

Predikátová logika prvního řádu

Aleš Horák

E-mail: hales@fi.muni.cz
<http://nlp.fi.muni.cz/uui/>

Obsah:

- Připomínka – průběžná písemka
- Predikátová logika prvního řádu
- Příklady jiných logik
- Logika a AI

Připomínka – průběžná písemka

- termín – příští **pondělí 4. listopadu, 20:00**, D1 a D3
- **náhradní** termín: pro nemocné s **omluvenkou**, požádejte e-mailem
- **nepřihlašuje se** – vaši posluchárnu najdete v ISu u termínu zkoušky
- test obsahuje **7 otázek**, 6 za 3 body a 1 za 2 body, celkem max **20 bodů** za nesprávnou odpověď se **0.5 bodu odečítá**.
- celkový čas na vypracování testu je **40 minut**.
- **nejsou povoleny** žádné materiály, ani prázdné papíry. Otázky **nesmíte** jakýmkoliv způsobem předávat nebo **šířit**.
- příklady (formou testu – odpovědi A, B, C, ...) z látky probrané na prvních **pěti přednáškách**

Obsah

- 1 Připomínka – průběžná písemka
- 2 Predikátová logika prvního řádu
 - Predikátová logika 1. řádu
 - Syntaxe predikátové logiky
 - Sémantika predikátové logiky
 - Kvantifikace
 - Substituce proměnných
 - Normální formy v predikátové logice
 - Báze znalostí v PL1
- 3 Příklady jiných logik
 - Fuzzy logika
 - Pravděpodobnost
 - Modální logiky
 - Logiky vyšších řádů
 - Temporální logiky a Intenzionální logiky
- 4 Logika a AI
 - Znalostní inženýrství

Výhody a nevýhody výrokové logiky

- 😊 výroková logika je **deklarativní**: syntaxe přímo koresponduje s fakty

Výhody a nevýhody výrokové logiky

- 😊 výroková logika je **deklarativní**: syntaxe přímo koresponduje s fakty
- 😊 výroková logika umožňuje zpracovávat částečné/disjunktivní/negované **informace** (což je víc, než umí většina datových struktur a databází)

Výhody a nevýhody výrokové logiky

- 😊 výroková logika je **deklarativní**: syntaxe přímo koresponduje s fakty
- 😊 výroková logika umožňuje zpracovávat částečné/disjunktivní/negované **informace** (což je víc, než umí většina datových struktur a databází)
- 😊 výroková logika je **kompoziční**:

Výhody a nevýhody výrokové logiky

- 😊 výroková logika je **deklarativní**: syntaxe přímo koresponduje s fakty
- 😊 výroková logika umožňuje zpracovávat částečné/disjunktivní/negované **informace** (což je víc, než umí většina datových struktur a databází)
- 😊 výroková logika je **kompoziční**:
význam $P_1 \wedge P_2$ je odvozen z významu P_1 a P_2

Výhody a nevýhody výrokové logiky

- 😊 výroková logika je **deklarativní**: syntaxe přímo koresponduje s fakty
- 😊 výroková logika umožňuje zpracovávat částečné/disjunktivní/negované **informace** (což je víc, než umí většina datových struktur a databází)
- 😊 výroková logika je **kompoziční**:
význam $P_1 \wedge P_2$ je odvozen z významu P_1 a P_2
- 😊 ve výrokové logice je význam **kontextově nezávislý**

Výhody a nevýhody výrokové logiky

- 😊 výroková logika je **deklarativní**: syntaxe přímo koresponduje s fakty
- 😊 výroková logika umožňuje zpracovávat částečné/disjunktivní/negované **informace** (což je víc, než umí většina datových struktur a databází)
- 😊 výroková logika je **kompoziční**:
význam $P_1 \wedge P_2$ je odvozen z významu P_1 a P_2
- 😊 ve výrokové logice je význam **kontextově nezávislý**
- 😞 výroková logika má velice **omezenou expresivitu** (narozdíl od přirozeného jazyka)
např. nemáme jak říct "**Jámy způsobují Vánek ve vedlejších místnostech**" jinak, než vyjmenovat odpovídající výrok pro každé pole

Výhody a nevýhody přirozeného jazyka

- 😊 většina lidí jazyk běžně používá k **vyjádření** myšlenek a k **vyvozování** závěrů

Výhody a nevýhody přirozeného jazyka

- 😊 většina lidí jazyk běžně používá k **vyjádření** myšlenek a k **vyvozování** závěrů
- 😊 v jazyce **umíme vyjádřit** (nebo aspoň popsat) téměř všechny myšlenky – má velkou **expresivitu**

Výhody a nevýhody přirozeného jazyka

- 😊 většina lidí jazyk běžně používá k **vyjádření** myšlenek a k **vyvozování** závěrů
- 😊 v jazyce **umíme vyjádřit** (nebo aspoň popsat) téměř všechny myšlenky – má velkou **expresivitu**
- 😞? jazyk je spíš prostředek **komunikace** než (jen) reprezentace
např. **Podívej!** (... *nad střechou se objevil Superman*)
věta bez daného **kontextu** nemusí nést (stejnou) informaci

Výhody a nevýhody přirozeného jazyka

- 😊 většina lidí jazyk běžně používá k **vyjádření** myšlenek a k **vyvozování** závěrů
- 😊 v jazyce **umíme vyjádřit** (nebo aspoň popsat) téměř všechny myšlenky – má velkou **expresivitu**
- 😞? jazyk je spíš prostředek **komunikace** než (jen) reprezentace
např. **Podívej!** (... *nad střechou se objevil Superman*)
věta bez daného **kontextu** nemusí nést (stejnou) informaci
- 😞 jazyk má velkou **víceznačnost** (*ambiguity*)
viz **matka**, **zaječí**, **travička**, ...

Výhody a nevýhody přirozeného jazyka

- 😊 většina lidí jazyk běžně používá k **vyjádření** myšlenek a k **vyvozování** závěrů
- 😊 v jazyce **umíme vyjádřit** (nebo aspoň popsat) téměř všechny myšlenky – má velkou **expresivitu**
- 😐? jazyk je spíš prostředek **komunikace** než (jen) reprezentace
např. **Podívej!** (... *nad střechou se objevil Superman*)
věta bez daného **kontextu** nemusí nést (stejnou) informaci
- 😞 jazyk má velkou **víceznačnost** (*ambiguity*)
viz **matka**, **zaječí**, **travička**, ...
- 😐? lidé si pamatují **obsah**, ale ne přesnou **formu**

Výhody a nevýhody přirozeného jazyka

- 😊 většina lidí jazyk běžně používá k **vyjádření** myšlenek a k **vyvozování** závěrů
- 😊 v jazyce **umíme vyjádřit** (nebo aspoň popsat) téměř všechny myšlenky – má velkou **expresivitu**
- 😐? jazyk je spíš prostředek **komunikace** než (jen) reprezentace
např. **Podívej!** (... *nad střechou se objevil Superman*)
věta bez daného **kontextu** nemusí nést (stejnou) informaci
- 😞 jazyk má velkou **víceznačnost** (*ambiguity*)
viz **matka**, **zaječí**, **travička**, ...
- 😐? lidé si pamatují **obsah**, ale ne přesnou **formu**
byla první věta předchozího slajdu
“**Výroková logika je deklarativní – syntaxe přímo koresponduje s fakty**”, nebo
“**Syntaxe výrokové logiky je deklarativní, jelikož přímo vyjadřuje fakta**”?

Výhody a nevýhody přirozeného jazyka

- 😊 většina lidí jazyk běžně používá k **vyjádření** myšlenek a k **vyvozování** závěrů
- 😊 v jazyce **umíme vyjádřit** (nebo aspoň popsat) téměř všechny myšlenky – má velkou **expresivitu**
- 😞? jazyk je spíš prostředek **komunikace** než (jen) reprezentace
např. **Podívej!** (... *nad střechou se objevil Superman*)
věta bez daného **kontextu** nemusí nést (stejnou) informaci
- 😞 jazyk má velkou **víceznačnost** (*ambiguity*)
viz **matka**, **zaječí**, **travička**, ...
- 😞? lidé si pamatují **obsah**, ale ne přesnou **formu**
- 😞 konkrétní **forma** přitom **ovlivňuje** vyvozování
Jakou rychlostí jela auta než nastal kontakt? vs ... **než se o sebe roztřískala?**

Predikátová logika prvního řádu

- **First-order predicate logic**, FOPL/PL1
- **vyšší expresivita** než výroková logika, **nižší složitost** než přirozený jazyk
- umožňuje **strukturovat** jednoduché výroky

PJ: Každý člověk je smrtelný
a Sokrates je člověk,
proto Sokrates je smrtelný.

Predikátová logika prvního řádu

- **First-order predicate logic**, FOPL/PL1
- vyšší expresivita než výroková logika, nižší složitost než přirozený jazyk
- umožňuje strukturovat jednoduché výroky

PJ: Každý člověk je smrtelný
a Sokrates je člověk,
proto Sokrates je smrtelný.

VL: (\checkmark
 $\wedge So$)
 $\Rightarrow Sm$

Predikátová logika prvního řádu

- **First-order predicate logic**, FOPL/PL1
- vyšší expresivita než výroková logika, nižší složitost než přirozený jazyk
- umožňuje strukturovat jednoduché výroky

PJ: Každý člověk je smrtelný
a Sokrates je člověk,
proto Sokrates je smrtelný.

VL: $(\checkmark$
 $\wedge So)$
 $\not\Rightarrow Sm$

Predikátová logika prvního řádu

- **First-order predicate logic**, FOPL/PL1
- **vyšší expresivita** než výroková logika, **nižší složitost** než přirozený jazyk
- umožňuje **strukturovat** jednoduché výroky

PJ: Každý člověk je smrtelný
a Sokrates je člověk,
proto Sokrates je smrtelný.

VL: (\checkmark
 $\wedge So$)
 $\not\Rightarrow Sm$

PL1: $(\forall x \text{ } \checkmark(x) \Rightarrow sm(x))$
 $\wedge \checkmark(So)$
 $\Rightarrow sm(So)$

Predikátová logika prvního řádu

- **First-order predicate logic**, FOPL/PL1
- **vyšší expresivita** než výroková logika, **nižší složitost** než přirozený jazyk
- umožňuje **strukturovat** jednoduché výroky

PJ: Každý člověk je smrtelný a Sokrates je člověk, proto Sokrates je smrtelný.	VL: (Č \wedge So) $\not\Rightarrow$ Sm	PL1: $(\forall x \text{ čl}(x) \Rightarrow \text{sm}(x))$ $\wedge \text{čl}(\text{So})$ $\Rightarrow \text{sm}(\text{So})$
--	--	--
- výroková logika \rightarrow svět (ontologie) obsahuje **fakty** \times PL1 předpokládá, že svět obsahuje:
 - **objekty** – lidi, domy, teorie, barvy, roky, ...
 - **relace** – červený, kulatý, prvočíselný, bratři, větší než, uvnitř, ...
 - **funkce** – otec někoho, nejlepší přítel, plus jedna, začátek čeho, ...

Predikátová logika prvního řádu

- **First-order predicate logic**, FOPL/PL1
- **vyšší expresivita** než výroková logika, **nižší složitost** než přirozený jazyk
- umožňuje **strukturovat** jednoduché výroky

PJ: Každý člověk je smrtelný
a Sokrates je člověk,
proto Sokrates je smrtelný.

VL: (\checkmark
 $\wedge So$)
 $\not\Rightarrow Sm$

PL1: $(\forall x \text{ } \checkmark(x) \Rightarrow sm(x))$
 $\wedge \checkmark(So)$
 $\Rightarrow sm(So)$

- výroková logika \rightarrow svět (ontologie) obsahuje **fakty** \times PL1 předpokládá, že svět obsahuje:
 - **objekty** – lidi, domy, teorie, barvy, roky, ...
 - **relace** – červený, kulatý, prvočíselný, bratři, větší než, uvnitř, ...
 - **funkce** – otec někoho, nejlepší přítel, plus jedna, začátek čeho, ...

Jedna plus dva rovná se tři: objekty **jedna, dva, tři, jedna plus dva**;
relace **rovná se**; funkce **plus**

Predikátová logika prvního řádu

- **First-order predicate logic**, FOPL/PL1
- **vyšší expresivita** než výroková logika, **nižší složitost** než přirozený jazyk
- umožňuje **strukturovat** jednoduché výroky

PJ: Každý člověk je smrtelný
a Sokrates je člověk,
proto Sokrates je smrtelný.

VL: (\checkmark
 $\wedge So$
 $\not\Rightarrow Sm$)

PL1: ($\forall x \text{ } \checkmark(x) \Rightarrow sm(x)$
 $\wedge \checkmark(So)$
 $\Rightarrow sm(So)$)

- výroková logika \rightarrow svět (ontologie) obsahuje **fakty** \times PL1 předpokládá, že svět obsahuje:
 - **objekty** – lidi, domy, teorie, barvy, roky, ...
 - **relace** – červený, kulatý, prvočíselný, bratři, větší než, uvnitř, ...
 - **funkce** – otec někoho, nejlepší přítel, plus jedna, začátek čeho, ...

Jedna plus dva rovná se tři: objekty **jedna, dva, tři, jedna plus dva**;
relace **rovná se**; funkce **plus**

Pozice vedle Wumpuse zapáchají: objekty **pozice, Wumpus**; relace **zapáchat, vedle**

Syntaxe predikátové logiky



*Richardus I. D. G. Rex Ang.
Dux Norman: etc Dom: Hib:*



*Johannes. D. G. Rex Ang.
Dux Norman: etc Dom: Hib:*

Syntaxe predikátové logiky

• základní prvky –

konstanty	KingJohn, 2, RichardTheLionheart, ...
funktory predikátů	Brother, >, ...
funkce	Sqrt, LeftLegOf, ...
proměnné	x, y, a, b, \dots
spojky	$\wedge \vee \neg \Rightarrow \Leftrightarrow$
rovnost	=
kvantifikátory	$\forall \exists$

výroková logika:

$S \rightarrow A \mid C$

$A \rightarrow \text{True} \mid \text{False} \mid P \mid Q \mid R \mid \dots$

$C \rightarrow (S) \mid \neg S$

$\mid S \wedge S \mid S \vee S$

$\mid S \Rightarrow S \mid S \Leftrightarrow S$

• atomické formule –

predikáty **Brother(KingJohn, RichardTheLionheart)**

složené termy $> \left(\text{Length}(\text{LeftLegOf}(\text{Richard})), \text{Length}(\text{LeftLegOf}(\text{KingJohn})) \right)$

• složené formule – z atomických formulí pomocí spojek a kvantifikátorů

$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2, \quad \forall x P(x), \quad \exists x P(x)$

např. **Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)**

$>(1, 2) \vee \leq(1, 2)$

$>(1, 2) \wedge \neg >(1, 2)$

Slido

Sémantika predikátové logiky

pravdivost formule se určuje vzhledem k *modelu* a *interpretaci*

model obsahuje ≥ 1 objektů a relace mezi nimi

objekty modelu se označují jako **doména** modelu

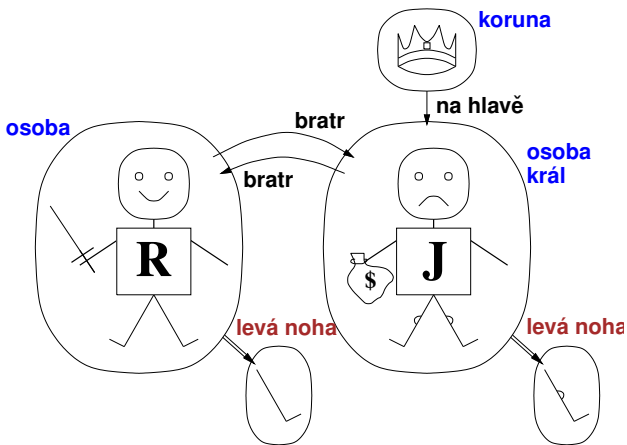
interpretace definuje vztah mezi syntaxí a modelem – určuje referenty pro:

konstantní symboly → *objekty*

predikátové symboly → *relace*

funkční symboly → *funkce*

Příklad modelu a interpretace ve FOPL



5 objektů, 2 binární relace, 3 unární relace (osoba, král, koruna) a 1 unární funkce (levá noha).

Sémantika predikátové logiky

- **konstanty** reprezentují **jména** objektů (individuí)
- **proměnné zastupují** jména objektů, jejich hodnoty se mohou měnit
- **funkce** reprezentují **složená jména** objektů
např. $add(1, 2)$ ($add(2, 1)$, $add(0, 3)$, ...) jsou složená jména pro konstantu 3
- poznámka: konstanty jsou nulární funkce
- **term** = **výraz** složený pouze z **funkčních symbolů**, **konstant** a **proměnných**
 $add(x, mul(y, sub(x, y), 1), z)$
- termy vyjadřují **aplikaci** funkce na argumenty, hodnoty jsou objekty

atomická **formule predikát**($term_1, \dots, term_n$) je **pravdivá**

\Leftrightarrow *objekty odkazované pomocí $term_1, \dots, term_n$ jsou v relaci pojmenované funktorem predikát*

Předpoklad uzavřeného světa

2 užitečné **předpoklady** ve spojení s bází znalostí:

- **předpoklad uzavřeného světa** (*closed world assumption*)
 - cokoliv o čem **nevíme**, že je **pravda** → bereme za dané, že je to **nepravda**
 - využitý např. v Prologu (negace jako neúspěch)
- **předpoklad jednoznačných pojmenování** (*unique names assumption*)
 - různá jména označují různé objekty

Vázaný a volný výskyt proměnných

- **podformule** formule A je libovolná spojitá část A , která je sama formulí $\exists x((\forall yP(z)) \Rightarrow R(x, y))$ má podformule:
 $\exists x((\forall yP(z)) \Rightarrow R(x, y)), (\forall yP(z)) \Rightarrow R(x, y), \forall yP(z), R(x, y), P(z)$
- **výskyt** proměnné x ve formuli A je **vázaný**, je x v A v dosahu svého kvantifikátoru
- výskyt proměnné je **volný**, není-li vázaný

např. výskyt x v předchozí formuli je vázaný
proměnné y a z jsou volné

Univerzální kvantifikace

\forall *proměnné* *formule*

“Každý na FI MU je inteligentní:” $\forall x \text{ Na}(x, \text{FI MU}) \Rightarrow \text{inteligentní}(x)$

$\forall x P$ je pravdivé v modelu $m \Leftrightarrow P$ je pravdivá pro $x =$ každý možný objekt z modelu m

zhruba odpovídá konjunkci instancí P

- $\text{Na}(\text{Petr}, \text{FI MU}) \Rightarrow \text{inteligentní}(\text{Petr})$
- $\wedge \text{Na}(\text{Honza}, \text{FI MU}) \Rightarrow \text{inteligentní}(\text{Honza})$
- $\wedge \text{Na}(\text{FI MU}, \text{FI MU}) \Rightarrow \text{inteligentní}(\text{FI MU})$
- $\wedge \dots$

Existenční kvantifikace

\exists *⟨proměnné⟩* *⟨formule⟩*

“Někdo na MFF UK je inteligentní:” $\exists x \text{ Na}(x, \text{MFF UK}) \wedge \text{inteligentní}(x)$

$\exists x P$ je pravdivé v modelu $m \iff P$ je pravdivá pro $x =$ nějaký objekt z modelu m

zhruba odpovídá disjunkci instancí P

- $\text{Na}(\text{Petr}, \text{MFF UK}) \wedge \text{inteligentní}(\text{Petr})$
- ∨ $\text{Na}(\text{Honza}, \text{MFF UK}) \wedge \text{inteligentní}(\text{Honza})$
- ∨ $\text{Na}(\text{MFF UK}, \text{MFF UK}) \wedge \text{inteligentní}(\text{MFF UK})$
- ∨ ...

Vlastnosti kvantifikací

- pozor při použití kvantifikátorů na záměnu \wedge a \Rightarrow :

	<i>dobře</i>	<i>špatně</i>	znamenalo by
“každý P je Q ”	$\forall x P \Rightarrow Q$	$\forall x P \wedge Q$	“každý je P i Q ”
“někdo P je Q ”	$\exists x (P \wedge Q)$	$\exists x (P \Rightarrow Q)$	“někdo není P nebo je Q ”

Vlastnosti kvantifikací

- pozor při použití kvantifikátorů na záměnu \wedge a \Rightarrow :

	<i>dobře</i>	<i>špatně</i>	znamenalo by
“každý P je Q ”	$\forall x P \Rightarrow Q$	$\forall x P \wedge Q$	“každý je P i Q ”
“někdo P je Q ”	$\exists x (P \wedge Q)$	$\exists x (P \Rightarrow Q)$	“někdo není P nebo je Q ”

- $\forall x \forall y$ je stejné jako $\forall y \forall x$
- $\exists x \exists y$ je stejné jako $\exists y \exists x$
- $\exists x \forall y$ **není** stejné jako $\forall y \exists x$

$\exists x \forall y \text{ má_rād}(x, y)$ – “Existuje osoba, která má ráda všechny lidi na světě.”

$\forall y \exists x \text{ má_rād}(x, y)$ – “Každého na světě má alespoň jedna osoba ráda.”
(potenciálně každého jiná)

Vlastnosti kvantifikací

- pozor při použití kvantifikátorů na záměnu \wedge a \Rightarrow :

	<i>dobře</i>	<i>špatně</i>	znamenalo by
“každý P je Q ”	$\forall x P \Rightarrow Q$	$\forall x P \wedge Q$	“každý je P i Q ”
“někdo P je Q ”	$\exists x (P \wedge Q)$	$\exists x (P \Rightarrow Q)$	“někdo není P nebo je Q ”

- $\forall x \forall y$ je stejné jako $\forall y \forall x$
- $\exists x \exists y$ je stejné jako $\exists y \exists x$
- $\exists x \forall y$ **není** stejné jako $\forall y \exists x$

$\exists x \forall y \text{ má_rád}(x, y)$ – “Existuje osoba, která má ráda všechny lidi na světě.”

$\forall y \exists x \text{ má_rád}(x, y)$ – “Každého na světě má alespoň jedna osoba ráda.”
(potenciálně každého jiná)

- dualita kvantifikátorů

oba mohou být vyjádřeny pomocí druhého

$\forall x \text{ má_rád}(x, \text{zmrzlina}) \equiv \neg \exists x \neg \text{má_rád}(x, \text{zmrzlina})$

$\exists x \text{ má_rád}(x, \text{mrkev}) \equiv \neg \forall x \neg \text{má_rád}(x, \text{mrkev})$

Substituce proměnných

- **substituce** $\sigma = \{x/add(2, 1), y/4\}$ definuje **dosazení** za volné proměnné
- pro větu S a substituci σ – $S\sigma$ označuje výsledek aplikace σ na S :

$$S = \text{chytřejší}(x, y)$$

$$\sigma = \{x/Petr, y/Honza\}$$

$$S\sigma = \text{chytřejší}(Petr, Honza)$$

Substituce proměnných

- **substituce** $\sigma = \{x/add(2, 1), y/4\}$ definuje **dosazení** za volné proměnné
- pro větu S a substituci σ – $S\sigma$ označuje výsledek aplikace σ na S :

$$S = \text{chytřejší}(x, y)$$

$$\sigma = \{x/\text{Petr}, y/\text{Honza}\}$$

$$S\sigma = \text{chytřejší}(\text{Petr}, \text{Honza})$$

- **term** t je **substituovatelný** za proměnnou x ve formuli A \Leftrightarrow nedochází ke kolizi volných proměnných v t a vázaných proměnných v A

$$S = \exists x P(x, y)$$

$$S\{y/z\} = \exists x P(x, z)$$

$$S\{y/f(z, z)\} = \exists x P(x, f(z, z))$$

nelze $S\{y/f(x, x)\} = \exists x P(x, f(x, x))$ kvůli změně vazby x

- kolizi lze řešit **přejmenováním volných proměnných** v termu t

Konjunktivní normální forma (CNF) v PL1

CNF v PL1 – **prenexová** (kvantifikátory na začátku) a **Skolemova NF** (bez \exists)

Algoritmus pro **převod** každé PL1 formule **do CNF**:

1. převedeme implikace na disjunkce: $P \Rightarrow Q \rightarrow \neg P \vee Q$
2. přesuneme \neg dovnitř k literálům: $\neg \forall x P \rightarrow \exists x \neg P$
3. přejmenujeme proměnné: $\forall x P \vee \exists x Q \rightarrow \forall x P \vee \exists y Q$
4. přesuneme kvantifikátory doleva: $\forall x P \vee \exists y Q \rightarrow \forall x \exists y P \vee Q$
5. eliminujeme \exists pomocí **Skolemizace**:
 $\exists x P(x) \rightarrow P(c_1)$
 $\forall x P(x) \Rightarrow \exists y Q(y) \rightarrow \forall x P(x) \Rightarrow Q(f(x))$
6. “zahodíme” univerzální kvantifikátory
7. roznásobíme \wedge pomocí \vee : $(P \wedge Q) \vee R \rightarrow (P \vee R) \wedge (Q \vee R)$

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x)$$

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x)$$

$$\forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x)$$

(\Rightarrow na \vee)

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x)$$

$$\forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x)$$

(\Rightarrow na \vee)

$$\forall x \exists y (P(x, y) \wedge \exists z \neg R(y)) \vee \forall x \neg Q(x)$$

(přesun negace 3 \times)

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x)$$

$$\forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x)$$

(\Rightarrow na \vee)

$$\forall x \exists y (P(x, y) \wedge \exists z \neg R(y)) \vee \forall x \neg Q(x)$$

(přesun negace 3 \times)

$$\forall x_1 \exists y (P(x_1, y) \wedge \exists z \neg R(y)) \vee \forall x_2 \neg Q(x_2)$$

(přejmenování x)

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x)$$

$$\forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x) \quad (\Rightarrow \text{ na } \vee)$$

$$\forall x \exists y (P(x, y) \wedge \exists z \neg R(y)) \vee \forall x \neg Q(x) \quad (\text{přesun negace } 3 \times)$$

$$\forall x_1 \exists y (P(x_1, y) \wedge \exists z \neg R(y)) \vee \forall x_2 \neg Q(x_2) \quad (\text{přejmenování } x)$$

$$\forall x_1 \exists y \exists z (P(x_1, y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2) \quad (\text{posun } \exists z \text{ doleva})$$

$$\forall x_1 \exists y \exists z \forall x_2 (P(x_1, y) \wedge \neg R(y)) \vee \neg Q(x_2) \quad (\text{posun } \forall x_2 \text{ doleva})$$

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\begin{aligned}
& \forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x) \\
& \quad \forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x) && (\Rightarrow \text{ na } \vee) \\
& \quad \forall x \exists y (P(x, y) \wedge \exists z \neg R(y)) \vee \forall x \neg Q(x) && (\text{přesun negace } 3\times) \\
& \quad \forall x_1 \exists y (P(x_1, y) \wedge \exists z \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{přejmenování } x) \\
& \quad \forall x_1 \exists y \exists z (P(x_1, y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{posun } \exists z \text{ doleva}) \\
& \quad \forall x_1 \exists y \exists z \forall x_2 (P(x_1, y) \wedge \neg R(y)) \vee \neg Q(x_2) && (\text{posun } \forall x_2 \text{ doleva}) \\
& \quad \forall x_1 \forall x_2 (P(x_1, f_1(x_1)) \wedge \neg R(f_1(x_1))) \vee \neg Q(x_2) && (\text{Skolemizace } y \text{ a } z)
\end{aligned}$$

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\begin{aligned}
& \forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x) \\
& \quad \forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x) && (\Rightarrow \text{ na } \vee) \\
& \quad \forall x \exists y (P(x, y) \wedge \exists z \neg R(y)) \vee \forall x \neg Q(x) && (\text{přesun negace } 3\times) \\
& \quad \forall x_1 \exists y (P(x_1, y) \wedge \exists z \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{přejmenování } x) \\
& \quad \forall x_1 \exists y \exists z (P(x_1, y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{posun } \exists z \text{ doleva}) \\
& \quad \forall x_1 \exists y \exists z \forall x_2 (P(x_1, y) \wedge \neg R(y)) \vee \neg Q(x_2) && (\text{posun } \forall x_2 \text{ doleva}) \\
& \quad \forall x_1 \forall x_2 (P(x_1, f_1(x_1)) \wedge \neg R(f_1(x_1))) \vee \neg Q(x_2) && (\text{Skolemizace } y \text{ a } z) \\
& \quad (P(x_1, f_1(x_1)) \wedge \neg R(f_1(x_1))) \vee \neg Q(x_2) && (\text{odstranění } \forall)
\end{aligned}$$

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\begin{aligned}
 & \forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x) \\
 & \quad \forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x) && (\Rightarrow \text{ na } \vee) \\
 & \quad \forall x \exists y (P(x, y) \wedge \exists z \neg R(y)) \vee \forall x \neg Q(x) && (\text{přesun negace } 3 \times) \\
 & \quad \forall x_1 \exists y (P(x_1, y) \wedge \exists z \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{přejmenování } x) \\
 & \quad \forall x_1 \exists y \exists z (P(x_1, y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{posun } \exists z \text{ doleva}) \\
 & \quad \forall x_1 \exists y \exists z \forall x_2 (P(x_1, y) \wedge \neg R(y)) \vee \neg Q(x_2) && (\text{posun } \forall x_2 \text{ doleva}) \\
 & \quad \forall x_1 \forall x_2 (P(x_1, f_1(x_1)) \wedge \neg R(f_1(x_1))) \vee \neg Q(x_2) && (\text{Skolemizace } y \text{ a } z) \\
 & \quad (P(x_1, f_1(x_1)) \wedge \neg R(f_1(x_1))) \vee \neg Q(x_2) && (\text{odstranění } \forall) \\
 & \quad (P(x_1, f_1(x_1)) \vee \neg Q(x_2)) \wedge (\neg R(f_1(x_1)) \vee \neg Q(x_2)) && (\text{roznásobení } \wedge)
 \end{aligned}$$

Konjunktivní normální forma (CNF) v PL1 – příklad

$$\begin{aligned}
& \forall x \exists y \neg (P(x, y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x) \\
& \quad \forall x \exists y \neg (\neg P(x, y) \vee \forall z R(y)) \vee \neg \exists x Q(x) && (\Rightarrow \text{ na } \vee) \\
& \quad \forall x \exists y (P(x, y) \wedge \exists z \neg R(y)) \vee \forall x \neg Q(x) && (\text{přesun negace } 3 \times) \\
& \quad \forall x_1 \exists y (P(x_1, y) \wedge \exists z \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{přejmenování } x) \\
& \quad \forall x_1 \exists y \exists z (P(x_1, y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2) && (\text{posun } \exists z \text{ doleva}) \\
& \quad \forall x_1 \exists y \exists z \forall x_2 (P(x_1, y) \wedge \neg R(y)) \vee \neg Q(x_2) && (\text{posun } \forall x_2 \text{ doleva}) \\
& \quad \forall x_1 \forall x_2 (P(x_1, f_1(x_1)) \wedge \neg R(f_1(x_1))) \vee \neg Q(x_2) && (\text{Skolemizace } y \text{ a } z) \\
& \quad (P(x_1, f_1(x_1)) \wedge \neg R(f_1(x_1))) \vee \neg Q(x_2) && (\text{odstranění } \forall) \\
& \quad \underline{(P(x_1, f_1(x_1))) \vee \neg Q(x_2)} \wedge \underline{(\neg R(f_1(x_1))) \vee \neg Q(x_2)} && (\text{roznásobení } \wedge)
\end{aligned}$$

Báze znalostí v PL1

předpokládejme, že agent ve Wumpusově jeskyni cítí Zápach a Vánek, ale nevidí Třpyt, nenarazil do zdi a nezabil Wumpuse v čase $t = 5$:

```
tell (KB, percept([zápach, vánek, nic, nic, nic], 5)).  
?- ask(KB, action(A,5)).      %  $\exists A$  action(A,5) ?
```

tj. dotaz “**Vyplývá nějaká akce z KB v čase $t = 5$?**”

Báze znalostí v PL1

předpokládejme, že agent ve Wumpusově jeskyni cítí Zápach a Vánek, ale nevidí Třpyt, nenarazil do zdi a nezabil Wumpuse v čase $t = 5$:

```
tell (KB, percept([zápach, vánek, nic, nic, nic], 5)).  
?- ask(KB, action(A,5)).      %  $\exists A$  action(A,5) ?
```

tj. dotaz “Vyplývá nějaká akce z KB v čase $t = 5$?”

odpověď: *true*, {*a/Výstřel*} ← *substitute* (hodnot proměnným)

Báze znalostí v PL1

předpokládejme, že agent ve Wumpusově jeskyni cítí Zápach a Vánek, ale nevidí Třpyt, nenarazil do zdi a nezabil Wumpuse v čase $t = 5$:

```
tell (KB, percept([zápach, vánek, nic, nic, nic], 5)).
?- ask(KB,action(A,5)).      %  $\exists A$  action(A,5) ?
```

tj. dotaz “Vyplývá nějaká akce z KB v čase $t = 5$?”

odpověď: $true, \{a/Výstřel\}$ ← **substituce** (hodnot proměnným)

ASK(KB, S) vrací některá/všechna σ takové, že $KB \models S\sigma$

Báze znalostí pro Wumpusovu jeskyni v PL1

Vnímání:

$\forall v, tr, n, w, t \text{ Percept}([Zápach, v, tr, n, w], t) \Rightarrow \text{Je_zápach}(t)$

$\forall z, v, n, w, t \text{ Percept}([z, v, Třpyt, n, w], t) \Rightarrow \text{Máme_zlato}(t)$

Reflex:

$\forall t \text{ Máme_zlato}(t) \Rightarrow \text{Action}(Zvednutí, t)$

Reflex s vnitřním stavem: neměli jsme už zlato?

$\forall t \text{ Máme_zlato}(t) \wedge \neg \text{Držím}(Zlato, t) \Rightarrow \text{Action}(Zvednutí, t)$

$\text{Držím}(Zlato, t)$ není pozorovatelné \Rightarrow je důležité držet si informace o vnitřních stavech (např. $\text{Mám_šíp}(t)$)

Báze znalostí pro Wumpusovu jeskyni pokrač.

Vyvozování skrytých skutečností:

- vlastnosti pozice:

$$\forall x, t \text{ Na_poli}(\text{Agent}, x, t) \wedge \text{Je_z\u00e1pach}(t) \Rightarrow \text{Zap\u00e1ch\u00e1}(x)$$

$$\forall x, t \text{ Na_poli}(\text{Agent}, x, t) \wedge \text{Je_v\u00e1nek}(t) \Rightarrow \text{S_v\u00e1nkem}(x)$$

- “V poli vedle Jámy je Vánek:”

- diagnostické pravidlo – odvodí příčiny z následku

$$\forall y \text{ S_v\u00e1nkem}(y) \Rightarrow \exists x \text{ J\u00e1ma}(x) \wedge \text{Vedle}(x, y)$$

- příčinné pravidlo – odvodí výsledek z premisy

$$\forall x, y \text{ J\u00e1ma}(x) \wedge \text{Vedle}(x, y) \Rightarrow \text{S_v\u00e1nkem}(y)$$

- ani jedno z nich není úplné

např. příčinné pravidlo neříká, jestli v poli daleko od Jámy nemůže být Vánek

- definice** vztahu Vánku a Jámy:

$$\forall y \text{ S_v\u00e1nkem}(y) \Leftrightarrow [\exists x \text{ J\u00e1ma}(x) \wedge \text{Vedle}(x, y)]$$

Báze znalostí pro Wumpusovu jeskyni – rozhodování

- počáteční podmínka v *KB*:
 $Na_poli(Agent, [1, 1], S_0)$
- dotaz
 $ASK(KB, \exists s Držím(Zlato, s))$
tj., “V jaké situaci budu držet Zlato?”
- situace jsou propojeny pomocí funkce *Result*:
 $Result(a, s) \dots$ situace, která je výsledkem činnosti *a* v *s*
- odpověď (např. v situaci, kdy hned na vedlejším poli je Zlato)
 $\{s / Result(Zvednutí, Result(Krok dopředu, S_0))\}$
tj., jdi dopředu a zvedni Zlato

Báze znalostí pro Wumpusovu jeskyni – rozhodování

- počáteční podmínka v KB :
 $Na_poli(Agent, [1, 1], S_0)$
- dotaz
 $ASK(KB, \exists s Držím(Zlato, s))$
tj., “V jaké situaci budu držet Zlato?”
- situace jsou propojeny pomocí funkce $Result$:
 $Result(a, s)$... situace, která je výsledkem činnosti a v s
- odpověď (např. v situaci, kdy hned na vedlejším poli je Zlato)
 $\{s / Result(Zvednutí, Result(Krok\ dopředu, S_0))\}$
tj., jdi dopředu a zvedni Zlato

SLiDo

PL1 je dostatečně expresivní logika pro bázi znalostí Wumpusovy jeskyně

Obsah

- 1 Přípomínka – průběžná písemka
- 2 Predikátová logika prvního řádu
 - Predikátová logika 1. řádu
 - Syntaxe predikátové logiky
 - Sémantika predikátové logiky
 - Kvantifikace
 - Substituce proměnných
 - Normální formy v predikátové logice
 - Báze znalostí v PL1
- 3 Příklady jiných logik
 - Fuzzy logika
 - Pravděpodobnost
 - Modální logiky
 - Logiky vyšších řádů
 - Temporální logiky a Intenzionální logiky
- 4 Logika a AI
 - Znalostní inženýrství

Příklady jiných logik

výroková a predikátová logika – **přesné vymezení** (výroků, objektů, relací, ...)

Příklady jiných logik

výroková a predikátová logika – **přesné vymezení** (výroků, objektů, relací, ...)

v reálném světě existují **logicky obtížné** otázky:

Je Brno velké město?

Mají v této restauraci lahodné jídlo?

Je tento člověk vysoký?

odpověď často “**záleží** na okolnostech”

Fuzzy logika

tvrzení mají míru pravdivosti $\in [0, 1]$

Brno je velké město – míra pravdivosti 0.7

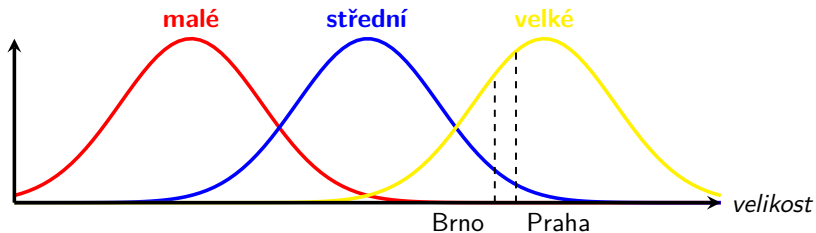
Praha je velké město – míra pravdivosti 0.9

Fuzzy logika

tvrzení mají **míru pravdivosti** $\in [0, 1]$

Brno je velké město – míra pravdivosti 0.7

Praha je velké město – míra pravdivosti 0.9

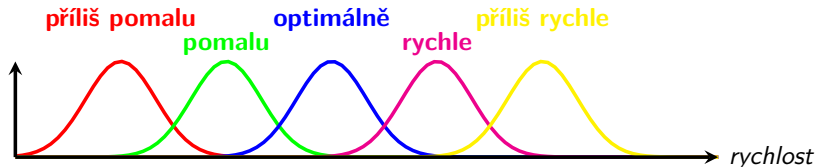
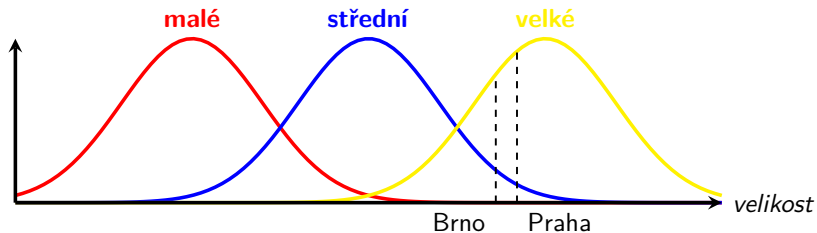


Fuzzy logika

tvrzení mají **míru pravdivosti** $\in [0, 1]$

Brno je velké město – míra pravdivosti 0.7

Praha je velké město – míra pravdivosti 0.9



Pravděpodobnost

hodnoty pravdivosti zůstávají **true** a **false**

liší se **pravděpodobnost** (míra jistoty), že pravdivost je **true** ($\in [0, 1]$)

může vycházet např. z naměřené **frekvence** hodnot v čase

Občané ČR hodnotí Brno jako velké město s pravděpodobností 0.8

rozdíly proti fuzzy logice:

- fuzzy logika pracuje lépe s jazykovými, **subjektivními** hodnotami
pravděpodobnost zpracovává frekvenční, **objektivní** data
- fuzzy logika používá **stupně** příslušnosti a **pravidla**
pravděpodobnost definuje **matematické** vztahy a distribuce
- fuzzy logika umožňuje **částečnou pravdivost**
pravděpodobnost počítá **náhodnost** pomocí statistiky

Modální logiky

nejistotu vyjadřují pomocí **nových operátorů** jako:

- **nutnost** $\Box\phi$ – nutně platí ϕ , ϕ je vždy pravda
- **možnost** $\Diamond\phi$ – možná platí ϕ , ϕ je někdy pravda

operátory \Box a \Diamond jsou vzájemně převoditelné:

$$\Box\phi \Leftrightarrow \neg\Diamond\neg\phi$$

$$\Diamond\phi \Leftrightarrow \neg\Box\neg\phi$$

Lidé jsou smrtelní, tedy neexistuje nesmrtelný člověk.

$$\Box(\forall x \text{člověk}(x) \Rightarrow \text{smrtelný}(x)) \Rightarrow \neg\Diamond(\exists x \text{člověk}(x) \wedge \neg\text{smrtelný}(x))$$

umožňuje zachytit filosofické vztahy **vědomosti** (nějaké tvrzení vím),
povinnosti (závazku) nebo **příčiny**

pravdivost se vyhodnocuje ve vztahu k **možným světům**

Logiky vyšších řádů

Kdo lže, ten krade.

Krást se nemá.

Logiky vyšších řádů

Kdo lže, ten krade.

$\forall x \text{ lže}(x) \Rightarrow \text{krade}(x)$

Krást se nemá.

$\text{špatná_vlastnost}(\text{krade})$

v **logikách vyšších řádů** (*higher-order logic, HOL*) je možné kvantifikovat nejen jednotlivé objekty, ale i **predikáty** a **funkce**

nejvyšší řád může být dán **pevně** (aritmetika – logika 2.řádu) nebo **induktivně** neomezeně

HOL mají **vyšší expresivitu**, ale od 2.řádu pro ně **neexistuje úplná inference**

Temporální logiky a Intenzionální logiky

Petr Pavel je prezident ČR:

- v roce 2024 **true**
- v roce 2020 **false**

temporální logiky pracují s **časem** jako parametrem pro vyhodnocení pravdivosti

Temporální logiky a Intenzionální logiky

Petr Pavel je prezident ČR:

- v roce 2024 **true**
- v roce 2020 **false**

temporální logiky pracují s **časem** jako parametrem pro vyhodnocení pravdivosti

intenzionální logiky (IL) definují extenze (z PL1) jako hodnoty intenzí v daném **světě** a **čase**.

IL umožňují:

- rozlišovat tvrzení **de re** a **de dicto**
*Biden je prezident USA. Trump chce být prezidentem USA,
tedy Trump chce být Bidenem.*

Temporální logiky a Intenzionální logiky

Petr Pavel je prezident ČR:

- v roce 2024 **true**
- v roce 2020 **false**

temporální logiky pracují s **časem** jako parametrem pro vyhodnocení pravdivosti

intenzionální logiky (IL) definují extenze (z PL1) jako hodnoty intenzí v daném **světě** a **čase**.

IL umožňují:

- rozlišovat tvrzení **de re** a **de dicto**
*Biden je prezident USA. Trump chce být prezidentem USA, tedy **neplatí**, že Trump chce být Bidenem.*

Temporální logiky a Intenzionální logiky

Petr Pavel je prezident ČR:

- v roce 2024 **true**
- v roce 2020 **false**

temporální logiky pracují s **časem** jako parametrem pro vyhodnocení pravdivosti

intenzionální logiky (IL) definují extenze (z PL1) jako hodnoty intenzí v daném **světě** a **čase**.

IL umožňují:

- rozlišovat tvrzení **de re** a **de dicto**
*Biden je prezident USA. Trump chce být prezidentem USA, tedy **neplatí**, že Trump chce být Bidenem.*
- rozlišovat postoje
Vím, že Petr věří, že Země je plochá.

Obsah

1 Přípomínka – průběžná písemka

2 Predikátová logika prvního řádu

- Predikátová logika 1. řádu
- Syntaxe predikátové logiky
- Sémantika predikátové logiky
- Kvantifikace
- Substituce proměnných
- Normální formy v predikátové logice
- Báze znalostí v PL1

3 Příklady jiných logik

- Fuzzy logika
- Pravděpodobnost
- Modální logiky
- Logiky vyšších řádů
- Temporální logiky a Intenzionální logiky

4 Logika a AI

- Znalostní inženýrství

Znalostní inženýrství

Knowledge engineering

1. identifikovat **otázky**
2. shromáždit příslušné **znalosti** (*knowledge acquisition*)
3. určit **slovník** predikátů, funkcí a konstant (ontologii) – ovlivňuje efektivitu
4. zakódovat obecné znalosti o **doméně**
5. zakódovat popis instance **problému**
6. položit **dotazy** inferenční proceduře a získat **odpovědi**
7. **ladit** a **evaluovat** bázi znalostí – hledat, jestli jde odvodit nepravdivé nebo nejde odvodit pravdivé závěry

Logika v AI

využití **logických principů** v **AI**:

- **strukturní ekvivalence** – např. elektronických obvodů, kódů
- **verifikace** programů
- **plánování** a **splňování podmínek**
- konzistenční **diagnóza**
- **rozhodování** na základě pravidel a vstupu
- zpracování **neúplných** nebo **nejistých** znalostí
- **strojové učení**:
 - **indukce** hypotéz
 - **neuro-symbolické** učení