné výsledky. (Při opakovaném řešení stejné kvadratické rovnice očekáváme totožné výstupy).[17]

[17] Existují výjimky: například v případě generování pseudonáhodných čísel by vracení jediného čísla nebylo žádoucí.

Poznámka: některé druhy algoritmů 1/2

- rekurzivní algoritmy – využívají (volají) sami sebe
- hladové algoritmy – k řešení se propracovávají po jednotlivých rozhodnutích, která jsou nevratná; například Kruskalův algoritmus pro hledání minimální kostry grafu
- algoritmy typu rozděl a panuj – dělí problém na menší podproblémy až po triviální podproblémy (které lze vyřešit přímo), dílčí řešení pak vhodným způsobem sloučí. (Jedná se o typický případ aplikace metody návrhu shora dolů.)
- pravděpodobnostní algoritmy – provádějí některá rozhodnutí náhodně či pseudonáhodně
- paralelní algoritmy – podstata tkví v rozdělení úlohy mezi více počítačů (respektive pro víceprocesorový počítač).

Poznámka: některé druhy algoritmů 2/2

- genetické algoritmy – pracují na základě napodobování evolučních procesů, postupným „pěstováním“ nejlepších řešení pomocí mutací a křížení
- algoritmy dynamického programování – postupně řeší části problému od nejjednodušších po složitější s tím, že využívají výsledky již vyřešených jednodušších podproblémů
- heuristické algoritmy – nekladou si za cíl nalézt nejlepší možné řešení; stačí jim nalézt dostatečně vhodné přiblížení. Používají se v situacích, kdy dostupné zdroje (nejčastěji čas) nepostačují na využití exaktních algoritmů (nebo pokud nejsou žádné přesné algoritmy vůbec známy).

Poznámka: Jeden algoritmus může patřit zároveň do více skupin; například quicksort může být rekurzivní algoritmus typu rozděl a panuj.

Poznámka k reprezentaci algoritmů

Aby mohl být algoritmus vhodně reprezentován, je třeba stanovit úroveň podrobností jeho popisu (zejména s ohledem na to, komu je popisován). Laik potřebuje výrazně podrobnější popis než odborník.

V praxi je (v počítačové komunitě) pod pojmem **program**[18] obvykle označována formální reprezentace algoritmu, která je určena k tomu, aby ji realizoval počítač.

Reprezentace algoritmů vyžaduje nějakou formu jazyka. Základními „stavebními kameny“ jsou tzv. **primitiva**[19]. Souhrn přesně definovaných primitiv a pravidla umožňující tvoření složitějších prvků (z primitiv) tvoří **programovací jazyk**. Proces vývoje programu, jeho zakódování do kompatibilní podoby a vložení do stroje se nazývá **programování**.

---

[18] Aktivita provádění programu bývá označována jako **proces**.

[19] Každé primitivum má vlastní syntaxi (symbolickou reprezentaci) a sémantiku (zamýšlený význam).

Poznámka k přesné definici algoritmu

Definovat pojem algoritmus přesně je nemožné.[20] To si dobře uvědomovali průkopníci moderní informatiky (zejména Alonzo Church a Alan Turing). Turing proto definoval matematické modely jednoduchého univerzálního počítače (tzv. Turingovy stroje).

Churchova–Turingova teze

Každý algoritmus je možné realizovat nějakým Turingovým strojem.

Churchova–Turingova teze není věta, kterou by bylo možno dokázat v matematickém smyslu. Není totiž formálně definováno, co je to algoritmus. Jelikož je ale výpočet na Turingově stroji přesně definován, informatici jej všeobecně považují za definici pojmu algoritmus.

[20] Stejně tak nelze přesně vymezit třeba pojem stůl, viz přednáška.

Algoritmicky neřešitelné problémy

S přesně definovaným pojmem algoritmus (zavedeným přes Turingovy stroje) se můžeme ptát, co vše lze algoritmicky řešit. Velmi překvapivě (koncem třicátých let 20. století) objevil Church a nezávisle na něm i Turing, že existují jednoduché, přesně formulované výpočetní problémy, které nejsou Turingovými stroji řešitelné, tedy nejsou na počítači řešitelné žádným algoritmem.[21]

[21] Algoritmicky neřešitelných problémů existuje nespočetně mnoho, přičemž algoritmicky řešitelných je „jen“ spočetně mnoho.

Dva významné algoritmicky neřešitelné (nerozhodnutelné) problémy

- **Problém zastavení** (Halting problem.[a])
  Lze sestrojit algoritmus, který by o každém algoritmu uměl rozhodnout, zda jeho činnost skončí po konečném počtu kroků či nikoliv?
- **Problém rozhodnutí** (Entscheidungsproblem[b]).
  Existuje algoritmus, který by uměl rozhodnout, zda je dané matematické tvrzení v daném formálním jazyce pravdivé nebo nepravdivé?

---

[a] Turing v roce 1936 dokázal, že neexistuje obecný algoritmus, který by řešil problém zastavení pro všechny vstupy všech programů.

[b] Entscheidungsproblem je úloha, kterou poprvé předložil německý matematik David Hilbert roku 1928.

Obsah

Již víme, že existují algoritmicky neřešitelné (nerozhodnutelné) problémy, pro které nemá smysl zkoušet algoritmy konstruovat. Příkladem je problém zastavení (halting problem). Jedná se o sestrojení algoritmu, který by o každém algoritmu uměl rozhodnout, zda jeho činnost skončí po konečném počtu kroků či nikoliv.

Redukcí z problému zastavení se dá ukázat nerozhodnutelnost celé řady problémů, které se týkají ověřování chování programů:

- Vydá daný program pro nějaký vstup odpověď „Ano“?
- Zastaví se daný program pro libovolný vstup?
- Dávají dva dané programy pro stejné vstupy stejný výstup?

Dále se budeme zabývat **algoritmicky řešitelnými problémy**, jejich časovými a pameťovými nároky. Bude nás tedy zajímat efektivita algoritmů z hlediska rychlosti výpočtu a velikosti potřebné (operační) paměti.

Složitost algoritmu

Složitost každého algoritmu může být studována buď z hlediska **paměťové náročnosti** nebo z hlediska **časové náročnosti**. Paměťovou náročností rozumíme požadavek na velikost paměti počítače, jež je zapotřebí k provedení výpočtu. Podobně časovou náročností rozumíme čas potřebný pro výpočet. Tento čas se obvykle neměří v časových jednotkách, ale počtem provedených elementárních kroků algoritmu.[22]

Poznámka. Dále se budeme věnovat pouze časové složitosti[23], neboť k paměťové složitosti se přistupuje analogicky.

---

[22] Při studiu algoritmů se rozlišuje **časová složitost v nejhorším případě** a **časová složitost v průměrném případě**.

[23] Složitost je funkce závislá na velikosti vstupních dat algoritmu. U funkcí popisujících časovou složitost budeme uvažovat pouze jejich **řádovou velikost**, tedy například složitosti lišící se konstantním násobkem budeme považovat za stejné.

Notace „velké O“

Definice – řádové porovnávání funkcí

Nechť f, g jsou dvě funkce, které přiřazují přirozeným číslům reálná čísla. Pak řekneme, že funkce f **roste řádově nejvýše** jako funkce g (f je třídy $O(g)$), píšeme $f(n) \in O(g(n))$, právě když existují čísla $K > 0$ a $n_0 \in \mathbb{N}$ taková, že pro každé přirozené číslo $n \geq n_0$ platí $f(n) \leq K \cdot g(n)$.

Definice

Řekneme, že algoritmus má **polynomiální (polynomickou) časovou složitost**, právě když existuje polynom p takový, že $f(n) \leq p(n)$ pro všechna $n \in \mathbb{N}$.

Polynomiální jsou tedy všechny algoritmy, jejichž funkci časové složitosti můžeme shora omezit polynomem $a_k n^k + a_{k-1} n^{k-1} + \cdots + a_2 n^2 + a_1 n + a_0$, kde $a_k, a_{k-1}, \ldots, a_2, a_1, a_0 \in \mathbb{N}_0$ a také $k \in \mathbb{N}_0$.

Příklad

Ověřte, že platí

(a) $\log_2 n \in O(n)$

(b) $5n^3 + 3n^2 + 7 \in O(n^3)$.

Řešení.

(a) Podle definice je $\log_2 n \in O(n)$, pokud existuje konstanta $K > 0$ a $n_0 \in \mathbb{N}$ tak, že $\log_2 n \leq Kn$ pro každé $n \in \mathbb{N}$, $n \geq n_0$. V tomto případě stačí zvolit třeba $K = 100$ a $n_0 = 1$.

(b) Podobně, $5n^3 + 3n^2 + 7 \in O(n^3)$, pokud existuje kladná konstanta K a přirozené číslo n_0 tak, že $5n^3 + 3n^2 + 7 \leq Kn^3$ pro každé $n \in \mathbb{N}$, $n \geq n_0$. Z nerovnice

$$K \geq 5 + \frac{3}{n} + \frac{7}{n^3},$$

vidíme, že stačí například položit $K = 6$ a $n_0 = 4$.

Příklad

$3n^3 - n^2 + 2n \in O(n^3)$, neboť $3n^3 - n^2 + 2n \leq Kn^3 \Leftrightarrow$
$n^3(K-3) + n^2 - 2n \geq 0$, což pro $K = 4$ dává $n^3 + n^2 - 2n \geq 0 \Leftrightarrow$
$n(n+2)(n-1) \geq 0$, což platí $\forall n \geq n_0 = 1$.

Příklad*

Zjistěte, zda $n^2 \in O(n\ln^2 n)$.

Řešení. Poznamenejme, že asymptotickou notaci lze definovat i přes výpočet limity a to následovně:

$$f(n) \in O(g(n)) \quad \Longleftrightarrow \quad \lim_{n\to+\infty} \frac{f(n)}{g(n)} \in \langle 0; +\infty).$$

Platí:

$$\lim_{n\to+\infty} \frac{n}{ln^2 n} = +\infty,$$

odkud $n^2 \notin O(n\ln^2 n)$.

Příklad

Některé typické příklady časové složitosti (od nejrychlejší po nejpomalejší):

- $O(1)$ – **konstantní** (indexování prvků v poli)
- $O(\log N)$ – **logaritmická** (vyhledání prvku v seřazeném poli metodou půlení intervalu)
- $O(N)$ – **lineární** (vyhledání prvku v neseřazeném poli sekvenčním vyhledáváním)
- $O(N \log N)$ – **lineárnělogaritmická** (seřazení pole N čísel dle velikosti; třídění sléváním, třídění haldou)
- $O(N^2)$ – **kvadratická** (třídění N čísel dle velikosti; třídění přímým výběrem, bublinkové třídění)
- ...
- $O(2^N)$ – **exponenciální** (Fibonacciho posloupnost řešená pomocí stromové rekurze)
- $O(N!)$ – **faktoriálová** (řešení problému obchodního cestujícího hrubou silou).

Definice

Nechť $f, g : \mathbb{N} \to \mathbb{R}$ jsou funkce. Pak píšeme

- $f(n) \in O(g(n)) \Leftrightarrow \exists K > 0, \exists n_0 \in \mathbb{N}, \forall n \geq n_0 : f(n) \leq K \cdot g(n)$
  a říkáme, že funkce f roste řádově nejvýše jako funkce g.
- $f(n) \in \Omega(g(n)) \Leftrightarrow \exists k > 0, \exists n_0 \in \mathbb{N}, \forall n \geq n_0 : f(n) \geq k \cdot g(n)$
  a říkáme, že funkce f roste řádově aspoň jako funkce g.
- $f(n) \in \Theta(g(n)) \Leftrightarrow f(n) \in O(g(n))$ a $f(n) \in \Omega(g(n))$
  a říkáme, že funkce f roste řádově stejně jako funkce g;
  nebo, že funkce f a g jsou řádově ekvivalentní (neboli asymptoticky ekvivalentní).

Příklad

Dokažte, že $\frac{n^2-1}{n+1} \in \Theta(n)$.

Řešení.

$$k \cdot n \leq \frac{n^2-1}{n+1} \leq K \cdot n$$

$$k \cdot n \leq n-1 \leq K \cdot n$$

$$k \leq 1 - \frac{1}{n} \leq K$$

Je to splněno například pro $k = \frac{1}{2}$, $K = 1$ a pro $\forall n \geq n_0 = 2$.

Příklad

Dokažte, že $3n^2 \in \Theta(n^2)$, přičemž $3n^2 \notin \Theta(n)$ a $3n^2 \notin \Theta(n^3)$.

Řešení. Zkuste sami.

Uvažme počítač, u nějž provedení jedné instrukce trvá jednu nanosekundu. Následující tabulka ukazuje délky trvání výpočtu, spustíme-li na takovém počítači algoritmus o řádové složitosti $f(n)$ se vstupními daty velikosti n.

$f(n)$	$n = 20$	$n = 40$	$n = 60$	$n = 80$	$n = 100$	$n = 1000$
n	$20ns$	$40ns$	$60ns$	$80ns$	$0,1\mu s$	$1\mu s$
$n \log n$	$86ns$	$0,2\mu s$	$0,35\mu s$	$0,5\mu s$	$0,7\mu s$	$10\mu s$
n^2	$0,4\mu s$	$1,6\mu s$	$3,6\mu s$	$6,4\mu s$	$10\mu s$	$1ms$
n^4	$0,16ms$	$2,56ms$	$13ms$	$41ms$	$0,1s$	$16,8min$
2^n	$1ms$	$16,8min$	$36,6let$			
$n!$	$77let$					

Předchozí tabulka potvrzuje oprávněnost představy: prakticky použitelný algoritmus je algoritmus s nejvýše polynomickou časovou složitostí.[24]

Předchozí představu rámcově potvrzuje i další tabulka, která popisuje, jak se zvětší rozsah zpracovatelných úloh v případě zvětšení výpočetní rychlosti použitého počítače 100x a 1000x, jestliže původně bylo možno v daném časovém limitu zpracovat vstupní data o velikosti $n = 100$.

[24] Nelze to však brát jako dogma. Viz následující dva příklady:

- $f_1(n) = 2^{100} \cdot n$,
- $f_2(n) = 2^{n^{0,0001}}$ $(= 2^{10}$ pro $n = 10^{10^4})$.

$f(n)$	zrych. výp. $1x$	zrych. výp. $100x$	zrych. výp. $1000x$
n	100	10000	100000
$n \log n$	100	5362	43150
n^2	100	1000	3162
n^4	100	316	562
2^n	100	106	109
$n!$	100	100	101

Z tabulek je vidět, že už pro algoritmy s exponenciální časovou složitostí je typická existence mezní velikosti vstupních dat, nad níž je úloha prakticky neřešitelná i při zvýšení rychlosti počítače o několik řádů.

Obsah

Konečný automat

Konečný automat je teoretický výpočetní model primitivního počítače používaný v informatice pro studium vyčíslitelnosti a obecně formálních jazyků. Konečné automaty se používají pro zpracování regulárních výrazů, například jako součást lexikálního analyzátoru v překladačích a uplatňují se třeba také při návrhu sekvenčních logických obvodů.

Konečný automat

Konečný automat lze chápat jako abstraktní model určitého specifického typu výpočtu. Výpočet probíhá diskrétním způsobem se symboly. Konečný automat sestává z několika přechodů a z konečné (odtud název) množiny stavů, přičemž v danou chvíli se automat nachází právě v jediném ze svých stavů (v tzv. aktuálním stavu). Jeden z jeho stavů je tzv. počáteční (výchozí) a určitá podmnožina všech jeho stavů vymezuje tzv. koncovou množinu stavů. Na vstupu automat obdrží tzv. vstupní slovo, které se skládá ze symbolů, a které automat zleva doprava postupně čte. Právě na základě symbolů, které čte ze vstupu a na základě tzv. přechodové funkce[25] může mezi svými stavy přecházet. Pokud po přečtení celého slova skončí v jednom z koncových stavů, dané slovo přijímá (v opačném případě jej zamítá).

[25] Přechodová funkce definuje výše zmíněné přechody – přiřazuje danému (aktuálnímu) stavu a symbolu na vstupu stav následující.

Dříve než definujeme potřebné pojmy, vyřešíme dva následující příklady.

Příklad

Nechť $T = \{a, b, c\}$. Sestavte konečný automat, který rozpozná slova, která obsahují podřetězec *abc*.

Řešení: na přednášce.

Příklad

Nechť $T = \{0, 1\}$. Sestavte konečný automat, který přijímá regulární jazyk řetězců, které vyjadřují binární číslo dělitelné třemi (beze zbytku).

Řešení: na přednášce.

Definice

Abeceda T je libovolná neprázdná konečná množina symbolů. **Řetězec (slovo)** α je libovolná konečná posloupnost symbolů abecedy T. Neprázdný řetězec (slovo) $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ budeme stručně zapisovat ve tvaru $\alpha = \alpha_1\alpha_2\ldots\alpha_n$. **Prázdný řetězec (prázdné slovo)** budeme značit ε. Počet symbolů v řetězci (ve slově) je jeho **délkou** a označuje se $|\alpha|$. V souladu s tím $|\varepsilon| = 0$. Řetězec (slovo) α je **podřetězec (podslovo)** řetězce (slova) β, jestliže je „souvislou“ podčástí β. **Zřetězením** řetězců (slov) $\alpha = \alpha_1\alpha_2\ldots\alpha_k$ a $\beta = \beta_1\beta_2\ldots\beta_l$ rozumíme řetězec (slovo) $\alpha\beta = \alpha_1\alpha_2\ldots\alpha_k\beta_1\beta_2\ldots\beta_l$. V souladu s tím $\alpha\varepsilon = \alpha = \varepsilon\alpha$. Pro $n \in \mathbb{N}$ symbol α^n označuje **n-násobné zřetězení** α; pro $n = 0$ je $\alpha^n = \varepsilon$.

Příklad

Pro $T = \{0,1,2\}$, $\alpha = 101122$ a $\beta = 112$ je $|\alpha| = 6$ a $|\beta| = 3$. Dále $\alpha\beta = 101122112$, $\beta^3 = 112112112$ a $\alpha^0 = \varepsilon$. Přitom $|\alpha\beta| = 6+3 = 9$ a $|\alpha^0\beta^3| = 0+9 = 9$. Zřejmě β je podřetězcem (podslovem) α, ale ne naopak.

Definice

Pozitivní uzávěr T^+ abecedy T je množina všech neprázdných řetězců (slov) symbolů množiny T. **Uzávěr** T^* je množina $T^+ \cup \{\varepsilon\}$. Libovolná podmnožina uzávěru T^* je **(formální) jazyk L** nad (abecedou) T. Tedy $L \subseteq T^*$.

Příklad

- Pro $T = \{a\}$ je uzávěr $T^* = \{a\}^* = \{a^n; n \in \mathbb{N}_0\}$.
- Pro $T = \{a, b\}$ je pozitivní uzávěr $T^+ = \{a, b\}^+$ množina všech konečných neprázdných řetězců (slov) sestávajících výhradně ze symbolů a nebo b, například: a, bab, a^5, b^3a^2.

Příklad

- Konečným (formálním) jazykem L nad $T = \{a, b\}$ je třeba množina $L = \{\varepsilon, a, b, aa, ab, bb\} \subseteq T^*$.
- Pro $T = \{a, b, c\}$ je $L = \{a^n b^n c^n; n \in \mathbb{N}_0\} \subseteq T^*$ (formálním) jazykem nad abecedou T.
- $\mathbb{Z}$ je (formální) jazyk nad $T = \{-, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Deterministický konečný automat (DKA)

Definice

Deterministický konečný automat[a] je pětice $\mathscr{A} = \langle T, Q, \delta, q_0, F \rangle$, kde

- T je konečná neprázdná množina vstupních symbolů (abeceda)
- Q je konečná množina stavů
- δ je tzv. **přechodová funkce**, $\delta : Q \times T \to Q$, popisující pravidla přechodů mezi stavy
- q_0 je **počáteční stav**, $q_0 \in Q$
- F je **množina koncových stavů**, $F \subseteq Q$.

[a] Existují i tzv. **nedeterministické konečné automaty** s přechodovou funkcí $\delta : Q \times T \to 2^Q$. Lze o nich dokázat, že mají stejnou výpočetní sílu jako deterministické konečné automaty.

Popis činnosti deterministického konečného automatu:

- Na počátku se automat nachází v počátečním stavu q_0 a na vstupu má nějaké slovo $w \in T^*$.
- Dokud není $|w| = 0$, tak se v každém kroku odebere nejlevější symbol x ze vstupního slova w a z aktuálního stavu q přejde automat (podle přechodové funkce) do stavu $\delta(q, x)$.
- Skončí-li automat po přečtení (zpracování) celého vstupního slova w v koncovém stavu, pak daný vstup přijímá (rozpoznává). V případě, že automat přečte celé vstupní slovo w a nenachází se v koncovém stavu, dané slovo nepřijímá (zamítá). Automat také slovo w nepřijímá, když pro aktuální stav q a odebraný symbol x neexistuje funkční hodnota $\delta(q, x)$, tedy, pokud není možný další přechod.

Poznámka. Konečný automat lze velmi názorně reprezentovat (mírně modifikovaným) orientovaným grafem – stavy jsou vrcholy, přechody jsou vyznačeny hranami ohodnocenými vstupními symboly a koncové stavy jsou vyznačeny dvojitým kroužkem.

Příklad

Nechť $T = \{0,1\}$ je abeceda. Graficky znázorněte DKA přijímající všechna slova, která obsahují lichý počet jedniček a libovolný počet nul.

Řešení. Pro vyřešení úlohy nám stačí uvažovat automat se dvěma různými stavy: s počátečním stavem q_0 a s jediným koncovým stavem q_1. Přechodová funkce δ je definována takto: $\delta(q_0,0) = q_0$, $\delta(q_0,1) = q_1$, $\delta(q_1,0) = q_1$, $\delta(q_1,1) = q_0$. Grafické znázornění na přednášce.

Příklad

Nechť množina $T = \{a, b, c, d\}$ je abeceda. Sestavte deterministický konečný automat, který přijme všechna slova, která obsahují podřetězec *dbca* nebo podřetězec *dbcd*.

Řešení. Pro vyřešení úlohy nám stačí automat s pěti stavy: q_0, q_1, q_2, q_3, q_4, kde q_0 je počáteční stav a q_4 je (jediný) koncový stav. Přechodová funkce δ je pak dána následovně:

$$
\begin{array}{lllll}
\delta(q_0,a)=q_0, & \delta(q_1,a)=q_0, & \delta(q_2,a)=q_0, & \delta(q_3,a)=q_4, & \delta(q_4,a)=q_4,\\
\delta(q_0,b)=q_0, & \delta(q_1,b)=q_2, & \delta(q_2,b)=q_0, & \delta(q_3,b)=q_0, & \delta(q_4,b)=q_4,\\
\delta(q_0,c)=q_0, & \delta(q_1,c)=q_0, & \delta(q_2,c)=q_3, & \delta(q_3,c)=q_0, & \delta(q_4,c)=q_4,\\
\delta(q_0,d)=q_1, & \delta(q_1,d)=q_1, & \delta(q_2,d)=q_1, & \delta(q_3,d)=q_4, & \delta(q_4,d)=q_4.
\end{array}
$$

Doporučený úkol. Daný DKA graficky znázorněte.

Definice

Množinu všech řetězců (slov) $w \in T^*$ přijímaných DKA nazveme **jazykem** $L(A)$ rozpoznávaným tímto automatem.

Příklad

Pro abecedu $T = \{a, b\}$ vytvořte DKA,

(a) který přijímá pouze dva řetězce: a, ba, tedy $L(A) = \{a, ba\}$.

(b) jehož $L(A) = \{a^n; n \in \mathbb{N}_0\} \cup \{(ba)^n; n \in \mathbb{N}\}$.

Řešení. Na přednášce.

Tvrzení

Množina všech řetězců (slov), které daný DKA přijme, tvoří tzv. **regulární jazyk**.

Více o tom na přednášce a na cvičení.

Obsah

Algebraická struktura

Definice

Nechť $A \neq \emptyset$ a $n \in \mathbb{N}_0$. Pak **n-ární operací na A** rozumíme každé zobrazení $f : A^n \to A$.

Definice

Neprázdnou množinu A spolu s neprázdnou množinou operací na A nazveme **algebraická struktura** (nebo stručněji **algebra**). Označení algebry s k operacemi: $\mathscr{A} = (A; f_1, f_2, \ldots, f_k)$.

Algebraická struktura je tedy každá neprázdná množina, na které jsou definovány nějaké operace, přičemž daná množina je vzhledem ke všem těmto operacím **uzavřená** (výsledkem operace s prvky této množiny je vždy prvek z této množiny).

My se zaměříme na případ, kdy na dané neprázdné množině bude pouze jedna binární operace (později pak dvě různé binární operace).

Grupoid

Definice

Binární operací na množině $A \neq \emptyset$ nazveme každé zobrazení $\circ : A \times A \to A$.

Definice

Grupoid je algebraická struktura $\mathscr{A} = (A; \circ)$, kde $\circ$ je binární operace na $A \neq \emptyset$.

Úmluva. Bude-li $\circ$ některá binární operace na A, pak budeme místo $\circ(a, b)$ zapisovat $a \circ b$.

Příklad

Na množině $\mathbb{N}$ všech přirozených čísel zřejmě $+$ přiřadí každé dvojici čísel $a, b \in \mathbb{N}$ číslo $a + b \in \mathbb{N}$. Je tedy $+$ binární operace na $\mathbb{N}$ a $(\mathbb{N}; +)$ je grupoid. Kupříkladu také $(\mathbb{N}; \cdot)$ je grupoid. Naproti tomu $(\mathbb{N}; -)$, ani $(\mathbb{N}; :)$ grupoidy nejsou.

Definice

- Grupoid $\mathscr{A} = (A; \circ)$ se nazývá **komutativní**, platí-li

$$\forall a, b \in A: \quad a \circ b = b \circ a,$$

  tedy operace $\circ$ je komutativní.
- Grupoid $\mathscr{A} = (A; \circ)$ se nazývá **pologrupa**, platí-li

$$\forall a, b, c \in A: \quad a \circ (b \circ c) = (a \circ b) \circ c,$$

  tedy jeho operace je asociativní.
- Grupoid $\mathscr{A} = (A; \circ)$ má **jednotkový prvek** e, platí-li

$$\exists e \in A \quad \forall a \in A: \quad a \circ e = a = e \circ a.$$

- Pologrupa v níž existuje jednotkový prvek, se nazývá **monoid**.

Příklady na přednášce a na cvičení.

Terminologická poznámka. Pokud je operace v grupoidu zapsána symbolem $+$, nazývá se grupoid $(A;+)$ **aditivní**. Je-li operace zapisována symbolem $\cdot$ (nebo, když je vynechávána), nazývá se příslušný grupoid $(A;\cdot)$ **multiplikativní**.

Tvrzení

Každý grupoid má nejvýše jeden jednotkový prvek.

Důkaz. Nechť e, f jsou dva jednotkové prvky v grupoidu $\mathscr{A} = (A;\circ)$. Pak $e = e \circ f = f$.

Definice

Nechť $\mathscr{A} = (A;\circ)$ je grupoid, nechť $\emptyset \neq B \subseteq A$. Jestliže $\forall a, b \in B$ platí $a \circ b \in B$, nazývá se $(B;\circ)$ **podgrupoid** grupoidu $\mathscr{A}$.

Příklad

Zřejmě $(\mathbb{N};+)$ je podgrupoid $(\mathbb{Z};+)$. A třeba $(\{0,1\};\cdot)$ je podgrupoid $(\mathbb{Q};\cdot)$.

Homomorfismy a izomorfismy grupoidů

Budeme se zabývat důležitými zobrazeními, která „zachovávají“ binární operace při „přechodu“ mezi dvěma grupoidy.

Definice

- Jsou-li $\mathscr{G} = (G; \circ)$ a $\mathscr{H} = (H; \star)$ grupoidy, pak se zobrazení $f : G \to H$ nazývá **homomorfismus** grupoidu $\mathscr{G}$ do grupoidu $\mathscr{H}$, platí-li

$$\forall a, b \in G: \quad f(a \circ b) = f(a) \star f(b).$$

- Jestliže f je navíc bijektivní, pak se nazývá **izomorfismus** grupoidu $\mathscr{G}$ na grupoid $\mathscr{H}$. V tom případě říkáme, že grupoid $\mathscr{H}$ **je izomorfní** s grupoidem $\mathscr{G}$.
- Řekneme, že grupoid $\mathscr{H}$ je **homomorfním obrazem** grupoidu $\mathscr{G}$, existuje-li surjektivní homomorfismus $\mathscr{G}$ na $\mathscr{H}$.

Poznámka. Jestliže f je homomorfismus grupoidu $\mathscr{G}$ do grupoidu $\mathscr{H}$ a g je homomorfismus grupoidu $\mathscr{H}$ do grupoidu $\mathscr{K}$, pak jejich složení „g po f“ je homomorfismem $\mathscr{G}$ do $\mathscr{K}$.
Podobně: složení dvou izomorfismů grupoidů je izomorfismem.
Dále zřejmě identické zobrazení id_G je izomorfismem $\mathscr{G}$ na $\mathscr{G}$ a inverzní zobrazení f^{-1} k izomorfismu f grupoidu $\mathscr{G}$ na grupoid $\mathscr{H}$ je izomorfismem $\mathscr{H}$ na $\mathscr{G}$.
Relace „být izomorfní s“ je tedy ekvivalencí na třídě všech grupoidů, a proto indukuje rozklad třídy všech grupoidů na třídy navzájem izomorfních grupoidů. Grupoidy, které patří do téže třídy rozkladu, mají stejné algebraické vlastnosti.

Poznámka. Protože relace „být izomorfní s“ je symetrická, můžeme v případě, kdy grupoid $\mathscr{H}$ je izomorfní s grupoidem $\mathscr{G}$, říkat také, že grupoidy $\mathscr{G}$ a $\mathscr{H}$ jsou (navzájem) izomorfní.
Označení: $\mathscr{G} \cong \mathscr{H}$.

Příklad

Uvažujme grupoidy $\mathscr{N} = (\mathbb{N}; +)$ a $2\mathscr{N} = (2\mathbb{N}; +)$, kde $2\mathbb{N} = \{2n;\, n \in \mathbb{N}\}$. Pak zobrazení $f : \mathbb{N} \to \mathbb{N}$ takové, že $\forall a \in \mathbb{N}$ je $f(a) = 2a$, je homomorfismus $\mathscr{N}$ do $\mathscr{N}$, který není surjektivní. Označme $\bar{f} : \mathbb{N} \to 2\mathbb{N}$ zobrazení, v němž opět platí, že $\forall a \in \mathbb{N}$ je $\bar{f}(a) = 2a$. Pak $\bar{f}$ je izomorfismus $\mathscr{N}$ na $2\mathscr{N}$, tedy $\mathscr{N} \cong 2\mathscr{N}$.

Příklad

Ukažme, že grupoid $\mathscr{A} = (\{-1, 1\}; \cdot)$ je homomorfním obrazem grupoidu $\mathscr{Z} = (\mathbb{Z}; +)$. Uvažujme zobrazení $f : \mathbb{Z} \to \{-1, 1\}$ takové, že

$$f(a) = \begin{cases} 1 & \text{pro } a \in 2\mathbb{Z} = \{2k;\, k \in \mathbb{Z}\} \\ -1 & \text{jinak (tedy pro } a \in \{2k-1;\, k \in \mathbb{Z}\}). \end{cases}$$

Snadno lze ověřit, že f je homomorfismus $\mathscr{Z}$ na $\mathscr{A}$. Přitom je zřejmé, že $\mathscr{Z}$ a $\mathscr{A}$ nejsou izomorfní (píšeme $\mathscr{Z} \ncong \mathscr{A}$).

Příklad

Uvažujme grupoidy $\mathscr{R} = (\mathbb{R}; +)$ a $\mathscr{R}_1 = (\mathbb{R}; \star)$, kde $\forall a, b \in \mathbb{R}$ je $a \star b = a + b + 3$. Grupoidy $\mathscr{R}$ a $\mathscr{R}_1$ jsou izomorfní, protože zobrazení $f : \mathbb{R} \to \mathbb{R}$ takové, že $\forall a \in \mathbb{R}$ je $f(a) = a - 3$, je izomorfismem $\mathscr{R}$ na $\mathscr{R}_1$. Oba grupoidy tak musí mít stejné vlastnosti.

Příklad

Dokažte, že grupoidy $\mathscr{N}_1 = (\mathbb{N}; \cdot)$ a $2\mathscr{N}_1 = (2\mathbb{N}; \cdot)$ nejsou izomorfní.

Řešení.
Sporem. Nechť f je izomorfismus $\mathscr{N}_1$ na $2\mathscr{N}_1$ a nechť $f(1) = 2x$. Pak

$$2x = f(1) = f(1 \cdot 1) = f(1) \cdot f(1) = 2x \cdot 2x = 4x^2,$$

což ale neplatí pro žádné $x \in \mathbb{N}$, spor. Proto izomorfismus f neexistuje.

Definice

Nechť grupoid $(A;\cdot)$ má jednotkový prvek e, nechť $a \in A$. Pak se prvek $b \in A$ nazývá **inverzní** k prvku a, platí-li $ab = e = ba$.

Poznámka. V monoidu má každý prvek a nejvýše jeden inverzní prvek, který označujeme a^{-1}.

Poznámka. Platí, že při homomorfismu f grupoidu $\mathscr{G}$ na grupoid $\mathscr{H}$ se jednotkový prvek e zobrazí na jednotkový prvek $f(e)$ a $f(a^{-1}) = (f(a))^{-1}$.

Definice

Grupa je libovolný monoid, v němž má každý jeho prvek inverzní prvek. Komutativní grupu $(G;\cdot)$, tedy takovou, kde $\forall a,b \in G$: $ab = ba$, budeme také nazývat **abelovská grupa**.

Poznámka. Podle uvedené definice tedy platí, že grupoid $(G;\cdot)$ je grupou, právě když

1. $\forall a,b,c \in G: \quad a(bc) = (ab)c$
2. $\exists e \in G \quad \forall a \in G: \quad ae = ea = a$
3. $\forall a \in G \quad \exists a^{-1} \in G: \quad aa^{-1} = a^{-1}a = e.$

Ze třetí podmínky $\forall a \in G$ plyne, že $(a^{-1})^{-1} = a$.

Poznámka. Budeme-li grupu $(G,\circ)$ zapisovat v aditivním tvaru, tedy $(G;+)$, pak její jednotkový prvek budeme značit symbolem 0 a inverzní prvek k prvku $a \in G$ budeme značit $-a$. Prvek $-a$ budeme nazývat **prvek opačný** k prvku a. Často místo $a+(-b)$ budeme zkráceně psát $a-b$.

Definice

Jestliže $\mathscr{G} = (G; \cdot)$ je taková grupa, že G má n prvků ($n \in \mathbb{N}$), pak řekneme, že grupa $\mathscr{G}$ má **konečný řád n**. Je-li množina G nekonečná, pak je grupa $\mathscr{G}$ **nekonečného řádu**.

Příklady

- Algebra $(\{0, 1\}; \leftrightarrow)$, kde $\leftrightarrow$ je booleovská spojka ekvivalence, je abelovská grupa konečného řádu 2.
- $(\mathbb{Z}; +)$ je abelovská grupa nekonečného řádu s jednotkovým prvkem 0.
- Nechť $\mathbb{R}^+$ je množina všech kladných reálných čísel. Pak $(\mathbb{R}^+; \cdot)$ je abelovská grupa nekonečného řádu s jednotkovým prvkem 1.

Příklad

Dokažte, že všechny zákrytové pohyby

(a) obdélníku

(b) rovnoramenného trojúhelníku, který není rovnostranný

(c) rovnostranného trojúhelníku

tvoří grupu vzhledem k operaci skládání zobrazení. Je tato grupa abelovská?

Obecně k řešení. Ve všech třech případech bude jednotkovým prvkem identické zobrazení ponechávající všechny body na místě. Dále není třeba ověřovat asociativitu, neboť ji splňuje každé zobrazení. Tím, že se jedná o zákrytové pohyby máme zaručenu uzavřenost, tedy to, že dané skládání bude vskutku operací na příslušné množině zákrytových pohybů. Pro určení (jednoznačné) existence inverzních prvků (ke všem zákrytovým pohybům) vyplníme odpovídající Cayleyho tabulky. Z nich pak okamžitě určíme, je-li operace skládání komutativní, nebo ne.

Řešení (a)

Obdélník má čtyři zákrytové pohyby: id (identitu), r (rotaci o 180°), a dvě osové souměrnosti o_1 a o_2. Tyto čtyři zákrytové pohyby tvoří vzhledem ke skládání zobrazení $\circ$ abelovskou (komutativní) grupu. Příslušná Cayleyho tabulka pro operaci $\circ$ vypadá takto:

$\circ$	id	r	o_1	o_2
id	id	r	o_1	o_2
r	r	id	o_2	o_1
o_1	o_1	o_2	id	r
o_2	o_2	o_1	r	id

Řešení (b)

Rovnoramenný trojúhelník, který není rovnostranný má právě dva zákrytové pohyby: *id* (identitu) a osovou souměrnost o_1. Tyto dva zákrytové pohyby tvoří vzhledem ke skládání zobrazení $\circ$ abelovskou grupu. Příslušná Cayleyho tabulka pro operaci $\circ$ vypadá takto:

$\circ$	id	o_1
id	id	o_1
o_1	o_1	id

Řešení (c)

Rovnostranný trojúhelník má celkem šest zákrytových pohybů: id (identitu), r_1 (rotaci o 120°), r_2 (rotaci o 240°) a tři osové souměrnosti o_1, o_2 a o_3. Těchto šest zákrytových pohybů tvoří vzhledem ke skládání zobrazení $\circ$ grupu, která není abelovská.[a] Příslušná Cayleyho tabulka pro operaci $\circ$ vypadá takto:

$\circ$	id	r_1	r_2	o_1	o_2	o_3
id	id	r_1	r_2	o_1	o_2	o_3
r_1	r_1	r_2	id	o_2	o_3	o_1
r_2	r_2	id	r_1	o_3	o_1	o_2
o_1	o_1	o_3	o_2	id	r_2	r_1
o_2	o_2	o_1	o_3	r_1	id	r_2
o_3	o_3	o_2	o_1	r_2	r_1	id

[a] Nejmenší neabelovská grupa je šestiprvková a je (až na izomorfismus) právě uvedenou grupou.

V posledním příkladě jsme viděli tři konkrétní grupy konečných řádů (grupu řádu čtyři, grupu řádu dva a grupu řádu šest).

Příklad*

Uveďte příklad abelovské a neabelovské grupy nekonečného řádu.

Nástin řešení.

- Množina všech zákrytových rotačních pohybů jednotkové kružnice tvoří vzhledem ke skládání zobrazení abelovskou grupu nekonečného řádu. Jednotkovým prvkem je identické zobrazení *id*, které odpovídá rotaci o nula stupňů.
- Množina všech regulárních čtvercových matic stupně 2 (nad $\mathbb{R}$) tvoří vzhledem k násobení matic nekomutativní grupu nekonečného řádu. Jednotkovým prvkem je v tomto případě matice

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Připomeňme, že na množině zbytkových tříd $\mathbb{Z}_n = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$ lze zavést binární operace $\oplus$ a $\odot$.
Pro libovolné $a, b \in \mathbb{Z}_n$ definujeme:

$$a \oplus b = a + b \pmod{n}, \qquad a \odot b = a \cdot b \pmod{n}.$$

Příklad

Algebry $\mathscr{G} = (\mathbb{Z}_5; \oplus)$ a $\mathscr{H} = (\mathbb{Z}_5 \setminus \{\overline{0}\}; \odot)$ tvoří konečné abelovské grupy řádu 5 a 4. Následují příslušné Cayleyho tabulky:

$\oplus$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$
$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$
$\overline{1}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{0}$
$\overline{2}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{0}$	$\overline{1}$
$\overline{3}$	$\overline{3}$	$\overline{4}$	$\overline{0}$	$\overline{1}$	$\overline{2}$
$\overline{4}$	$\overline{4}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$

$\odot$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$
$\overline{1}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$
$\overline{2}$	$\overline{2}$	$\overline{4}$	$\overline{1}$	$\overline{3}$
$\overline{3}$	$\overline{3}$	$\overline{1}$	$\overline{4}$	$\overline{2}$
$\overline{4}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$

Jednotkovým prvkem v $\mathscr{G}$ je zřejmě $\overline{0}$ a v $\mathscr{H}$ prvek $\overline{1}$.

Věta

Nechť $\mathscr{G} = (G; \circ)$ je grupa. Pak pro každé dva prvky $a, b \in G$ existují $x, y \in G$ tak, že platí $a \circ x = b$, $y \circ a = b$.

Důkaz. Nechť $a, b \in G$. Položme $x = a^{-1} \circ b$, $y = b \circ a^{-1}$. Pak

$$a \circ x = a \circ (a^{-1} \circ b) = (a \circ a^{-1}) \circ b = e \circ b = b,$$

$$y \circ a = (b \circ a^{-1}) \circ a = b \circ (a \circ a^{-1}) = b \circ e = b.$$

Poznámka. Následující věta umožňuje definovat grupu jiným způsobem.

Věta

Pologrupa $\mathscr{G} = (G; \circ)$ je grupou, právě když pro každé $a, b \in G$ jsou v $\mathscr{G}$ řešitelné rovnice $a \circ x = b$, $y \circ a = b$.

Důkaz. Dle předchozí věty dokážeme jen obrácenou implikaci.

(i) Nechť $a \in G$. Pak existují prvky $e, f \in G$ tak, že $a \circ e = a$, $f \circ a = a$. Nechť dále $x \in G$ je libovolný prvek. Pak existuje $y \in G$ tak, že $x = y \circ a$, tedy

$$x \circ e = (y \circ a) \circ e = y \circ (a \circ e) = y \circ a = x.$$

Analogicky lze dokázat $f \circ x = x$.
Volbou $x = f$ obdržíme $f \circ e = f$. Pro $x = e$ dostaneme $f \circ e = e$, tedy $e = f \circ e = f$. Dohromady, v G existuje prvek e takový, že $e \circ x = x = x \circ e$ pro každé $x \in G$, tedy e je jednotkovým prvkem $\mathscr{G}$.

(ii) Z předpokladu plyne, že pro každé $a \in G$ existují $x, y \in G$ tak, že $a \circ x = e$, $y \circ a = e$. Potom

$$x = e \circ x = (y \circ a) \circ x = y \circ (a \circ x) = y \circ e = y,$$

tedy $x = y$. Odtud $x = a^{-1}$ je prvek inverzní k a.

Poznámka. Předchozí věta nám dává užitečné kritérium pro určování toho, jestli je konečný grupoid grupou. Je-li totiž dána Cayleyova tabulka grupy konečného řádu, pak se v každém řádku a v každém sloupci musí vyskytovat všechny její prvky. Tato podmínka je nutná, ale není postačující, protože například nezaručuje asociativnost operace.

Doporučený úkol. Vymyslete příklad konečného grupoidu, který není grupou a přitom uvedené kritérium splňuje.

Definice

Řekneme, že v grupoidu $(G;\cdot)$ platí **pravidlo o krácení**, jestliže pro každé prvky $a,b,c,d \in G$ platí

$$ca = cb \quad \Rightarrow \quad a = b,$$
$$ad = bd \quad \Rightarrow \quad a = b.$$

Věta

V každé grupě platí pravidlo o krácení.[a]

[a] Dříve zmíněné rovnice $ax = b$ a $ya = b$ jsou tak v každé grupě řešitelné jednoznačně.

Komentář k důkazu na přednášce.

Cyklická grupa

Definice

Grupa $\mathscr{G} = (G; \cdot)$ se nazývá **cyklická**, jestliže existuje prvek $a \in G$ takový, že $G = \{a^k \mid k \in \mathbb{Z}\}$. Prvek a se nazývá **generátor** cyklické grupy. Označení: $G = \langle a \rangle$.

Pojmem cyklická grupa se tedy označuje grupa, která může být „generována“ operováním s jedním jediným prvkem (jejím generátorem).

Příklad

Cyklická grupa může mít více než jeden generátor. Například grupa $(\mathbb{Z}_5; \oplus)$ všech celých čísel modulo 5 se sčítáním modulo 5 má čtyři generátory: 1, 2, 3 a 4.

Definice

Nechť $\mathscr{G} = (G; \cdot)$ je grupa a nechť $a \in G$. Jestliže $n \in \mathbb{N}$, pak $(-\mathbf{n})$**-tou mocninou prvku a** rozumíme prvek $a^{-n} \in G$ takový, že $a^{-n} = (a^n)^{-1}$.

Poznámka. Pro každé $n \in \mathbb{N}$ platí $a^{-n} = (a^{-1})^n$.

Věta

Jestliže jsou a, b prvky grupy $(G; \cdot)$ a $m, n \in \mathbb{Z}$, pak platí

(a) $a^m \cdot a^n = a^{m+n}$

(b) $(a^m)^n = a^{mn}$

(c) jestliže $ab = ba$, pak $(ab)^n = a^n b^n$.

Definice

Nechť $\mathscr{G} = (G; \cdot)$ a $\mathscr{G}' = (G'; \star)$ jsou grupy a $f : G \to G'$ zobrazení. Pak se f nazývá **homomorfismus grupy $\mathscr{G}$ do grupy $\mathscr{G}'$**, jestliže pro každé $a, b \in G$ platí $f(a \cdot b) = f(a) \star f(b)$. Je-li homomorfismus f bijektivní, pak se nazývá **izomorfismus $\mathscr{G}$ na $\mathscr{G}'$**.

Poznámka. Homomorfismus grupy $\mathscr{G}$ do grupy $\mathscr{G}'$ se definuje stejně jako homomorfismus grupoidu $\mathscr{G}$ do grupoidu $\mathscr{G}'$; vyžaduje se po něm jen přenášení binární operace násobení.[26]

[26] Grupu lze chápat také jako algebraickou strukturu s jednou binární, jednou nulární a jednou unární operací. Použitá definice homomorfismu f je však dostatečná, neboť f přenáší i jednotkový prvek a prvky inverzní.

Poznámka. Připomeňme, že grupoidy $\mathscr{G}$ a $\mathscr{G}'$ se nazývají izomorfní (označení $\mathscr{G} \cong \mathscr{G}'$), existuje-li alespoň jeden izomorfismus jednoho z nich na druhý. Přitom relace „být izomorfní“ je relací ekvivalence na třídě všech grupoidů. Grupoidy, které patří do téže třídy odpovídajícího rozkladu, mají stejné algebraické vlastnosti. Pro grupy jsme definovali pojem izomorfismu stejně jako pro grupoidy, proto také každá grupa jednoznačně patří do některé třídy uvedeného rozkladu a platí, že všechny grupoidy, které jsou izomorfní s danou grupou, jsou také grupami.

Příklad

Uvažujme grupy $(\mathbb{R}^+;\cdot)$ a $(\mathbb{R};+)$ a zobrazení $\log : \mathbb{R}^+ \to \mathbb{R}$ takové, že $\log : x \mapsto \log x$. Je zřejmé, že log je bijektivní zobrazení $\mathbb{R}^+$ na $\mathbb{R}$, a je známo, že pro každé $x, y \in \mathbb{R}^+$ platí $\log(xy) = \log x + \log y$. Tedy $(\mathbb{R}^+;\cdot) \cong (\mathbb{R};+)$.

Faktorové struktury (získané přes kongruence)

Důležitá úloha — vytvořit homomorfní obrazy dané algebraické struktury — se nejjednodušeji řeší přes konstrukce odpovídajících faktorových algebraických struktur. V případě grupoidů lze ke konstrukci faktorových struktur použít univerzální metodu založenou na pojmu kongruence. Pro grupy, které jsou speciálním případem grupoidů, tato konstrukce také funguje.[27]

[27] U grup lze postupovat (faktorizovat) i přes tzv. normální podgrupy.

Definice

Nechť $\mathscr{G} = (G; \cdot)$ je grupoid. Pak **kongruencí** grupoidu $\mathscr{G}$ rozumíme každou relaci ekvivalence ρ na G, pro kterou je splněna podmínka

$$\forall a, b, c, d \in G : (\langle a, b \rangle \in \rho \wedge \langle c, d \rangle \in \rho) \Rightarrow \langle ac, bd \rangle \in \rho .$$

Příklad

Jestliže $n \in \mathbb{N}$, pak relace kongruence podle modulu n je grupoidovou kongruencí na grupoidu $\mathscr{Z}' = (\mathbb{Z}; \cdot)$. Vskutku, nechť $a, b, c, d \in \mathbb{Z}$, $a \equiv b \pmod{n}$, $c \equiv d \pmod{n}$. Pak víme, že i $ac \equiv bd \pmod{n}$.

Věta o homomorfismu grup

Jestliže ρ je kongruence grupoidu $\mathscr{G} = (G; \cdot)$ a jestliže pro libovolné $a, b \in G$ položíme $[a]_\rho \cdot [b]_\rho = [a \cdot b]_\rho$, pak $(G/\rho; \cdot)$ je grupoid.

Definice

Grupoid $\mathscr{G}/\rho = (G/\rho; \cdot)$ z předchozí věty se nazývá **faktorový grupoid** grupoidu $\mathscr{G}$ podle kongruence ρ.

Příklad

Uvažujeme grupoid $\mathscr{Z}' = (\mathbb{Z}; \cdot)$ a kongruenci podle modulu n, kde $n \in \mathbb{N}$. Pak faktorovým grupoidem $\mathscr{Z}'$ podle této kongruence je množina všech zbytkových tříd podle modulu n. Přitom například pro $n = 4$ je Cayleyova tabulka pro násobení $\odot$ ve faktorovém grupoidu následující:

$\odot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$
$\overline{1}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
$\overline{2}$	$\overline{0}$	$\overline{2}$	$\overline{0}$	$\overline{2}$
$\overline{3}$	$\overline{0}$	$\overline{3}$	$\overline{2}$	$\overline{1}$

Poznámka. Jestliže $\mathscr{G} = (G; \cdot)$ je grupa, pak grupovou kongruencí grupy $\mathscr{G}$ rozumíme právě každou kongruenci grupoidu $(G; \cdot)$. Proto přívlastek „grupová“ můžeme vynechávat.

Poznámka. Dá se dokázat, že faktorový grupoid grupy je vždy grupou.

Poznámka. Nechť $\mathscr{A}$ je některá třída grup. Jestliže platí, že danou vlastnost mají právě všechny grupy z třídy $\mathscr{A}$ a všechny grupy, které jsou izomorfní s některou grupou z $\mathscr{A}$, pak říkáme, že tuto vlastnost „**mají, až na izomorfismus**“, právě grupy z třídy $\mathscr{A}$. Platí, že homomorfními obrazy grupy $\mathscr{G}$ jsou, až na izomorfismus, právě všechny faktorové grupy podle kongruence ρ.

Cyklické grupy a jejich vlastnosti

Připomeňme, že grupa $(G;\cdot)$ je cyklická, pokud v ní existuje prvek $a \in G$ takový, že $G = \langle a \rangle$. Příklady cyklických grup jsou $(\mathbb{Z};+)$, kde $\mathbb{Z} = \langle 1 \rangle = \langle -1 \rangle$, popřípadě $(\mathbb{Z}_n;\oplus)$ pro $n \in \mathbb{N}$, kde vždy $\mathbb{Z}_n = \langle 1 \rangle$. Existují tedy cyklické grupy nekonečného řádu i libovolného konečného řádu.

Věta

- Každá nekonečná cyklická grupa je izomorfní s grupou $\mathscr{Z} = (\mathbb{Z}; +)$.
- Každá konečná cyklická grupa řádu n je izomorfní s grupou $\mathscr{Z}_n = (\mathbb{Z}_n; \oplus)$.

Vidíme tedy, že, až na izomorfismus, existuje jediná cyklická grupa nekonečného řádu, a to $\mathscr{Z} = (\mathbb{Z}; +)$. Podobně pro libovolné $n \in \mathbb{N}$ existuje cyklická grupa konečného řádu n, která je také, až na izomorfismus, určena jednoznačně.

Věta

Každá grupa prvočíselného řádu je cyklická, přičemž pro každé prvočíslo p existuje, až na izomorfismus, právě jedna grupa řádu p.

K aplikaci cyklických grup v kryptografii 1/3

Alice a Bob se chtějí domluvit na společném tajném klíči k, s nímž by šifrovali své zprávy. Využijí teorii grup a veřejným kanálem se domluví na vhodné cyklické grupě $(G;\cdot)$ prvočíselného řádu p, kterou generuje prvek a. Jelikož je p prvočíslo, tak lze každý nenulový prvek z grupy $\langle a \rangle$ vyjádřit jako přirozenou mocninu prvku a.

Předpokládejme, že Alice si zvolí číslo $m \in \{1,2,\ldots,|G|-1\}$ a v dané grupě vypočítá číslo $x = a^m$, které pošle veřejně Bobovi. Podobně si vybere Bob číslo $n \in \{1,2,\ldots,|G|-1\}$ a v $(G;\cdot)$ dopočítá číslo $y = a^n$, které pošle veřejně Alici.

Společným tajným klíčem Alice a Boba je číslo $k = a^{mn}$. Alice jej získá, když Bobovo číslo $y = a^n$ umocní na m-tou. Bob klíč získá umocněním Alicina čísla $x = a^m$ na n-tou.[28]

[28] Šifrování zprávy $z \in G$ probíhá vynásobením klíčem k, $s = z \cdot k$ v grupě $(G;\cdot)$. Dešifruje se vynásobením inverzním prvkem ke k, $z = s \cdot k^{-1}$ v $(G;\cdot)$.

K aplikaci cyklických grup v kryptografii 2/3

Nepřítel (Eva) má k dispozici generátor a cyklické grupy $(G; \cdot)$, kterou také zná. Navíc zná i čísla $x = a^m$, $y = a^n$.[29] Z těchto informací chce získat tajný klíč Alice a Boba: číslo a^{mn}.

V současné době není znám rychlý (polynomiální) algoritmus, jak ve výše popsané grupě získat hodnoty m a n z čísel $x = a^m$ a $y = a^n$. Jedná se o tzv. problém diskrétního logaritmu. Přesněji: ke zjištění tajného klíče je třeba vyřešit tzv. Diffie-Hellmanův problém: ze znalosti a^m a a^n spočítat a^{mn}. To se zatím umí pouze přes diskrétní logaritmus, tedy přes vypočítání m a n. Bezpečnost šifrování se tak opírá o exponenciální časovou složitost výpočtu diskrétního logaritmu.

Příklad na cvičení.

[29] Eva z hodnot $x = a^m$ a $y = a^n$ snadno získá číslo $x \cdot y = a^m \cdot a^n = a^{m+n}$, které jí je ale k ničemu.

K aplikaci cyklických grup v kryptografii 3/3

Definice

Nechť $(G;\cdot)$ je cyklická grupa řádu n s generátorem a. Pak každý prvek $b \in G$ lze zapsat ve tvaru $b = a^k$ pro jediné $k \in \mathbb{Z}_n$. Číslo k se nazývá **diskrétní logaritmus** o základu a z prvku b v grupě $(G;\cdot)$ a značí se $d\log_a(b)$.

Příklad na cvičení.

Závěrem k teorii grup 1/2

Grupa je algebraická struktura, která popisuje a formalizuje koncept symetrie. Matematická disciplína zabývající se studiem grup se nazývá teorie grup.[30] Teorie grup vznikla počátkem 19. století. U jejího zrodu stál matematik Évariste Galois, který dokázal, že polynomiální rovnice (stupně vyššího než 4) nelze obecně řešit pomocí vzorců, ve kterých se vyskytují jen operace sčítání, odčítání, násobení, dělení a odmocňování.

Grupy našly později uplatnění také v geometrii, teorii čísel, algebraické topologii i ve fyzice, informatice a chemii. Reprezentace grup hrají důležitou úlohu v částicové fyzice, kvantové teorii pole anebo v teorii strun. Chemie používá grupy pro popis symetrií molekul a krystalových mřížek v krystalografii.

[30] Klasifikace jednoduchých konečných grup byla dokončena koncem 20. století a patří k největším výsledkům matematiky vůbec.

Závěrem k teorii grup 2/2

V informatice se grupy vyskytují například při zpracování obrazu, v kódování nebo v kryptografii (souvisí třeba s diskrétním logaritmem). Konečné grupy symetrií (jako například Mathiovy grupy) se využívají v kódování a v korekci chyb přenášených dat. Multiplikativní grupy konečných těles (definice viz dále) se využívají v cyklickém kódování, které se používá například v CD přehrávačích.

Převzato (a upraveno) z: http://cs.wikipedia.org/wiki/Grupa.

Obsah

Definice

Algebraická struktura $\mathscr{R} = (R; +, \cdot)$ s binárními operacemi $+$ a $\cdot$ se nazývá **okruh**, pokud

(i) $(R; +)$ je abelovská grupa (0 její jednotkový prvek)

(ii) $(R; \cdot)$ je pologrupa

(iii) platí **distributivní zákony**[a]: pro každé $a, b, c \in R$:

$$a \cdot (b + c) = a \cdot b + a \cdot c, \qquad (b + c) \cdot a = b \cdot a + c \cdot a.$$

Okruh $\mathscr{R}$ se nazývá **komutativní**, jestliže $a \cdot b = b \cdot a$ pro každé $a, b \in R$. Okruh $\mathscr{R}$ se nazývá **unitární**, má-li pologrupa $(R; \cdot)$ jednotkový prvek (**jednotku**). Je-li $\mathscr{R}$ unitární, budeme jeho jednotku označovat 1. Prvek 0 se nazývá **nulou okruhu** $\mathscr{R}$.

[a] Násobení je distributivní zleva i zprava vzhledem ke sčítání.

Příklad

Komutativní unitární okruhy jsou například: okruh celých čísel $(\mathbb{Z};+,\cdot)$, okruh racionálních čísel $(\mathbb{Q};+,\cdot)$, okruh reálných čísel $(\mathbb{R};+,\cdot)$ a okruh komplexních čísel $(\mathbb{C};+,\cdot)$. Komutativním okruhem (který není unitární) je okruh $(2\mathbb{Z};+,\cdot)$.

Poznámka. Název nula okruhu pro prvek 0 je oprávněný, neboť je nulou pologrupy $(R;\cdot)$, tedy prvkem, který pro libovolné $a \in R$ splňuje: $a \cdot 0 = 0 = 0 \cdot a$. Vskutku, dle distributivních zákonů platí $\forall a \in R$:

$$a \cdot a = a \cdot (a+0) = a \cdot a + a \cdot 0,$$

$$a \cdot a = (a+0) \cdot a = a \cdot a + 0 \cdot a,$$

avšak $(R;+)$ je abelovská grupa, tedy $a \cdot 0 = 0 = 0 \cdot a$. Prvku 0 se někdy říká **agresívní prvek** pologrupy $(R;\cdot)$.

Pravidla pro počítání v okruzích 1/2

Mějme dán okruh $\mathscr{R} = (R; +, \cdot)$.
Jak jsme již ukázali, $\forall a \in R$ platí $a \cdot 0 = 0 = 0 \cdot a$.
Ověříme, že $\forall a, b \in R$ platí $a \cdot (-b) = (-a) \cdot b = -a \cdot b$.
Totiž, $a \cdot (-b) + a \cdot b = a \cdot (-b + b) = a \cdot 0 = 0$, odkud
$a \cdot (-b) = -a \cdot b$, analogicky se dá ukázat, že $(-a) \cdot b = -a \cdot b$.
V unitárním okruhu navíc $\forall a \in R$ platí $a \cdot (-1) = (-1) \cdot a = -a$.

Nechť $\mathscr{R} = (R; +, \cdot)$ je komutativní okruh. Jelikož $(R; +)$ je grupa (tedy $+$ je asociativní), nemusíme součty ve tvaru $a_1 + a_2 + a_3 + \cdots + a_n$ závorkovat. Jsou-li $a_1, \ldots, a_n \in R$ budeme používat tzv. **sumační symbol**

$$a_1 + a_2 + a_3 + \cdots + a_n = \sum_{i=1}^{n} a_i.$$

Číslo i nazveme **součtový index**.

Pravidla pro počítání v okruzích 2/2

Snadno lze (použitím asociativního a komutativního zákona a distributivních zákonů) ověřit platnost následujících rovností:

(i)
$$\sum_{i=1}^{m} a_i + \sum_{i=m+1}^{n} a_i = \sum_{i=1}^{n} a_i$$

(ii)
$$\sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i = \sum_{i=1}^{n} (a_i + b_i)$$

(iii)
$$c \cdot \sum_{i=1}^{n} a_i = \sum_{i=1}^{n} c \cdot a_i$$

(iv)
$$\Big(\sum_{i=1}^{m} a_i\Big) \cdot \Big(\sum_{i=1}^{n} b_j\Big) = \sum_{i=1}^{m} \Big(a_i \cdot \sum_{j=1}^{n} b_j\Big) = \sum_{i=1}^{m} \Big(\sum_{j=1}^{n} a_i \cdot b_j\Big)$$

(v)
$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij}.$$

Definice

Prvek a okruhu $\mathscr{R} = (R; +, \cdot)$ se nazývá **netriviální dělitel nuly**, jestliže $a \neq 0$ a existuje $b \neq 0$, $b \in R$ tak, že $a \cdot b = 0$.

Příklad

Nechť A je množina všech funkcí jedné reálné proměnné na intervalu $\langle 0, 1 \rangle$, nechť $+$ značí operaci pro sčítání funkcí a $\cdot$ značí operaci pro násobení funkcí. Pak $\mathscr{A} = (A; +, \cdot)$ je komutativní unitární okruh, kde jednotkou je konstantní funkce $f(x) = 1$. Tento okruh má netriviální dělitele 0: nechť $g(x)$ je funkce: $g(x) = 0$ pro $x \in \langle 0, \frac{1}{2} \rangle$, $g(x) \neq 0$ pro $x \in (\frac{1}{2}, 1 \rangle$. Nechť $h(x)$ je funkce: $h(x) \neq 0$ pro $x \in \langle 0, \frac{1}{2} \rangle$, $h(x) = 0$ pro $x \in (\frac{1}{2}, 1 \rangle$. Pak $g(x)$ i $h(x)$ jsou nenulové, ale $g(x) \cdot h(x)$ je nulová funkce na $\langle 0, 1 \rangle$.

Obor integrity, (komutativní) těleso

Definice

Okruh $\mathscr{R} = (R; +, \cdot)$ se nazývá **obor integrity**, je-li komutativní, unitární (s jednotkou $1 \neq 0$) a neobsahuje netriviální dělitele 0.

Příklad

Každý z okruhů $(\mathbb{Z}; +, \cdot)$, $(\mathbb{Q}; +, \cdot)$, $(\mathbb{R}; +, \cdot)$, $(\mathbb{C}; +, \cdot)$ je obor integrity.

Definice

Okruh $\mathscr{T} = (T; +, \cdot)$ s alespoň dvěma prvky se nazývá **těleso**, je-li $(T \setminus \{0\}; \cdot)$ grupa. Těleso $\mathscr{T}$ se nazývá **komutativní**, je-li $(T \setminus \{0\}; \cdot)$ abelovská grupa.

Příklad

Okruhy $(\mathbb{Q}; +, \cdot)$, $(\mathbb{R}; +, \cdot)$, $(\mathbb{C}; +, \cdot)$ jsou komutativní tělesa.
Okruh $(\mathbb{Z}; +, \cdot)$ není těleso.

Věta

Každé komutativní těleso je obor integrity.

Důkaz. Zřejmě stačí dokázat, že komutativní těleso $\mathscr{T} = (T; +, \cdot)$ má jednotku a neobsahuje netriviální dělitele 0. Avšak, je-li $\mathscr{T}$ těleso, je $(T \setminus \{0\}; \cdot)$ grupa s jednotkou 1, což je i jednotka okruhu $\mathscr{T}$. Dále, nechť $a, b \in T$, $a \neq 0 \neq b$. Pak $a, b \in T \setminus \{0\}$, tedy také $a \cdot b \in T \setminus \{0\}$, odkud $a \cdot b \neq 0$.

Příklad

$\mathscr{Z}_n = (\mathbb{Z}_n; \oplus, \odot)$, kde operace „$\oplus$“ a „$\odot$“ jsou sčítání a násobení modulo $n \in \mathbb{N}$, je komutativní, unitární okruh.

- Je-li n číslo složené (tedy $n = x \cdot y$, kde $x, y \in \mathbb{N}$ a $1 < x, y < n$), pak unitární komutativní okruh $\mathscr{Z}_n$ není oborem integrity (ani tělesem). Obsahuje totiž netriviální dělitele nuly, neboť $\overline{x} \odot \overline{y} = \overline{0}$.
- Je-li n prvočíslo, pak je $\mathscr{Z}_n$ komutativním tělesem (a tedy i oborem integrity).

Příklad

Vytvořte Cayleyho tabulky pro operace $\oplus$ a $\odot$ u okruhu

(a) $(\mathbb{Z}_3; \oplus, \odot)$

(b) $(\mathbb{Z}_6; \oplus, \odot)$,

kde $\oplus$ je operace sčítání modulo 3 (respektive modulo 6) a $\odot$ je operace násobení modulo 3 (respektive modulo 6) a kde $\mathbb{Z}_3 = \{\overline{0}, \overline{1}, \overline{2}\}$ je množina zbytkových tříd po celočíselném dělení třemi a $\mathbb{Z}_6 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{5}\}$ je množina zbytkových tříd po celočíselném dělení šesti. Určete, zda je daný okruh komutativní; unitární a zda je oborem integrity; tělesem.

Řešení (a):

$\oplus$	$\overline{0}$	$\overline{1}$	$\overline{2}$
$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{2}$
$\overline{1}$	$\overline{1}$	$\overline{2}$	$\overline{0}$
$\overline{2}$	$\overline{2}$	$\overline{0}$	$\overline{1}$

$\odot$	$\overline{0}$	$\overline{1}$	$\overline{2}$
$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$
$\overline{1}$	$\overline{0}$	$\overline{1}$	$\overline{2}$
$\overline{2}$	$\overline{0}$	$\overline{2}$	$\overline{1}$

Algebraická struktura $(\mathbb{Z}_3; \oplus, \otimes)$ je komutativním tělesem.

Dokončení příkladu

Řešení (b):

$\oplus$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{1}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{0}$
$\overline{2}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{0}$	$\overline{1}$
$\overline{3}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{0}$	$\overline{1}$	$\overline{2}$
$\overline{4}$	$\overline{4}$	$\overline{5}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
$\overline{5}$	$\overline{5}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$

$\odot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$
$\overline{1}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{2}$	$\overline{0}$	$\overline{2}$	$\overline{4}$	$\overline{0}$	$\overline{2}$	$\overline{4}$
$\overline{3}$	$\overline{0}$	$\overline{3}$	$\overline{0}$	$\overline{3}$	$\overline{0}$	$\overline{3}$
$\overline{4}$	$\overline{0}$	$\overline{4}$	$\overline{2}$	$\overline{0}$	$\overline{4}$	$\overline{2}$
$\overline{5}$	$\overline{0}$	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$

Zřejmě $(\mathbb{Z}_6; \oplus, \odot)$ je komutativním, unitárním okruhem. Nejedná se o obor integrity, ani o těleso, neboť u operace násobení existují případy, kdy je součin dvou nenulových prvků roven nulovému prvku, například: $\overline{2} \odot \overline{3} = \overline{0}$.

Definice

Je-li $\mathscr{R} = (R; +, \cdot)$ okruh, $A \subseteq R$ taková, že $\mathscr{A} = (A; +, \cdot)$ je opět okruh, pak $\mathscr{A}$ nazveme **podokruh okruhu** $\mathscr{R}$. Zápis: $\mathscr{A} \leq \mathscr{R}$.

Příklad

Je-li $\mathscr{R} = (R; +, \cdot)$ okruh s nulovým prvkem 0, pak vždy $(\{0\}; +, \cdot) \leq \mathscr{R}$ a $\mathscr{R} \leq \mathscr{R}$.

Definice

Je-li $\mathscr{T} = (T; +, \cdot)$ těleso, $A \subseteq T$ taková, že $(A; +, \cdot)$ je opět těleso, pak $(A; +, \cdot)$ nazveme **podtěleso tělesa** $\mathscr{T}$. Každé podtěleso tělesa $\mathscr{C} = (C; +, \cdot)$ komplexních čísel nazveme **číselné těleso**. Každý podokruh $\mathscr{C}$ nazveme **číselný okruh**.

Příklad

$\mathscr{C} = (\mathbb{C}; +, \cdot)$, $\mathscr{R} = (\mathbb{R}; +, \cdot)$, $\mathscr{Q} = (\mathbb{Q}; +, \cdot)$ jsou číselná tělesa, $\mathscr{Z} = (\mathbb{Z}; +, \cdot)$ je číselný okruh (obor integrity), který není tělesem.

Příklady

(a) Množina všech polynomů jedné proměnné nad libovolným číselným tělesem je vzhledem ke sčítání a násobení polynomů komutativním okruhem, v němž je jednotkovým prvkem konstantní polynom 1. Tento unitární okruh $\mathscr{T}[x] = (T[x]; +, \cdot)$ je oborem integrity, který není tělesem. Přitom množina všech konstantních polynomů (tedy polynomů stupně 0 a s nulovým polynomem) spolu s operacemi sčítání a násobení polynomů je podtělesem v $\mathscr{T}[x]$.

(b) Množina všech čtvercových matic stupně $n \geq 2$ nad některým číselným tělesem tvoří okruh vzhledem ke sčítání a násobení matic. Tento okruh není komutativní, ale je unitární, přičemž jednotkovým prvkem je jednotková matice stupně n.

Homomorfismus (a izomorfismus) okruhů

Definice

(a) Nechť $\mathscr{M}_1 = (M_1; +_1, \cdot_1)$ a $\mathscr{M}_2 = (M_2; +_2, \cdot_2)$ jsou okruhy a f je zobrazení z množiny M_1 do množiny M_2. Pak se f nazývá **homomorfismus okruhu $\mathscr{M}_1$ do okruhu $\mathscr{M}_2$**, platí-li $\forall a, b \in M_1$:

$$f(a +_1 b) = f(a) +_2 f(b), \qquad f(a \cdot_1 b) = f(a) \cdot_2 f(b),$$

tedy f je současně homomorfismem grupy $(M_1; +_1)$ do grupy $(M_2; +_2)$ a homomorfismem pologrupy $(M_1; \cdot_1)$ do pologrupy $(M_2; \cdot_2)$.

(b) Řekneme, že okruh $\mathscr{M}_2$ je **homomorfním obrazem** okruhu $\mathscr{M}_1$, existuje-li alespoň jeden surjektivní homomorfismus $\mathscr{M}_1$ na $\mathscr{M}_2$.

(c) Bijektivní homomorfismus okruhu $\mathscr{M}_1$ na $\mathscr{M}_2$ se nazývá **izomorfismus**. Příslušné okruhy $\mathscr{M}_1$ a $\mathscr{M}_2$ se nazývají **izomorfní**.

Poznámka. Podobně jako v případě grup platí, že identické zobrazení je izomorfismem okruhu $\mathscr{M}_1$ na okruh $\mathscr{M}_1$, že složení dvou izomorfismů okruhů je opět izomorfismem okruhů, a že je-li f izomorfismus okruhu $\mathscr{M}_1$ na okruh $\mathscr{M}_2$, pak inverzní zobrazení f^{-1} je izomorfismem okruhu $\mathscr{M}_2$ na okruh $\mathscr{M}_1$. Proto relace „být izomorfní s“ je ekvivalencí na třídě všech okruhů, která tuto třídu rozkládá na třídy navzájem izomorfních okruhů, které mají stejné algebraické vlastnosti. To znamená, že i zde můžeme používat formulaci, že „danou vlastnost mají, až na izomorfimus, právě jisté okruhy“.

Dvě vlastnosti konečných těles

Věta

Každé konečné komutativní těleso má počet prvků roven některé mocnině prvočísla.

Poznámka. Víme, že existuje alespoň jedna grupa řádu $n \in \mathbb{N}$. Podobně každé $n \in \mathbb{N}$ je počtem prvků některého okruhu. Podle předchozí věty však obdobná situace nenastane pro komutativní tělesa. Například neexistuje žádné komutativní těleso, které by mělo právě šest prvků.[31]

Věta

Má-li konečné komutativní těleso $\mathscr{T} = (T; +, \cdot)$ q prvků, pak pro každý prvek $a \in T$ platí $a^q = a$.

[31] Ve skutečnosti nemůže existovat vůbec žádné konečné těleso, jehož počet prvků by nebyl mocninou některého prvočísla. Podle Wedderburnovy věty je totiž každé konečné těleso komutativní. Příkladem nekomutativního tělesa je těleso kvaternionů.

Poznámka o využití konečných těles v teorii kódování

S využitím konečných těles jsou konstruovány třeba cyklické samoopravné kódy BCH.[32] Důležitou vlastností BCH kódů je možnost v průběhu návrhu kódu přesně kontrolovat počet opravitelných chyb ve výsledném kódu. Další výhodou je jejich snadné dekódování, které umožňuje zjednodušit návrh dekodérů s použitím malého výkonnostně slabého hardwaru. BCH kódy jsou používány v satelitní komunikaci, CD a DVD přehrávačích, pevných discích, flash discích a QR kódech.

Reedovy–Solomonovy (RS) kódy jsou nebinární cyklické samoopravné kódy, které mohou detekovat více náhodných chyb. Přidáním t kontrolních písmen k datům může RS kód detekovat libovolnou kombinaci až t chybných písmen či opravovat až $\frac{t}{2}$ písmen. V případě chybějících písmen dokáže doplnit až t chybějících písmen. Kód může také detekovat a opravovat kombinace chybných a chybějících písmen.

[32] Podrobněji na https://en.wikipedia.org/wiki/BCH_code.

Obsah

Číselná posloupnost (posloupnost reálných čísel)

Definice

Číselnou posloupností budeme rozumět každé zobrazení z $\mathbb{N}$ do $\mathbb{R}$. Zápis: $(a_n)_{n=1}^{+\infty}$ nebo jen (a_n). Reálné číslo a_n nazveme **n-tý člen posloupnosti**.

Poznámka. Číselnou posloupnost lze definovat i jako zobrazení z $\mathbb{N}$ do $\mathbb{C}$ (či do nějaké jiné číselné množiny).[33]
My ale zůstaneme u zobrazení z množiny přirozených čísel do množiny reálných čísel.

Poznámka. Číselná posloupnost tedy přiřazuje

$$1 \mapsto a_1, \quad 2 \mapsto a_2, \quad \ldots, \quad n \mapsto a_n, \quad n+1 \mapsto a_{n+1}, \quad \ldots$$

Poznámka. Pro stručnost budeme často místo „číselná posloupnost“ říkat jen „posloupnost“.

---

[33] Konečná posloupnost jakožto zobrazení z konečné podmnožiny $\mathbb{N}$ do $\mathbb{R}$ je vlastně uspořádanou n-ticí reálných čísel.

Způsoby zadání posloupnosti

Posloupnost bývá zadána:

- několika prvními členy (tak, aby bylo patrné pravidlo, jak vytvářet další členy); například $a_1 = 2$, $a_2 = 4$, $a_3 = 8$, $a_4 = 16$, $a_5 = 32$, $a_6 = 64$, ... vede na posloupnost (2^n)
- vzorcem pro n-tý člen; například pro $a_n = 3^n$ dostaneme triviálně posloupnost (3^n)
- rekurentně; například: $a_1 = 0$, $a_2 = 1$ a pravidlo $a_{n+2} = a_{n+1} + a_n$ (pro $n \in \mathbb{N}$) definují Fibonacciho posloupnost.

Poznámka. Rekurentní definice obsahuje zpravidla 1. člen (nebo několik prvních členů) a pravidlo, jak vytvořit další člen ze členů předcházejících. Rekurentní definice aritmetické posloupnosti: $a_1 = a$, $a_{n+1} = a_n + d$. Rekurentní definice geometrické posloupnosti: $a_1 = a \neq 0$, $a_{n+1} = a_n \cdot q$, kde $q \neq 0$.

Příklad

U následujících posloupností určete podle prvních šesti členů jejich n-tý člen:

(a) $a_1 = 6,\ a_2 = 16,\ a_3 = 26,\ a_4 = 36,\ a_5 = 46,\ a_6 = 56, \ldots$

(b) $a_1 = 1,\ a_2 = -1,\ a_3 = 1,\ a_4 = -1,\ a_5 = 1,\ a_6 = -1, \ldots$

(c) $a_1 = -4,\ a_2 = 7,\ a_3 = -10,\ a_4 = 13,\ a_5 = -16,\ a_6 = 19, \ldots.$

Řešení. Na přednášce.

Příklad

U následujících posloupností vypočítejte členy a_1, a_2, a_3, a_4:

(a) $\left((-1)^n \cdot 5n\right)_{n=1}^{+\infty}$

(b) $\left(\frac{n}{n+1}\right)_{n=1}^{+\infty}$

(c) $\left((1+\frac{1}{n})^n\right)_{n=1}^{+\infty}$.

Řešení. Na přednášce.

Operace s posloupnostmi

Poznámka. Posloupnost (a_n) je třeba odlišovat od množiny (všech) jejích členů. Například množina (všech) členů posloupnosti $(\frac{1}{n})$ je $\{1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{n}, \ldots\}$, množina (hodnot) členů posloupnosti $\big((-1)^n\big)$ je $\{-1, 1\}$.

Operace s posloupnostmi zavádíme takto:

- násobení reálným číslem c: $c \cdot (a_n) = (c \cdot a_n)$
- aritmetické operace (součet, rozdíl, součin, podíl):
  $(a_n) + (b_n) = (a_n + b_n)$,
  $(a_n) - (b_n) = (a_n - b_n)$,
  $(a_n) \cdot (b_n) = (a_n \cdot b_n)$,
  $(a_n)/(b_n) = (a_n/b_n)$ (pro $b_n \neq 0$)
- opačná posloupnost k (a_n) je posloupnost $(-a_n)$.

(Shora, zdola) omezená posloupnost

Definice

Posloupnost (a_n) se nazývá

- **shora omezená**, pokud existuje reálné číslo H tak, že $\forall n \in \mathbb{N}$ platí $a_n \leq H$.
- **zdola omezená**, pokud existuje reálné číslo L tak, že $\forall n \in \mathbb{N}$ platí $L \leq a_n$.
- **omezená**, je-li omezená shora i zdola.

Příklad

Posloupnost $(3n-2)$ je zdola omezená, není omezená shora, není omezená. Posloupnost $\left((-1)^n\right)$ je omezená shora i zdola, je omezená. Stacionární posloupnost (c), kde $c \in \mathbb{R}$ je omezená.

(Ryze) monotónní posloupnosti

Definice

Posloupnost (a_n) se nazývá

- **rostoucí**, pokud $\forall n \in \mathbb{N}$ platí $a_n < a_{n+1}$
- **klesající**, pokud $\forall n \in \mathbb{N}$ platí $a_n > a_{n+1}$
- **nerostoucí**, když $\forall n \in \mathbb{N}$ platí $a_n \geq a_{n+1}$
- **neklesající**, když $\forall n \in \mathbb{N}$ platí $a_n \leq a_{n+1}$.

Tyto čtyři druhy posloupností nazýváme **posloupnosti monotónní**, přičemž pro první dva druhy používáme název **posloupnosti ryze monotónní**.

Příklad

Posloupnost $(\frac{1}{n})$ je klesající (a nerostoucí). Posloupnost (n^3) je rostoucí (a neklesající). Každá stacionární (konstantní) posloupnost (c) je současně neklesající a nerostoucí, přičemž není ani rostoucí, ani klesající. Posloupnosti $\left((-1)^n\right)$, $\left((n-3)^2\right)$ a $(\sin n)$ monotónní nejsou.

Příklad

Dokažte, že posloupnost $(\frac{2n-3}{n+1})_{n=1}^{+\infty}$ je rostoucí a omezená.

Řešení. Víme, že posloupnost je rostoucí, pokud pro každé n přirozené platí: $a_n < a_{n+1}$. Ověřme tuto nerovnost:

$$\begin{aligned} \frac{2n-3}{n+1} &< \frac{2(n+1)-3}{(n+1)+1} \\ (2n-3)\cdot(n+2) &< (2n-1)\cdot(n+1) \\ 2n^2+n-6 &< 2n^2+n-1 \\ -6 &< -1. \end{aligned}$$

Tyto čtyři (na $\mathbb{N}$) navzájem ekvivalentní nerovnosti platí, což znamená, že posloupnost $(\frac{2n-3}{n+1})_{n=1}^{+\infty}$ je rostoucí.
Tato posloupnost je tedy zřejmě zdola omezená svým prvním členem, číslem $a_1 = \frac{-1}{2}$.
Na přednášce dále ukážeme, že je shora omezená číslem 2, tedy, že pro každé n přirozené platí: $a_n \leq 2$. (Tím důkaz dokončíme.)

Obsah

Rozšíření množiny reálných čísel o $+\infty$ a $-\infty$

Množinu reálných čísel rozšíříme o dva nové prvky: nevlastní číslo $+\infty$ a nevlastní číslo $-\infty$. Dostaneme tak množinu $\mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$. Na ní pro každé $x \in \mathbb{R}$ platí:

$-\infty < x < +\infty$, $-(-\infty) = +\infty$, $-(+\infty) = -\infty$,

$x \pm \infty = \pm\infty = \pm\infty + x$, $|+\infty| = |-\infty| = +\infty$,

$x - (-\infty) = (+\infty) + (+\infty) = (+\infty) - (-\infty) = +\infty$,

$x - (+\infty) = (-\infty) + (-\infty) = (-\infty) - (+\infty) = -\infty$,

$(+\infty) \cdot (+\infty) = (-\infty) \cdot (-\infty) = +\infty$,

$(-\infty) \cdot (+\infty) = (+\infty) \cdot (-\infty) = -\infty$.

Dále pro každé $x \in \mathbb{R}$, $x > 0$ platí:

$x \cdot (+\infty) = (+\infty) \cdot x = +\infty$, $x \cdot (-\infty) = (-\infty) \cdot x = -\infty$.

Podobně pro každé $x \in \mathbb{R}$, $x < 0$ platí:

$x \cdot (+\infty) = (+\infty) \cdot x = -\infty$, $x \cdot (-\infty) = (-\infty) \cdot x = +\infty$.

Nedefinujeme:

$(+\infty) - (+\infty)$, $(+\infty) + (-\infty)$, $(-\infty) + (+\infty)$, $(-\infty) - (-\infty)$,

$0 \cdot (+\infty)$, $(+\infty) \cdot 0$, $0 \cdot (-\infty)$, $(-\infty) \cdot 0$.

Další početní operace s nevlastními čísly

Na $\mathbb{R}^*$ jsme (rozšířením příslušných pravidel platných na $\mathbb{R}$) zavedli přirozené uspořádání a početní operace sčítání, odčítání a násobení. Nyní se zaměříme na operaci dělení a na mocnění.

Pro každé $x \in \mathbb{R}$ definujeme: $\frac{x}{+\infty} = \frac{x}{-\infty} = 0$.

Je-li $x > 0$, pak $\frac{+\infty}{x} = +\infty, \quad \frac{-\infty}{x} = -\infty$.

Pro $x < 0$ je $\frac{+\infty}{x} = -\infty, \quad \frac{-\infty}{x} = +\infty$.

Pro každé $n \in \mathbb{N}$ definujeme:
$(+\infty)^n = +\infty, \quad (\pm\infty)^{-n} = 0, \quad (-\infty)^n = (-1)^n \cdot (+\infty)$.

Pro $a, b \in \{-\infty, +\infty\}$ a pro $x \in \mathbb{R}^*$ **nedefinujeme:**
$\frac{a}{b}, \quad \frac{x}{0}, \quad a^0, \quad 1^b, \quad 0^0$.

Příklad

Určete, zda dané výrazy mají smysl, případně vypočítejte jejich hodnotu:

(a) $a = \frac{1000!}{-\infty} + (-\infty)^5 \cdot (1000^{1000} - \infty) + \sqrt{3} \cdot (+\infty) \cdot 10$

(b) $b = \frac{-\infty}{-73} \cdot (-\infty)^{12} + 3^{+\infty} \cdot (e - \pi)$

(c) $c = \sin(+\infty)$

(d) $d = \frac{100}{-\infty} + \frac{0}{+\infty} + \frac{333}{12^4} \cdot 0^7 + 3 \cdot (2^{-\infty} - 1) - \frac{5005}{2^{+\infty}}$

(e) $e = \left(\frac{3}{-\infty}\right)^{\left(\frac{-5}{\ln(+\infty)}\right)}$.

Řešení.

(a) $a = 0 + (-\infty) \cdot (-\infty) + \infty = +\infty$

(b) $b = +\infty + (-\infty)$ ... není definováno

(c) $c = \sin(+\infty)$ ... není definováno

(d) $d = 0 + 0 + 0 + 3 \cdot (-1) - 0 = -3$

(e) $e = 0^0$ ... není definováno.

Definice vlastní limity posloupnosti

Definice

Posloupnost (a_n) **má vlastní limitu** $a \in \mathbb{R}$, ((a_n) **je konvergentní**), pokud pro libovolné kladné reálné číslo ε existuje přirozené číslo n_0 tak, že pro všechna přirozená čísla n větší nebo rovna než n_0 platí: $|a_n - a| < \varepsilon$.

Používané zápisy:

$$\lim_{n \to +\infty} a_n = a; \qquad \lim a_n = a$$

$$(a_n)_{n=1}^{+\infty} \to a$$

$$a_n \to a \quad \text{pro} \quad n \to +\infty.$$

Více (k definici vlastní limity posloupnosti) na přednášce.

Definice nevlastní limity posloupnosti

Definice

- Posloupnost (a_n) **má nevlastní limitu a** $= +\infty$, pokud pro každé reálné číslo A existuje přirozené číslo n_0 tak, že pro všechna přirozená čísla n větší nebo rovna než n_0 platí: $a_n > A$.
- Posloupnost (a_n) **má nevlastní limitu a** $= -\infty$, pokud pro každé reálné číslo B existuje přirozené číslo n_0 tak, že pro všechna přirozená čísla n větší nebo rovna než n_0 platí: $a_n < B$.

V obou případech říkáme, že je posloupnost (a_n) **divergentní**. Divergentní je také každá **oscilující** posloupnost, tedy posloupnost, která nemá limitu vlastní ani nevlastní.

Poznámka. Pokud $lim_{n\to+\infty} = +\infty$, říkáme, že posloupnost (a_n) roste nade všechny meze. Podobně, v případě, že $lim_{n\to+\infty} = -\infty$, říkáme, že (a_n) roste pode všechny meze.

Každá posloupnost tedy konverguje nebo diverguje. Přitom ve druhém případě buď osciluje (tedy nemá limitu vlastní ani nevlastní) nebo diverguje k $+\infty$ nebo k $-\infty$.

Příklady

- Posloupnost $(\frac{2n-3}{n+1})$ je konvergentní a má (vlastní) limitu 2.
- Stacionární posloupnost (c) je konvergentní a má (vlastní) limitu c.
- Posloupnost $(3n^2)$ je divergentní, má (nevlastní) limitu $+\infty$.
- Posloupnost $(-0,001n)$ je divergentní a má (nevlastní) limitu $-\infty$.
- Posloupnost (q^n) je pro $q \leq -1$ divergentní, nemá limitu (osciluje). Pro $q = 1$ má (vlastní) limitu 1, pro $q > 1$ má (nevlastní) limitu $+\infty$ a pro $q \in (-1,1)$ má (vlastní) limitu 0.

Věty o limitách 1/2

Bez důkazů nyní uvedeme některá tvrzení o limitách posloupností.

Věta

Každá posloupnost má nejvýše jednu limitu.

Věta

Každá konvergentní posloupnost je omezená.

Věta

Každá neklesající shora omezená posloupnost je konvergentní.

Věta

Nechť $\lim a_n = a$, $\lim b_n = b$ a pro nekonečně mnoho n platí $a_n \leq b_n$. Pak $a \leq b$.

Věty o limitách 2/2

Věta

Nechť $\lim a_n = a$, $\lim b_n = b$. Pak platí, pokud výrazy na pravých stranách mají v $\mathbb{R}^*$ smysl:

1. $\lim(a_n \pm b_n) = a \pm b$
2. $\lim(a_n \cdot b_n) = a \cdot b$
3. pro $b_n \neq 0$, $b \neq 0$ je $\lim(a_n / b_n) = a/b$
4. $\lim |a_n| = |a|$.

Věta

Nechť $\lim a_n = a = \lim b_n$ a pro nekonečně mnoho n je $a_n \leq c_n \leq b_n$. Pak $\lim c_n = a$.

Nulové posloupnosti

Posloupnost (a_n), pro kterou $\lim_{n\to+\infty} a_n = 0$ nazveme **nulová posloupnost**. Tyto konvergentní posloupnosti se s výhodou využívají při výpočtu limit a (mimo jiné) umožňují definovat konvergenci podle následující věty.

Věta

$$\lim_{n\to+\infty} a_n = a \quad \Longleftrightarrow \quad \lim_{n\to+\infty} (a_n - a) = 0.$$

Věta

Jestliže $\lim_{n\to+\infty} |a_n| = +\infty$, pak $\lim_{n\to+\infty}(1/a_n) = 0$.

Věta

Je-li posloupnost (a_n) nulová a pro každé $n \in \mathbb{N}$ je

- $a_n > 0$, pak $\lim_{n\to+\infty} 1/a_n = +\infty$.
- $a_n < 0$, pak $\lim_{n\to+\infty} 1/a_n = -\infty$.
- $a_n \neq 0$, pak $\lim_{n\to+\infty} 1/|a_n| = +\infty$.

Příklad

Vypočítejte limity následujících posloupností:

(a)

$$A = \lim_{n\to+\infty} \frac{3n^2+5n+10}{-4n^3-9n+150}$$

(b)

$$B = \lim_{n\to+\infty} \frac{3\cdot 2^{n+1}+13}{5\cdot 2^n-7}.$$

Řešení.

(a)

$$A = \lim_{n\to+\infty} \frac{\frac{3n^2}{n^3}+\frac{5n}{n^3}+\frac{10}{n^3}}{\frac{-4n^3}{n^3}+\frac{-9n}{n^3}+\frac{150}{n^3}} = \frac{0+0+0}{-4+0+0} = 0$$

(b)

$$B = \lim_{n\to+\infty} \frac{\frac{3\cdot 2^{n+1}+13}{2^{n+1}}}{\frac{5\cdot 2^n-7}{2^{n+1}}} = \frac{3+0}{(5\cdot\frac{1}{2})-0} = \frac{6}{5}.$$

Příklad

Vypočítejte limity následujících posloupností:

(a)

$$A = \lim_{n\to+\infty} \cos\left(\frac{n+3}{\pi}\right)$$

(b)

$$B = \lim_{n\to+\infty} n\cdot\left(\sqrt{7+n^2}-n\right).$$

Řešení.

(a) Daná limita neexistuje.

(b)

$$\begin{aligned} B &= \lim_{n\to+\infty} \frac{n\cdot\left(\sqrt{7+n^2}-n\right)\cdot\left(\sqrt{7+n^2}+n\right)}{\sqrt{7+n^2}+n} \\ &= \lim_{n\to+\infty} n\cdot\frac{(7+n^2)-n^2}{\sqrt{7+n^2}+n} = \lim_{n\to+\infty} \frac{7}{\sqrt{\frac{7}{n^2}+1}+1} = \frac{7}{2}. \end{aligned}$$

Příklad

Najděte dvě nulové posloupnosti $(a_n)_{n=1}^{+\infty}$, $(b_n)_{n=1}^{+\infty}$ tak, aby

(a)

$$\lim_{n\to+\infty} \frac{a_n}{b_n} = \frac{5}{3}$$

(b)

$$\lim_{n\to+\infty} \frac{a_n}{b_n} = -\infty.$$

Řešení. Na přednášce.

Příklad*

Ukažte, že každé iracionální číslo je limitou neklesající posloupnosti racionálních čísel. Najděte takovou posloupnost pro číslo π.

Řešení. Na přednášce.

Některé významné limity

Věta

Nechť $n \in \mathbb{N}$, $0 < k \in \mathbb{R}$, $1 < a \in \mathbb{R}$, pak

- $$\lim_{n\to+\infty} \sqrt[n]{k} = 1, \qquad \lim_{n\to+\infty} \sqrt[n]{n} = 1$$
- $$\lim_{n\to+\infty} \frac{n}{\log_a n} = +\infty, \qquad \lim_{n\to+\infty} \frac{\log_a n}{n} = 0$$
- $$\lim_{n\to+\infty} \frac{a^n}{n^k} = +\infty, \qquad \lim_{n\to+\infty} \frac{n^k}{a^n} = 0$$
- $$\lim_{n\to+\infty} \frac{n!}{k^n} = +\infty, \qquad \lim_{n\to+\infty} \frac{k^n}{n!} = 0$$
- $$\lim_{n\to+\infty} \left(1+\frac{1}{n}\right)^n = e \doteq 2,718281828459045235 36.$$

Poznámka. Předchozí věta je v souladu s dříve uvedeným porovnáním rychlosti růstu u vybraných číselných funkcí.

Věta

Nechť $f, g : \mathbb{N} \to \mathbb{R}_0^+$ a $\lim_{n\to+\infty} \frac{f(n)}{g(n)} = a$. Pak

1. je-li $a = 0$ je $f(n) \in O(g(n))$.
2. je-li $a \in (0, +\infty)$ je $f(n) \in \Theta(g(n))$.

Princip důkazu. Podle definice limity pro $\varepsilon \in \mathbb{R}$, $n_0, n \in \mathbb{N}$:

$$\forall \varepsilon > 0 \ \exists n_0 \ \forall n \geq n_0 : \ \left| \frac{f(n)}{g(n)} - a \right| < \varepsilon.$$

Upravíme poslední nerovnost (odstraníme absolutní hodnotu, vynásobíme $g(n)$) a obdržíme:

$$(a - \varepsilon) \cdot g(n) < f(n) < (a + \varepsilon) \cdot g(n),$$

odkud dostaneme požadované.

Příklad

Nechť $f, g : \mathbb{N} \to \mathbb{R}^+$, přičemž

$$f(n) = 3n^3 + 5n^2 + 4n + 7, \quad g(n) = 2n^3 + n^2 + n.$$

Pak $f(n) \in \Theta(g(n))$ a tedy i $g(n) \in \Theta(f(n))$ (funkce f a g jsou asymptoticky ekvivalentní). Také $f(n), g(n) \in \Theta(n^3)$ a $f(n), g(n) \in O(n^4)$.

Výpočet a komentář na přednášce.

Obsah

V praxi je mnoho situací, kdy známe několik prvních členů a_1, a_2, a_3, ..., a_n nějaké posloupnosti a pomocí této znalosti chceme zjistit, zkonstruovat nebo předpovědět její další člen a_{n+1}. Může jít například o posloupnost peněžních částek, (časovou) posloupnost údajů o objemu výroby, a podobně. Problémem je, jak určit další člen (nebo alespoň jeho přibližnou hodnotu) ze znalosti předchozích. Zvláštní pozornost si zaslouží posloupnost aritmetická a posloupnost geometrická, které se vyskytují poměrně často.

Aritmetická posloupnost

Definice

Aritmetická posloupnost[a] je posloupnost, která je dána svým prvním členem a_1, konstantní diferencí d a rekurentním pravidlem $\forall n \in \mathbb{N}$: $a_{n+1} = a_n + d$.

[a] Aritmetickou posloupnost lze rovněž definovat jako posloupnost, u níž rozdíl libovolných dvou po sobě jdoucích členů je konstantní.

Z rekurentního pravidla dostaneme vzorec pro n-tý člen[34]:

$$a_n = a_1 + (n-1)d.$$

Snadno nahlédneme, že aritmetická posloupnost je pro $d > 0$ rostoucí a pro $d < 0$ klesající.

Příklady

- Aritmetická posloupnost, jejíž první čtyři členy jsou: $a_1 = 7$, $a_2 = 4$, $a_3 = 1$, $a_4 = -2$, má diferencí $d = -3$.
- Posloupnost $(2n-1)$ je aritmetická s diferencí $d = 2$.

[34] Dokazuje se jednoduše například matematickou indukcí.

Součet prvních n členů aritmetické posloupnosti

Praktický význam má součet s_n prvních n členů aritmetické posloupnosti. Vzorec pro s_n lze odvodit například tak, že si ho vyjádříme dvěma způsoby:

$$\begin{aligned} s_n &= a_1 + (a_1 + d) + (a_1 + 2d) + \cdots + (a_1 + (n-1)d), \\ s_n &= a_n + (a_n - d) + (a_n - 2d) + \cdots + (a_n - (n-1)d). \end{aligned}$$

Po sečtení máme $2s_n = n \cdot (a_1 + a_n)$, takže $s_n = \frac{n}{2} \cdot (a_1 + a_n)$.

Příklad

Ve skladu jsou na sobě naskládány krabice tak, že v každé řadě je o jednu krabici méně než v řadě pod ní. Vypočítejte, kolik je ve skladu celkem krabic, pokud spodní řada obsahuje 26 krabic a vrchní řada 8 krabic.

Řešení. Víme, že $a_1 = 26$, $n = (26 - 8) + 1 = 19$, $a_{19} = 8$. Dosazením těchto hodnot do obecného vzorce $s_n = \frac{n}{2} \cdot (a_1 + a_n)$ získáme hodnotu $s_{19} = \frac{19}{2} \cdot (26 + 8) = 323$. Ve skladu je 323 krabic.

Geometrická posloupnost

Definice

Geometrická posloupnost[a] je posloupnost, která je dána svým 1. členem $a_1 \neq 0$, konstantním kvocientem $q \neq 0$ a rekurentním pravidlem $\forall n \in \mathbb{N}$: $a_{n+1} = a_n \cdot q$.

[a] Geometrickou posloupnost lze rovněž definovat jako posloupnost, u níž podíl libovolných dvou po sobě jdoucích členů je konstantní.

Z rekurentního pravidla dostaneme vzorec pro n-tý člen[35]:

$$a_n = a_1 \cdot q^{n-1}.$$

Příklady

- Geometrická posloupnost mající prvních pět členů: $a_1 = \frac{1}{3}$, $a_2 = 1$, $a_3 = 3$, $a_4 = 9$, $a_5 = 27$ má kvocient $q = 3$.
- Posloupnost $\left(\left(\frac{-1}{2}\right)^n\right)$ je geometrická s kvocientem $q = -\frac{1}{2}$.
- Posloupnosti (n), $(2n^2 - n + 1)$, $(n!)$ geometrické nejsou.

[35] Dokazuje se jednoduše například matematickou indukcí.

Příklad

Na spořicím účtu je uloženo 200 000 Kč. Vypočítejte, jaká částka bude na účtu přesně po pěti letech, připisuje-li se

(a) čtvrtletně na konto 0,8% úrok.

(b) ročně na konto 3,2% úrok.

(c) ročně na konto 3,6% úrok.

Řešení.
Využijeme vzorec pro výpočet n-tého členu geometrické posloupnosti: $a_n = a_1 \cdot q^{n-1}$.

(a) Víme, že $a_1 = 200\,000$; $q = 1,008$; $n = 5 \cdot 4 = 20$. Po dosazení máme: $a_{20} = 200\,000 \cdot 1,008^{19} \doteq 232\,691,28$. Částka se tedy po pěti letech zvýší na 232 691,28 Kč (po zaokrouhlení na dvě desetinná místa).

(b) a (c) Analogicky, zkuste sami.

Součet prvních n členů geometrické posloupnosti

Praktický význam má součet prvních n členů geometrické posloupnosti (tzv. n-tý částečný součet geometrické řady). Vzorec pro s_n lze odvodit tak, že vyjádříme s_n a $q \cdot s_n$:

$$\begin{aligned} s_n &= a_1 + a_1 \cdot q + a_1 \cdot q^2 + \cdots + a_1 \cdot q^{n-1}, \\ q \cdot s_n &= a_1 \cdot q + a_1 \cdot q^2 + \cdots + a_1 \cdot q^{n-1} + a_1 \cdot q^n. \end{aligned}$$

Po odečtení je $s_n \cdot (1 - q) = a_1 \cdot (1 - q^n)$, takže (pro $q \neq 1$):

$$s_n = a_1 \cdot \frac{1 - q^n}{1 - q} = a_1 \cdot \frac{q^n - 1}{q - 1}.$$

Příklad

Vynálezce šachové hry požadoval podle pověsti odměnu za každé ze 64 polí šachovnice takto: za 1. pole jedno obilné zrno, za 2. pole 2 zrna, za 3. pole 4 zrna, atd., za každé další vždy dvojnásobek. Kolik zrnek obilí měl dostat?

Řešení. Na přednášce.

Obsah

Číselná řada (konvergentní a divergentní)

Definice

Nechť $(a_n)_{n=1}^{+\infty}$ je číselná posloupnost (posloupnost reálných čísel). **Číselnou řadou** (řadou reálných čísel) budeme rozumět symbol

$$\sum_{n=1}^{+\infty} a_n \quad \text{nebo} \quad a_1 + a_2 + a_3 + \cdots + a_n + \ldots$$

Posloupnost $(s_n)_{n=1}^{+\infty}$ definovanou vztahy[a]

$$s_1 = a_1\,,\ s_2 = a_1 + a_2,\ \ldots,\ s_n = a_1 + \cdots + a_n,\ \ldots$$

nazýváme **posloupností částečných součtů** řady $\sum_{n=1}^{\infty} a_n$. Jestliže existuje vlastní limita $\lim_{n\to+\infty} s_n = s$, říkáme, že **řada** $\sum_{n=1}^{+\infty} a_n$ **konverguje** a **má součet** s (pak píšeme $\sum_{n=1}^{+\infty} a_n = s$). V opačném případě říkáme, že **řada diverguje**.

---

[a] Číslo s_n nazýváme **n-tým částečným součtem** řady $\sum_{n=1}^{+\infty} a_n$.

Podle definice řada $\sum a_n$ diverguje, jestliže neexistuje vlastní limita posloupnosti částečných součtů. Tedy v případě, že $\lim_{n\to+\infty} s_n$ je nevlastní (pak ji též nazýváme součet řady) nebo neexistuje (pak **řada nemá součet**).

Poznámka. Číselnou řadu $\sum_{n=1}^{+\infty} a_n$ lze považovat za zobecnění součtu konečného počtu reálných čísel. Pokud použijeme zkrácený zápis $\sum a_n$ (tedy vynecháme-li podmínku pro n), uvažujeme členy vždy od nejmenšího $n \in \mathbb{N}$, pro něž má výraz a_n smysl. U každé řady vyvstávají dva základní problémy: zda konverguje, a když konverguje, tak jaký má řada součet.

Příklad

Stanovte součet řady $\sum_{n=1}^{+\infty} \frac{1}{n\cdot(n+1)}$.

Řešení. Na přednášce s tím, že $a_n = \frac{1}{n\cdot(n+1)} = \frac{1}{n} - \frac{1}{n+1}$.

Geometrická řada $\sum_{n=1}^{+\infty} aq^{n-1}$

Příkladem řady, u níž lze snadno rozhodnout o konvergenci a určit její součet s, je geometrická řada ($a \neq 0 \neq q$):

$$a + aq + aq^2 + \cdots + aq^n + \ldots$$

Víme, že její n-tý částečný součet je (pro $q \neq 1$):

$$s_n = a \cdot \frac{1 - q^n}{1 - q}.$$

Geometrická řada tedy

- pro $|q| < 1$ konverguje a její součet $s = \frac{a}{1-q}$
- pro $q \geq 1$ diverguje, $s = +\infty$ pro $a > 0$, $s = -\infty$ pro $a < 0$
- pro $q \leq -1$ neexistuje $\lim s_n$, řada diverguje a součet nemá.

Zadání příkladu: Achilles a želva

Vysvětlete, v čem je mylná starověká Zenónova aporie, podle které rychlonohý běžec Achilles nikdy nedohoní želvu, která je o kus před ním. Pro konkrétnost si představme, že má závod délku 140 metrů. Jelikož je Achilles dvacetkrát rychlejší než želva, dá jí na začátku náskok 100 metrů. V době, kdy Achilles uběhne 100 metrů, bude želva od startu vzdálena 105 metrů. Když Achilles uběhne dalších 5 metrů, bude želva zase o kousek dál a tak až do nekonečna. Protože se želva vždy trochu posune, Achilles ji (podle Zenóna) nikdy nedohoní.

Řešení příkladu: Achilles a želva

Pointa tkví v tom, že součet (nekonečné) řady může být konečný. Snadno můžeme určit místo, kde Achilles želvu dožene. Stačí sečíst nekonečně mnoho úseků, o které se želva posune směrem od startu k cíli:

$$s = 100 + \frac{100}{20} + \frac{100}{20^2} + \frac{100}{20^3} + \dots$$

Jedná se o geometrickou řadu s prvním členem $a = 100$ a s kvocientem $q = \frac{1}{20}$. Jelikož je $|q| < 1$, daná řada konverguje. Po dosazení do vzorce dostaneme:

$$s = \frac{a}{1-q} = \frac{100}{1-\frac{1}{20}} = \frac{2000}{19} \doteq 105,2631578947.$$

Achilles želvu dožene přibližně ve vzdálenosti 105 metrů a 26,3 centimetrů od startu.

Základní harmonická řada diverguje

Základní harmonická řada $\sum_{n=1}^{+\infty} \frac{1}{n}$ je důležitý příklad číselné řady, kde

$$s_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{n}.$$

Dá se dokázat, že je divergentní. To je celkem neintuitivní výsledek, neboť například: $s_{1000} \doteq 7,48$ a $s_{1000000} \doteq 14,39$.

Příklad

Dokažte divergenci řady $\sum_{n=1}^{+\infty} \frac{1}{\sqrt{n}}$.

Nástin řešení.

$$s_n = 1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \cdots + \frac{1}{\sqrt{n}} > n \cdot \frac{1}{\sqrt{n}} = \sqrt{n} \to +\infty,$$

tedy daná řada je divergentní.

Nutná podmínka konvergence řad

Nutná podmínka konvergence

Jestliže řada $\sum_{n=1}^{+\infty} a_n$ konverguje, pak $\lim_{n\to+\infty} a_n = 0$.

Důkaz. Nechť $\sum_{n=1}^{+\infty} a_n$ konverguje, pak $\lim_{n\to+\infty} s_n = s \in \mathbb{R}$. Pak

$$\lim_{n\to+\infty} a_n = \lim_{n\to+\infty} (s_n - s_{n-1}) = \lim_{n\to+\infty} s_n - \lim_{n\to+\infty} s_{n-1} = s - s = 0.$$

Poznámka. Uvedená podmínka konvergence není postačující, neboť například základní harmonická řada tuto podmínku splňuje ($\lim_{n\to+\infty} \frac{1}{n} = 0$), i když je divergentní ($\sum_{n=1}^{+\infty} \frac{1}{n} = +\infty$).

Příklad

Následující tři číselné řady:

$$\sum_{n=1}^{+\infty} \sqrt{n}, \qquad \sum_{n=1}^{+\infty} (-1)^n \cdot n, \qquad \sum_{n=1}^{+\infty} \frac{n!}{n^{10}}$$

divergují, protože nesplňují nutnou podmínku konvergence.

Stejné chování řad

Některé formulace vlastností řad se zjednoduší, jestliže zavedeme pojem chování řady.

Definice

Říkáme, že dvě řady mají **stejné chování**, právě když jsou obě konvergentní, nebo obě mají nevlastní součet nebo obě nemají součet.

Věta o vynechání prvních k členů

Chování řady se nezmění, vynecháme-li jejích prvních k členů.

K důkazu. V původní řadě je $s_n = a_1 + a_2 + \cdots + a_n$, v upravené řadě je částečný součet $\sigma_m = a_{k+1} + a_{k+2} + \cdots + a_{k+m}$. Pro $n > k$ položme $n = k + m$. Pak $s_n = s_k + \sigma_m$ a tedy částečné součty s_n, σ_m se navzájem liší jen o reálnou konstantu s_k. Odsud plyne tvrzení pro všechny tři druhy chování.

Řady s nezápornými členy

Řady $\sum a_n$ s nezápornými členy, $a_n \geq 0$, mají některé význačné vlastnosti pokud jde o konvergenci a její zjišťování. Jsou založeny zejména na tom, že posloupnost s_n jejich částečných součtů je neklesající, takže má vždy limitu. Pokud je posloupnost s_n shora omezená, je řada $\sum a_n$ konvergentní, není-li s_n shora omezená, má řada $\sum a_n$ součet $+\infty$.

Nyní pojednáme o limitních kritériích konvergence a divergence řad s nezápornými členy. V případě, že má daná číselná řada všechny své členy kladné, budeme mluvit o **kladné řadě**.

Limitní srovnávací kritérium

Tvrzení

Máme-li řady $\sum a_n$ a $\sum b_n$, kde $0 \leq a_n \leq b_n$ pro každé $n \in \mathbb{N}$. Pak

- z konvergence majorantní řady $\sum b_n$ plyne konvergence řady $\sum a_n$.
- z divergence minorantní řady $\sum a_n$ plyne divergence řady $\sum b_n$.

Toto tvrzení umožňuje dokázat následující větu.

Věta (limitní srovnávací kritérium)

Mějme dvě kladné řady $\sum a_n$, $\sum b_n$, a nechť existuje $\lim_{n\to+\infty} \frac{a_n}{b_n} = K$. Pak pro $K \in (0, +\infty)$ mají obě řady stejné chování.

Příklad

Rozhodněte o konvergenci řady

$$\sum_{n=1}^{+\infty} \frac{1}{3n+5}.$$

Řešení. Danou řadu srovnáme se základní harmonickou řadou, o které víme, že je divergentní. Máme:

$$\lim_{n\to+\infty} \frac{\frac{1}{3n+5}}{\frac{1}{n}} = \lim_{n\to+\infty} \frac{n}{3n+5} = \frac{1}{3}.$$

Zjistili jsme, že podle limitního srovnávacího kritéria zadaná řada diverguje.

Limitní podílové kritérium

Věta (limitní podílové kritérium)

Nechť $\sum a_n$ je kladná řada a existuje

$$\lim_{n\to+\infty} \frac{a_{n+1}}{a_n} = A.$$

Pak pro $A < 1$ daná řada konverguje a pro $A > 1$ řada diverguje.

Příklad

Rozhodněte o konvergenci řady $\sum_{n=1}^{+\infty} \frac{3^n}{n!}$.

Řešení. Užitím limitního podílového kritéria:

$$\lim_{n\to+\infty} \frac{\frac{3^{n+1}}{(n+1)!}}{\frac{3^n}{n!}} = \lim_{n\to+\infty} \frac{3}{n+1} = 0 < 1,$$

vidíme, že je zadaná řada konvergentní.

Limitní odmocninové kritérium

Věta (limitní odmocninové kritérium)

Nechť $\sum a_n$ je řada s nezápornými členy a existuje $\lim \sqrt[n]{a_n} = A$. Pak pro $A < 1$ daná řada konverguje a pro $A > 1$ řada diverguje.

Příklad

Rozhodněte o konvergenci řady

$$\sum_{n=1}^{+\infty} \frac{n \cdot 2^n}{3^n}.$$

Řešení. Podle limitního odmocninového kritéria:

$$\lim_{n \to +\infty} \sqrt[n]{\frac{n \cdot 2^n}{3^n}} = \frac{2}{3} < 1,$$

je zadaná řada konvergentní.

Poznámka. Řady se často využívají při určování přibližných hodnot výrazů. Například z toho, že

$$e^x = \sum_{n=0}^{+\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots,$$

můžeme relativně snadno určit třeba hodnotu $\frac{1}{\sqrt[3]{e}}$ s přesností na deset desetinných míst. Dodejme, že se řady uplatňují také při určování přibližných hodnot určitých integrálů.

Obsah

Stupeň vrcholu

Definice

Mějme dán neorientovaný graf $G = \langle V, E \rangle$. **Stupeň vrcholu** $u \in V$ je počet hran, pro které je vrchol u koncovým vrcholem. Označení: $\deg(u)$.

Poznámka. U orientovaných grafů se zavádí vstupní stupeň vrcholu (počet hran, které do vrcholu přicházejí) a výstupní stupeň vrcholu (počet hran, které z vrcholu vycházejí). Stupněm vrcholu je pak součet obou hodnot. My se v této části zaměříme výhradně na grafy neorientovné.

Princip sudosti a jeho důsledek

Věta

V každém neorientovaném grafu $G = \langle V, E \rangle$ platí, že

$$\sum_{u \in V} \deg(u) = 2|E|.$$

K důkazu. Při sčítání stupňů vrcholů v grafu G započítáme každou hranu dvakrát (jednou za každý její konec). Z toho ihned plyne dokazované tvrzení.

Důsledek

V každém neorientovaném grafu je počet vrcholů majících lichý stupeň sudé číslo.

Důkaz. Na přednášce.

Skóre grafu

Definice

Mějme dán neorientovaný graf $G = \langle V, E \rangle$, kde $V = \{v_1, v_2, \ldots, v_n\}$. Pak posloupnost stupňů jeho vrcholů: $\deg(v_1), \deg(v_2), \ldots, \deg(v_n)$ nazveme **skóre grafu**.[a]

[a] Dvě skóre považujeme za stejná, pokud jedno dostaneme permutací druhého, na zvoleném pořadí vrcholů tedy nezáleží.

Poznámka. Dva izomorfní grafy mají stejné skóre, odkud obměnou implikace vyplývá, že dva grafy s různým skóre jsou neizomorfní. Opačná implikace však neplatí, neboť mají-li dva grafy stejné skóre, nemusí být izomorfní. Tento fakt lze snadno demonstrovat třeba na dvou grafech majících skóre 2,2,2,2,2,2 (podrobněji na přednášce).

Věta o skóre grafu

Věta Havlova–Hakimiho

Mějme dána nezáporná celá čísla $d_1 \geq d_2 \geq d_3 \geq \cdots \geq d_n$, kde $1 \leq d_1 \leq n-1$. Pak je posloupnost

$$d_1, d_2, d_3, \ldots, d_n$$

skóre nějakého grafu s n vrcholy, právě když je posloupnost

$$d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n$$

skóre nějakého grafu s $n-1$ vrcholy.

Princip důkazu na přednášce.

Test skóre grafu

Algoritmus testující skóre grafu

Vstup: nezáporná celá čísla $d_1 \geq d_2 \geq \cdots \geq d_n$, kde $n \in \mathbb{N}$.
Výstup: odpověď ANO, je-li posloupnost $d_1, d_2, \ldots, d_n$ skóre grafu; jinak odpověď NE.

1. Je-li $d_1 = 0$, odpověz ANO a skonči.
2. V případě, že $d_1 > n-1$, odpověz NE a skonči.
3. Z posloupnosti $d_1, d_2, d_3, \ldots, d_n$ urči novou posloupnost $d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n$. Vyskytuje-li se v nové posloupnosti záporné číslo, odpověz NE a skonči. Je-li v nové posloupnosti nejvyšším číslem 0, odpověz ANO a skonči.
4. Novou posloupnost uspořádej sestupně. Pak z ní odstraň členy s nulovou hodnotou a pokračuj (s ní) bodem 2.

Ukázka použití na přednášce a na cvičení.

Eulerovský tah

Definice

Nechť $G = \langle V, E \rangle$ je neorientovaný graf. Pak **eulerovský tah** je tah, který obsahuje všechny vrcholy grafu G a ve kterém se každá hrana vyskytuje právě jednou. Je-li tento tah uzavřený nazývá se **uzavřený eulerovský tah**.

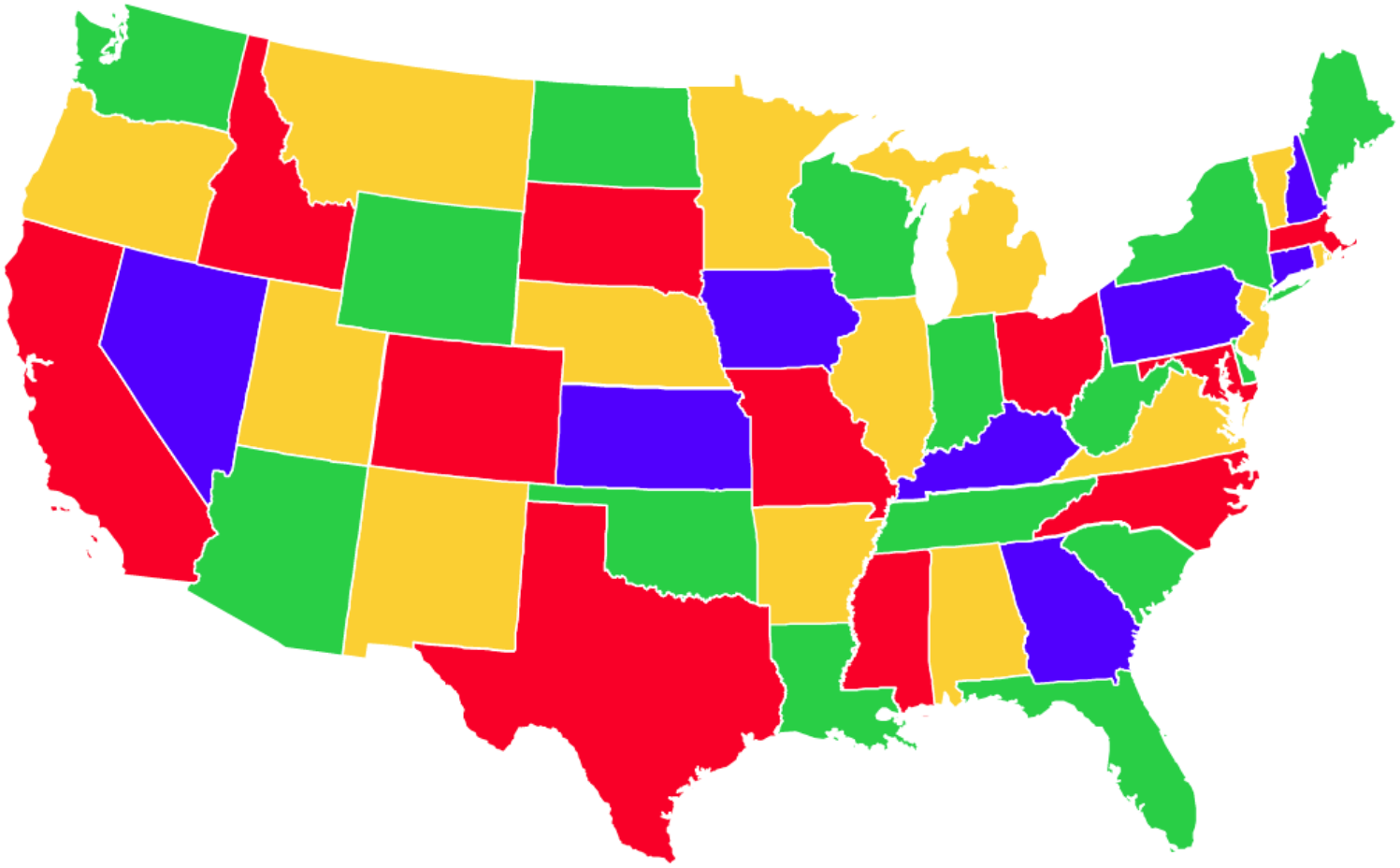
Věta

- V neorientovaném grafu $G = \langle V, E \rangle$ existuje uzavřený eulerovský tah, právě když je graf G souvislý a každý jeho vrchol má sudý stupeň.
- V neorientovaném grafu $G = \langle V, E \rangle$ existuje neuzavřený eulerovský tah, právě když je graf G souvislý a má právě dva vrcholy lichého stupně.

Princip důkazu na přednášce.
Ukázka použití věty na přednášce a na cvičení.

Obsah

Problém čtyř barev 1/4



Zdroj: 3.bp.blogspot.com/-PtOxbW9Ngns/Uwugi7fBg6I/AAAAAAAABwE/mA89sdschq0/s1600/Four+Colors.png

Problém čtyř barev

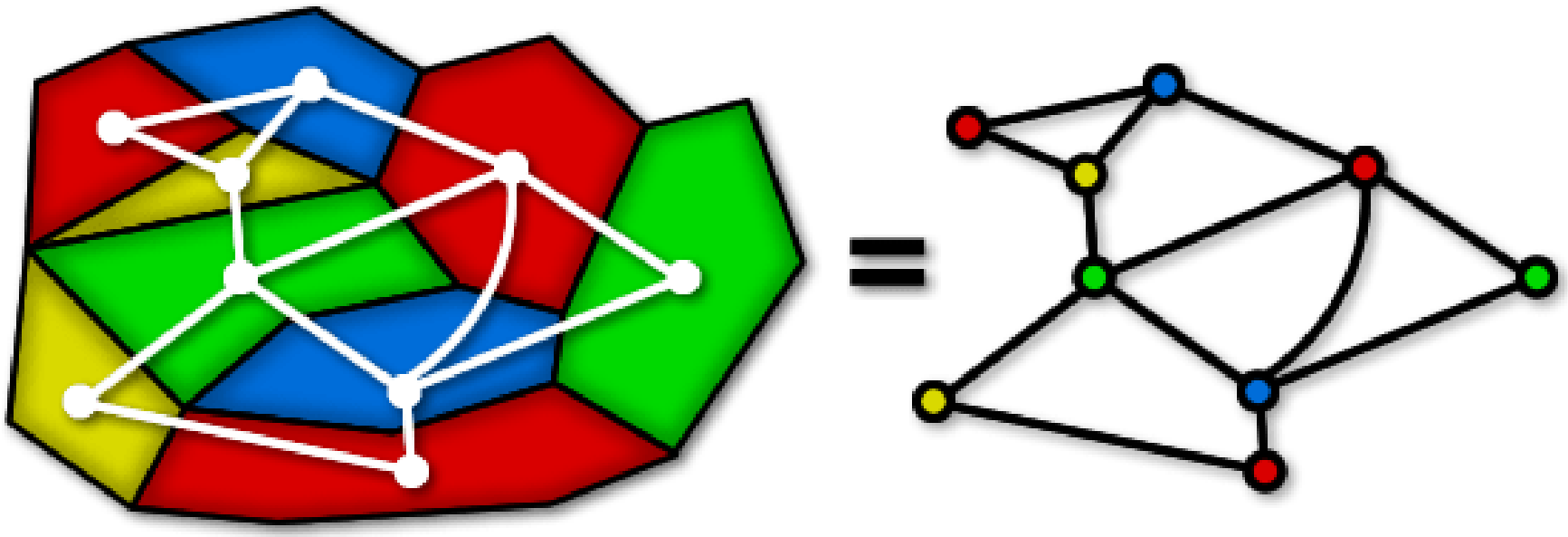
Kolik nejméně (různých) barev je potřeba k obarvení libovolné politické mapy tak, aby žádné dva sousední[a] státy nebyly obarveny stejnou barvou?

[a] Státy uvažujeme jako souvislé oblasti s tím, že za sousední státy bereme takové, jenž mají společnou hranici (nestýkají se jen v jednotlivých izolovaných bodech).

Poznámka. To, že stačí pět barev lze dokázat relativně snadno. Dnes víme, že čtyři barvy stačí pro všechny možné případy (konfigurace). Jak je uvedeno dále, důkaz tohoto tvrzení (věty o čtyřech barvách) je velmi obtížný.

Problém čtyř barev 3/4

- Poprvé o problému čtyř barev pojednal v roce 1840 (na své přednášce) A. F. Möbius.[36]
- Problém čtyř barev dlouhou dobu významně podněcoval rozvoj teorie grafů.



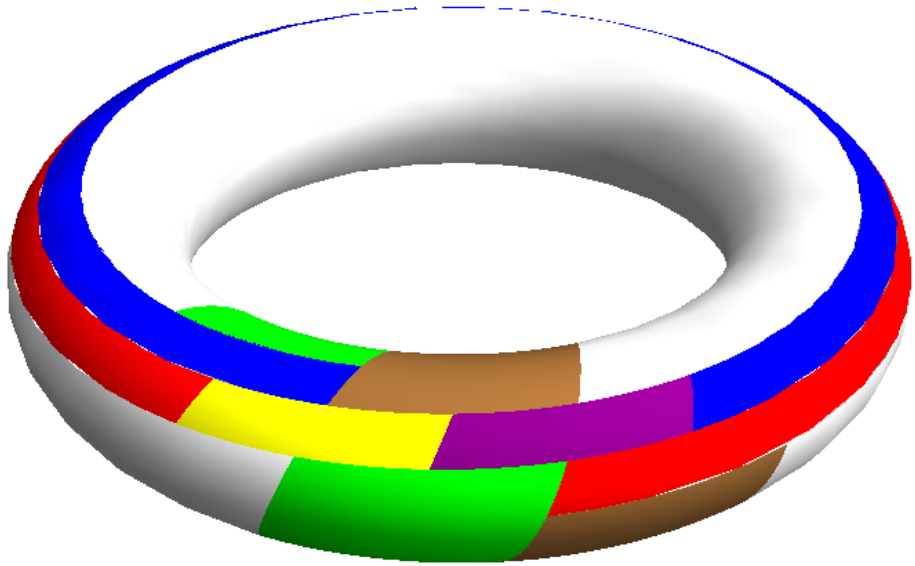
Zdroj: world.mathigon.org/resources/Graph_Theory/fourcolour.png

[36] Německý matematik August Ferdinand Möbius (1790 – 1868) položil základy topologie. Je po něm pojmenována Möbiova páska – trojrozměrný útvar mající pouze jednu stranu.

Problém čtyř barev 4/4

- V roce 1976 Wolfgang Haken ve spolupráci s Kennethem Appelem našli pomocí počítače kompletní důkaz věty o čtyřech barvách. Uměli se vypořádat se všemi (tehdy 1936) konfiguracemi. Řešení si vyžádalo cca 1200 hodin strojového času, přičemž jejich test odstranitelnosti používal 487 pravidel. Zhruba čtyři roky trvala příprava metod, programu a samotná práce s počítačem.
- Jejich pokračovatelé vylepšili test odstranitelnosti a zjistili, že stačí ověřovat podstatně méně konfigurací. Zatím však nikdo nedokázal snížit počet konfigurací natolik, aby byl důkaz ověřitelný člověkem. Problém čtyř barev se stal první velkou větou dokázanou s využitím počítače, bez možnosti tradičního přímého ověření od recenzentů (z oblasti matematiky).

Poznámka. Z následujícího obrázku je patrné, že k obarvení „mapy“ na anuloidu je potřeba alespoň sedmi různých barev.



Zdroj: i.stack.imgur.com/6Ay5o.png

Vrcholové barvení grafu, chromatické číslo grafu

Barvení grafu je jednou z disciplín teorie grafů, která se zabývá přiřazováním barev (typicky reprezentovaných přirozenými čísly) různým vrcholům[37] v daném grafu.

Definice

Nechť $k \in \mathbb{N}$ a $G = \langle V, E \rangle$ je neorientovaný graf. Zobrazení $f : V \to \{1, 2, \ldots, k\}$ nazveme obarvením grafu G pomocí k barev, pokud pro každou hranu $\{u, v\} \in E$ je $f(u) \neq f(v)$. Říkáme, že graf G je **k-chromatický**. Nejmenší číslo k, pro které je graf G k-chromatický se nazývá **chromatické číslo** grafu. Značí se $\chi(G)$.[a]

[a] $\chi(G)$ je tedy minimální počet barev potřebný pro obarvení grafu G.

[37] Kromě vrcholů lze v daném grafu barvit i hrany či stěny. V případě hranového barvení je každá hrana incidentní s vrcholem v obarvena jinou barvou. My se tím zabývat nebudeme.

Vlastnosti $\chi(G)$

Základní vlastnosti $\chi(G)$ neorientovaného grafu $G = \langle V, E \rangle$:

1. Chromatické číslo 1 mají pouze grafy, kde $|E| = 0$. Tyto grafy, sestavající z izolovaných uzlů, nazýváme **diskrétní**.
2. Stromy, kde $|V| > 1$, mají chromatické číslo 2. (Pro strom s jediným vrcholem je zřejmě $\chi(G) = 1$.)
3. $\chi(G) \geq 3$, právě když G obsahuje kružnici liché délky.
4. $\chi(G) \leq 4$ pro libovolný planární graf[38] (podle věty o čtyřech barvách).
5. $\chi(G) = |V|$ pro libovolný úplný graf G.[39]
6. $\chi(G) - 1 \leq m$, kde m je maximální stupeň vrcholu v G.

Konkrétní příklady na přednášce.

---

[38] Graf je **planární** (též **rovinný**), lze-li jej nakreslit v rovině tak, že se žádné dvě hrany neprotínají.

[39] **Úplný** graf je neorientovaný graf, v němž jsou každé dva různé vrcholy spojené hranou.

Algoritmus sekvenčního vrcholového barvení grafu

Stanovení $\chi(G)$ u obecného grafu je NP-úplný problém. Přesné algoritmy určující $\chi(G)$ mají exponenciální časovou složitost. Pro praktické účely tak vznikají heuristické algoritmy, které problém vyřeší poměrně rychle (avšak bez záruky přesného řešení). Následuje volný popis jednoho z těchto algoritmů:

- Nejprve je skóre daného neorientovaného grafu $G = \langle V, E \rangle$ uspořádáno do nerostoucí konečné posloupnosti. Pro $|V| = n$ jsou tedy stupně vrcholů grafu G seřazeny takto: $d_1 \geq d_2 \geq d_3 \geq \cdots \geq d_n$. Nechť jim odpovídající vrcholy tvoří konečnou posloupnost $(v_i)_{i=1}^n$.
- Barva 1 je přiřazena vrcholu $v_1 \in V$, jehož stupeň je d_1.
- Dokud nejsou přiřazeny barvy všem n vrcholům grafu G, postupně se (zleva doprava) prochází posloupnost $(v_i)_{i=1}^n$ dosud neobarvených vrcholů, se snahou přiřadit již použitou barvu. Pokud použitou barvu nelze přiřadit je přiřazena barva nová (dosud nepoužitá).

Konkrétní příklad na přednášce.

Poznámka. Předchozí (heuristický, hladový) algoritmus[40] spadá do třídy algoritmů, které barví vrcholy grafu iterativně s kvadratickou časovou složitostí. U těchto algoritmů je klíčovou otázkou, jak vybrat dosud neobarvený vrchol. Lépe než použít náhodný výběr je právě upřednostnění vrcholu s aktuálně nejvyšším stupněm, nebo například výběr vrcholu s nejnižším počtem dosud neobarvených sousedů. Smyslem takového počínání je zbavit se nejprve nejobtížnějších vrcholů, u nichž by odkládání obarvení mohlo blokovat řešení, které se blíží optimu.

[40] Na podobném principu funguje i tzv. Welsh-Powellův algoritmus z roku 1967. Viz https://www.youtube.com/watch?v=CQIW2mLfG04

Algoritmus vrcholového barvení grafu pomocí nezávislých množin

Množina vrcholů N se nazývá **nezávislá**, právě když neexistuje hrana, která by spojovala dva vrcholy ležící v množině N. Pro každé obarvení grafu je zřejmě množina vrcholů obarvených stejnou barvou nezávislá.

Následuje volný popis algoritmu vrcholového barvení grafu pomocí nezávislých množin:

- Zvolíme neobarvený vrchol v (podobně jako u sekvenčního barvení) a určíme největší nezávislou množinu $N(v)$ vrcholů obsahující v. Všechny vrcholy z množiny $N(v)$ obarvíme novou barvou.
- Postup opakujeme s dosud neobarvenými vrcholy, dokud nejsou všechny vrcholy obarveny.

Konkrétní příklad na přednášce.

Aplikace související s barvením grafu

Barvení grafů lze modifikovat a použít na řešení následujících praktických problémů:

- skladování nebezpečných látek, z nichž se některé mohou vzájemně ovlivňovat
- podávání léků, z nichž některé spolu nelze kombinovat
- plánování procesů s jedním zdrojem, které nemohou probíhat naráz
- tvorba rozvrhu hodin, plánování schůzek a jednání
- barvení map
- řízení světelných křižovatek
- řešení Sudoku (hranami jsou spojena všechna čísla (vrcholy) v rámci jednoho řádku/sloupce/buňky)
- určování frekvencí vysílačů mobilního signálu, kdy dva na sebe „vidící“ vysílače nemohou mít stejnou frekvenci. Je chtěno, aby bylo použito co nejméně různých frekvencí. Poznamenejme, že v městských částech najdeme poměrně často třeba pětici vysílačů, které na sebe všechny současně „vidí“.

Obsah

Úvodní motivace

Sítí rozumíme orientovaný graf (představující „potrubí“), ve kterém je užitečné umět nalézt tzv. maximální tok. Souvisí s tím například přenos dat v internetu, průchodnost vodovodu či ropovodu, případně silniční nebo železniční sítě, kde si lze vrcholy grafu představit třeba jako křižovatky silnic nebo železniční uzly.

Zmíněné „potrubí“ má danou kapacitu, kolik látky (dat, vody, ropy, zboží apod.) může v daných úsecích rozvést. Látka putuje vždy ze startu do cíle; těmto vrcholům se v terminologii sítí říká zdroj a stok (též spotřebič). Nikde jinde, než v těchto dvou různých místech látka nevzniká, ani neubývá.

Budeme se zabývat problémem, jak v dané síti nalézt největší tok ze zdroje do stoku vzhledem ke známé kapacitě (propustnosti) hran.

Definice sítě

Definice

Síť je orientovaný graf $G = \langle V, E \rangle$, který

- je hranově ohodnocen zobrazením $c : E \to \mathbb{R}_0^+$, zvaným **kapacita hran**
- obsahuje dva různé vrcholy $z \in V$, $s \in V$, zvané **zdroj** a **stok** (spotřebič), přičemž do z žádné hrany nevstupují a z vrcholu s žádné hrany nevystupují.

Vrcholy sítě různé od zdroje a stoku nazýváme **vnitřní vrcholy**.

Poznámka. V prakticky zaměřených úlohách se může vyskytovat více zdrojů. V definici uvedený jeden zdroj však postačuje, neboť z něj mohou vést hrany do ostatních zdrojů, přičemž pak původní zdroje mohou mít své vlastní kapacity. Podobně se lze vypořádat s více stoky.

Definice toku v síti a definice nenasycené cesty

Definice

Tok v síti je zobrazení $t : E \to \mathbb{R}_0^+$, které

- každé hraně $e \in E$ přiřazuje číslo $t(e)$, pro které platí $0 \leq t(e) \leq c(e)$, kde $c(e)$ je kapacita hrany e
- pro každý vnitřní vrchol $v \in V$ splňuje podmínku, že součet toků hranami vstupujícími do v je roven součtu toků hranami vystupujícími z v.[a]

[a] Pro vnitřní vrcholy je tak splněn tzv. zákon kontinuity.

Definice

- Je-li $t(e) < c(e)$ (tedy tok hrany je ostře menší než kapacita hrany), pak řekneme, že hrana e je **nenasycená** tokem t.
- **Nenasycenou cestou** směřující ze zdroje z do stoku s rozumíme orientovanou cestu, jejíž všechny hrany jsou nenasycené.

Maximální tok v síti a minimální řez

Nechť symbol $|t|$ označuje tzv. **velikost toku** t, což je celkové množství, které odteče ze zdroje z (nebo přiteče do stoku s). Pak tok t nazveme **maximální**, jestliže pro každý jiný tok t^* v dané síti platí $|t^*| \leq |t|$. Nyní zavedeme pojem velikost řezu, který s velikostí toku úzce souvisí.

Definice

Řez v síti je vlastní podmnožina H množiny všech hran E taková, že v podgrafu $G - H$ (kde jsou z G odebrány všechny hrany z H) nezbude žádná orientovaná cesta ze zdroje z do stoku s. **Velikost řezu** H je součet kapacit všech hran z H, tedy číslo $\sum_{e \in H} c(e)$.

Věta

Maximální velikost toku v síti je rovna *minimální* velikosti řezu.

Tuto větu nebudeme dokazovat. Hlavní myšlenka důkazu je ale v souladu s intuicí a je patrná z řešení konkrétních příkladů.

Algoritmus pro hledání maximálního toku v síti

Následuje volný popis jednoho algoritmu[41] pro hledání maximálního toku t v síti orientovaného grafu $G = \langle V, E \rangle$ s kapacitou hran c, se zdrojem z a stokem s:

- Nejprve nastavíme pro každou hranu $e \in E$ hodnotu $t(e) = 0$.
- Dokud lze vybrat jednu z nejkratších nenasycených cest směřujících ze z do s provádíme v cyklu následující. Pro hrany vyskytující se v dané cestě zvětšíme hodnotu toku t o největší možnou přípustnou hodnotu (danou aktuálním stavem toku a nastavením kapacit dotčených hran).
- V případě, že (už) žádná nenasycená cesta neexistuje, algoritmus vypíše na výstup aktuální (maximální) tok t a skončí.

Konkrétní příklad na přednášce.

[41] Zde uvedený Edmonds-Karpův algoritmus je vylepšenou verzí Ford-Fulkersonova algoritmu pro hledání maximálního toku v síti.
Viz https://www.youtube.com/watch?v=RppuJYwlcI8

Poznámka k možným modifikacím u sítí a toků v nich

Některé modifikace u sítí a toků v nich:

1. U sítí může být užitečné zadávat i kapacity vrcholů. Jednotlivými vrcholy pak nemůže protéct více než je povolené množství. Tuto síť je ale snadné převést na klasickou síť. Stačí totiž „zdvojit“ všechny vnitřní vrcholy a hrany mezi nimi ohodnotit kapacitami původních vrcholů.
2. U hran sítě je uveden požadavek na minimální kapacitu, tedy dolní mez toku. (Třeba, když je chtěno, aby nedocházelo k zanesení potrubí.)
3. V síti je řešena přeprava více látek současně, což vede na zajímavý (a složitý) problém vícekomoditních toků v síti.

Reference

- Část o logice byla zpracována s využitím textu R. Bělohlávka: Matematická logika – poznámky k přednáškám, 2004. A dle učebního textu R. Bělohlávka a V. Vychodila: Diskrétní matematika pro informatiky I a II, Olomouc 2006.
- Část o složitosti byla zpracována s využitím textu P. Martinka: Základy teoretické informatiky, Olomouc 2006.
- Část o algebraických strukturách byla zpracována s využitím textu J. Rachůnka: Grupy a okruhy. VUP Olomouc, 2005.
- Část o posloupnostech a řadách byla zpracována dle textu S. Trávníčka: Matematická analýza 1, 1997.