

2 Matematická logika a důkazy

Jak jsme již poznali, výrokové logice chybí možnost „parametrizace“ výroků, což znamená zavedení omezeného a dobře definovaného kontextu pro každé tvrzení, ve kterém pak můžeme už jednotlivé výroky vyhodnocovat podle pravidel výrokové logiky.

Takovou parametrizaci si lze v jednoduché ukázce představit jako proměnnou X , jež třeba nabývá hodnot ze skupiny našich studentů a umožňuje nám hovořit v různých výrocích o stejném nespécifikovaném studentovi X ... □

Obecně se tak dostáváme na „vyšší úroveň“ predikátové logiky.

$$\forall X \exists Y \dots$$

□

Stručný přehled lekce

- * Vysvětlení kvantifikátorů a zjednodušené střípky predikátové logiky.
- * Normální tvar formule, mechanická negace formulí.
- * Jak vymýšlet (tvořit) a správně formulovat matematické důkazy.

2.1 Kvantifikace a predikátová logika

Dříve popsaná výroková logika je velmi omezená faktem, že každý výrok musí být (tzv. absolutně) vyhodnocen jako pravda nebo nepravda.

- Predikátová logika pracuje s *predikáty*. Predikáty jsou „parametrizované výroky“, které jsou buď pravdivé nebo nepravdivé pro každou konkrétní volbu parametrů. □ Výrok. prom. lze chápat jako predikáty bez parametrů. □

Pro neformální přiblížení si uvedeme několik ukázek predikátů:

- * $x > 3$ (parametrem je zde $x \in \mathbb{R}$), □
- * ‘čísla x a y jsou nesoudělná’ (parametry $x, y \in \mathbb{N}$),
- * obecně jsou predikáty psány $P(x, y)$, kde x, y jsou libovolné parametry. □

Následně pak můžeme psát formule ve stylu (a více viz následující definice)

- * $(P(x, y) \wedge Q(x, z)) \rightarrow R(y, z)$,
- * $(P(x, y) \wedge Q(y, z)) \rightarrow \neg R(x, z)$. □

Parametrům predikátů se také říká *proměnné*, ale je třeba dávat pozor, aby se nepletly s výrokovými proměnnými, jedná se o jiné objekty.

Syntaxe a sémantika predikátové logiky

Definice: Z predikátů (zahrnují i výrokové proměnné) lze vytvářet *predikátové formule* pomocí už známých výrokových spojek a takzvaných *kvantifikátorů*. □

Přesněji, necht' φ je formule výrokové či predikátové logiky, pak také

* (*všeobecný kvantifikátor*) formule $\forall x . \varphi$ a □

* (*existenční kvantifikátor*) formule $\exists x . \varphi$

jsou formulemi predikátové logiky. □

Pro ukázkou složitějšího zápisu si uvedeme například predikátovou formuli:

$$\forall x . \exists y . (x + 2 < y \wedge \forall z . (x + y + z > 2024))$$

Náš zápis predikátových formulí ve stylu $\forall x . \varphi$ není jediný možný, často se také setkáte se zápisy $\forall x : \varphi$ nebo $\forall x (\varphi)$. Všechny tyto tři způsoby můžete používat.

Definice 2.1. Zjednodušená sémantika (význam) predikátové logiky.

Uvažujme libovolnou množinu (třídu) \mathcal{U} , zvanou *univerzum*, a předpokládejme, že každý predikátový parametr (proměnná) x, y, \dots má zvolenou hodnotu z \mathcal{U} . Vyhodnocení predikátové formule σ pak definujeme induktivně takto:

- Každý predikát $P(x, y, \dots)$ se vyhodnotí (na 0 nebo 1) jako výrok v aktuálním kontextu, tj. vzhledem k aktuálně zvoleným hodnotám svých parametrů x, y, \dots . \square
- Výrokové spojky jsou vyhodnoceny podle pravidel Definice 1.8. \square
- Formule tvaru $\forall x. \varphi$ se vyhodnotí jako pravdivá (1), právě když pro každou volbu parametru $x \in \mathcal{U}$ je pravdivá formule φ . \square
- Formule tvaru $\exists x. \varphi$ se vyhodnotí jako pravdivá (1), právě když existuje alespoň jedna volba parametru $x \in \mathcal{U}$, pro kterou je pravdivá formule φ . \square

Všimněte si, že Definice 2.1 nedává konzistentní návod na vyhodnocení takové predikátové formule, ve které se některá proměnná vyskytuje mimo kvantifikátor. \square

Fakt: Je-li každá proměnná – parametr predikátu – v dané formuli kvantifikovaná (tj. formule je *uzavřená*), pak je formule buď pravdivá nebo nepravdivá.

Příklad 2.2. Ukažme si vyjádření násl. slovních výroků v predikátové logice:

- Výrok „každé prvočíslo větší než 2 je liché“ za použití predikátů $Pr(n)$ a $Li(n)$ vyjadřujících po řadě prvočíselnost a lichost n ;

$$\forall n [(n \in \mathbb{N} \wedge Pr(n) \wedge n > 2) \Rightarrow Li(n)], \quad \square$$

přičemž lze rozepsat $Li(n) \equiv \exists k \in \mathbb{N} (n = 2k + 1)$. \square

- Každé celé číslo $n > 1$, které není prvočíslem, je dělitelné nějakým celým číslem y takovým, že $n \neq y$ a $y > 1$;

$$\forall n [(n \in \mathbb{Z} \wedge n > 1 \wedge \neg Pr(n)) \Rightarrow \exists y (y \in \mathbb{Z} \wedge y > 1 \wedge n \neq y \wedge y | n)].$$

\square

\square

Příklad 2.3. Proč na pořadí kvantifikátorů *velmi záleží*:

- Pro každého studenta A v posluchárně platí, že existuje student B v posluchárně takový, že A je kamarád B . \square
- Existuje student B v posluchárně, že pro každého studenta A v posluchárně platí, že A je kamarád B . \square

Konvence zápisu predikátových formulí

- Pokud píšeme více kvantifikátorů za sebou, vynecháváme zbytečně opakované symboly jako $\forall x \forall y \forall z \exists s \exists t . \varphi$,
nebo dokonce $\forall x, y, z \exists s, t (\varphi)$. \square
- Jsou-li ve formuli φ některé parametry predikátů bez kvantifikace („nezavřené“), například x, y, \dots , pak mohou být v zápise zdůrazněny jako parametry samotné formule $\varphi(x, y, \dots)$.
Například $\varphi(x, y) \equiv \exists z (x < z < y)$. \square
- Další možná modifikace se týká univerza pro hodnoty kvantifikovaných proměnných. Zatímco striktně bychom měli psát $\forall n (n \in \mathbb{Z} \Rightarrow n^2 \geq n)$, často vidíme jiný možný zápis $\forall n \in \mathbb{Z} (n^2 \geq n)$.

2.2 Normální tvar logických formulí

Přesný význam tvrzení se zanořenými negacemi je někdy skutečně obtížné pochopit. □

„Není pravda, že nemohu neříct, že není pravda, že tě nemám nerad.“ □

Výrokové formule se proto obvykle prezentují v tzv. normálním tvaru, ve kterém se negace zanořených podformulí nevyskytují, formálně: □

Definice: Formule φ je v *normálním tvaru*, pokud se v ní operátor negace aplikuje pouze na výrokové proměnné (případně na predikáty).

- Pro ilustraci, k formuli $\neg(A \Rightarrow B)$ je ekvivalentní normální tvar $A \wedge \neg B$, □
- k formuli $\neg(C \wedge (\neg A \Rightarrow B))$ je ekvivalentní $\neg C \vee (\neg A \wedge \neg B)$, □
- k formuli $\neg((A \Rightarrow B) \Rightarrow C)$ je ekvivalentní $(A \Rightarrow B) \wedge \neg C$ □
- a pokud důsledně aplikujeme přirozené pravidlo dvojí negace ($\models \neg\neg A \Leftrightarrow A$), tak výše napsané tvrzení „... nemám nerad“ si převedeme na lépe srozumitelný tvar:

„Nemusím říct, že tě mám nerad.“

Negace formule v normálním tvaru

Použitím následujících neformálních „přepisovacích“ pravidel dokážeme převést každou formuli predikátové logiky na normální tvar:

- Je-li (již znegovaná) formule tvaru $\neg(\varphi \Rightarrow \psi)$, bude přepsána na $(\varphi \wedge \neg\psi)$: \square

$$\neg(\varphi \Rightarrow \psi) \quad \rightsquigarrow \quad \varphi \wedge \neg\psi$$

- Podobně:

$$\neg(\varphi \vee \psi) \quad \rightsquigarrow \quad \neg\varphi \wedge \neg\psi \quad \square$$

$$\neg(\varphi \wedge \psi) \quad \rightsquigarrow \quad \neg\varphi \vee \neg\psi \quad \square$$

- Pro kvantifikátory se pak použije následující intuitivní převod:

$$\neg(\exists x . \varphi) \quad \rightsquigarrow \quad \forall x . \neg\varphi \quad \square$$

$$\neg(\forall x . \varphi) \quad \rightsquigarrow \quad \exists x . \neg\varphi \quad \square$$

Plný formální popis této metody převodu do normálního tvaru ponecháváme na budoucí Oddíl 6.4.

$$\neg(\varphi \Rightarrow \psi) \quad \rightsquigarrow \quad \varphi \wedge \neg\psi$$

$$\neg(\varphi \vee \psi) \quad \rightsquigarrow \quad \neg\varphi \wedge \neg\psi$$

$$\neg(\varphi \wedge \psi) \quad \rightsquigarrow \quad \neg\varphi \vee \neg\psi$$

Uvažme výrokovou formuli $\neg(A \Rightarrow \neg(B \vee \neg(C \Rightarrow \neg A)))$. Užitím uvedeného postupu ji upravíme takto: \square

$$\neg(A \Rightarrow \neg(B \vee \neg(C \Rightarrow \neg A))) \quad \rightsquigarrow \quad A \wedge \neg(\neg(B \vee \neg(C \Rightarrow \neg A))) \quad \rightsquigarrow \square$$

$$A \wedge (B \vee \neg(C \Rightarrow \neg A)) \quad \rightsquigarrow \quad A \wedge (B \vee (C \wedge \neg(\neg A))) \quad \rightsquigarrow \square$$

$$A \wedge (B \vee (C \wedge A))$$

Obdobně uvažme predikátovou formuli $\neg(\forall x \exists y \neg P(x, y))$. Tu užitím našeho postupu upravíme následovně: \square

$$\neg(\forall x \exists y \neg P(x, y)) \quad \rightsquigarrow \quad \exists x \neg(\exists y \neg P(x, y)) \quad \rightsquigarrow \square$$

$$\exists x \forall y \neg(\neg P(x, y)) \quad \rightsquigarrow \quad \exists x \forall y P(x, y)$$

2.3 Tvoření matematických důkazů

Jak **moc formální** mají správné matematické důkazy vlastně být? □

- Záleží, komu je důkaz určen — **konzument** musí být schopen „snadno“ ověřit korektnost každého tvrzení v důkazu a plně pochopit, z čeho vyplývá.
- Je tedy hlavně na vás zvolit tu správnou úroveň formálnosti zápisu vět i důkazů podle situace. □
- Avšak vůbec **neplatí**, že čím více formálních matematických symbolů v důkazu použijete místo běžného jazyka, tím by byl důkaz přesnější! □

A jak na ten správný matematický důkaz máme přijít?

- No... , □nalézání matematických důkazů je tvůrčí činnost, která není vůbec snadná a jako taková vyžaduje tvůrčí (přímo „**umělecké**“) matematické vlohy. I pokud takové vlohy (zatím) v sobě necítíte, snažte se jí alespoň trochu přiučit.

Dokazovat či vyvracet tvrzení?

Představme si, že našim úkolem je rozhodnout platnost matematického tvrzení. Jak pak matematicky správně zdůvodníme nalezenou odpověď?

- Záleží na odpovědi samotné... □
- Pokud je to ANO (platí), prostě podáme důkaz podle uvedených zvyklostí.
- Pokud je odpověď NE, tak naopak podáme důkaz *negace* daného tvrzení. □

Poměrně častým případem je matematická věta T , která tvrdí nějaký závěr pro širokou oblast vstupních možností. Potom je postup následující: □

- Pokud T platí, nezbyvá než podat *vyčerpávající důkaz* platnosti pro všechny vstupy. □
- Avšak pokud T je nepravdivá, stačí *uhodnout* vhodný *protipříklad* a jen pro něj dokázat, že při platnosti všech předpokladů závěr tvrzení platný není. Ke slovíčku „stačí“ ještě dodáváme, že pro správné matematické vyvrácení nepravdivého tvrzení T protipříklad uvést přímo **musíte**

(viz pravidlo $\neg(\forall x . \varphi) \rightsquigarrow \exists x . \neg\varphi$).

Příklad 2.5. Rozhodněte platnost následujícího tvrzení: Pro všechna reálná x platí

$$x^2 + 3x + 2 \geq 0. \square$$

Důkaz: Standardními algebraickými postupy si můžeme upravit vztah na $x^2 + 3x + 2 = (x + 1) \cdot (x + 2) \geq 0$. Co nám tato úprava naznačuje? \square Například to, že k porušení daného tvrzení stačí volit x tak, aby jedna ze závorek byla kladná a druhá záporná. To nastane třeba pro $x = -\frac{3}{2}$. \square

Pro vyvrácení tvrzení nám tedy stačí začít volbou protipříkladu reálného čísla $x = -\frac{3}{2}$ (není nutno zdůvodňovat, jak jsme jej „uhodli“!) a následně dokázat úpravou

$$x^2 + 3x + 2 = (x + 1) \cdot (x + 2) = \left(-\frac{3}{2} + 1\right) \cdot \left(-\frac{3}{2} + 2\right) = \left(-\frac{1}{2}\right) \cdot \left(+\frac{1}{2}\right) = -\frac{1}{4} < 0. \square$$

Dané tvrzení tudíž není platné. \square

Konstruktivní a existenční důkazy

Z hlediska praktické využitelnosti je vhodné rozlišovat tyto dvě kategorie důkazů (třebaže z formálně–matematického pohledu mezi nimi kvalitativní rozdíl není).

- Důkaz z Příkladu 1.5 je *konstruktivní*. Dokázali jsme nejen, že číslo z existuje, ale podali jsme také návod, jak ho pro dané x a y *sestrojit*.
- *Existenční* důkaz je takový, kde se prokáže existence nějakého objektu *bez toho*, aby byl podán použitelný návod na jeho konstrukci. □

Příklad 2.6. Čistě *existenčního* důkazu.

Věta. *Existuje program, který vypíše na obrazovku čísla tažená ve 45. tahu sportky v roce 2024.* □

Důkaz: Existuje pouze konečně mnoho možných výsledků losování 45. tahu sportky v roce 2024. Pro každý možný výsledek *existuje* program, který tento daný výsledek vypíše na obrazovku. Mezi těmito programy je tedy jistě takový, který vypíše právě ten výsledek, který bude ve 45. tahu sportky v roce 2024 skutečně vylosován. □

To je ale *podvod*, že? □ A přece *není*. . . Formálně správně to je prostě tak a tečka.