

Logické systémy

- Logický systém matematicky přesně definuje

- Co je logický výrok, jaký je jeho význam (syntax a sémantika)
- Co je logická argumentace (důkaz)

- Hilbertovský důkaz ve výrokové logice

- Většinou při důkazu už mám nějaké pravdy (axiómy) a pravidla z kterých vyvozují další věci

* Schémata výrokových axiomů

$$\begin{array}{ll} (A1) & A \rightarrow (B \rightarrow A) \\ (A2) & (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)) \\ (A3) & (\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B) \end{array} \quad \Bigg|$$

- - Kde A, B, C jsou libovolné výrokové formule
 - A1: Pokud A určitě platí a dovím se B, tak A bude pořád platit
 - A2: V podstatě tranzitivita implikace
 - A3: Pokud z výroku ($\neg B$) plyne spor, pak neplatí. Tedy platí opak (B)
- **Odvozovací pravidlo modus ponens (pravidlo odloučení)**
 - Z předpokladů A a ($A \rightarrow B$) odvodíme závěr B
 - MP: Z (prší $\rightarrow$ nemám deštník) a prší odvodím, že nemám deštník
- **Důkaz formule φ**
 - Je sekvence formulí $\varphi_1, \dots, \varphi_n$, kde $\varphi_n = \varphi$ a pro každé $i: 1 \leq i \leq n$, formule φ_i je axiomem nebo vznikla z $\varphi_1, \dots, \varphi_{i-1}$ aplikací odvozovacích pravidel.
 - Pokud existuje důkaz φ , tak je φ **dokazatelná**, psáno $\vdash \varphi$
- **Důkaz z předpokladů**
 - Kde P je množina formulí, je sekvenci formulí $\varphi_1, \dots, \varphi_n$, kde $\varphi_n = \varphi$ a pro každé $i: 1 \leq i \leq n$, formule φ_i je axiomem **nebo prvkem P**
 - nebo vznikla z $\varphi_1, \dots, \varphi_{i-1}$ aplikací odvozovacích pravidel. Píšeme **$P \vdash \varphi$**
- **Příklad důkazu**
 - $\vdash A \rightarrow A$
 - Aplikace A1: $A \rightarrow ((A \rightarrow A) \rightarrow A)$
 - Aplikace A2: $(A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A)$
 - Aplikace MP: $A \rightarrow (A \rightarrow A)$
 - Aplikace A1: $A \rightarrow A$

- **Hilbertovský důkaz ve predikátové logice**

* **Schémata výrokových axiomů**

$$\begin{aligned} (A1) \quad & \varphi \rightarrow (\psi \rightarrow \varphi) \\ (A2) \quad & (\varphi \rightarrow (\psi \rightarrow \eta)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \eta)) \\ (A3) \quad & (\neg\psi \rightarrow \neg\varphi) \rightarrow ((\neg\psi \rightarrow \varphi) \rightarrow \psi) \end{aligned}$$

kde φ, ψ, η jsou formule PL.

* **Schéma axiomů kvantifikátoru:** Není-li x volné ve φ , pak

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

* **Schéma axiomů substitute:**

$$(\forall x \varphi) \rightarrow \varphi[x/t]$$

kde t je term substituovatelný za x do φ (proměnné t nejsou volné ve φ).

* Pokud pracujeme s jazykem s rovností, přidáváme **axiomy rovnosti**:

Pro libovolné funkční a predikátové symboly f/n a p/n a proměnné

$x, x_1, \dots, x_n, y_1, \dots, y_n$

$$x = x$$

$$x_1 = y_1 \rightarrow (x_2 = y_2 \rightarrow (\dots (x_n = y_n \rightarrow f(x_1, \dots, x_n) = f(y_1, \dots, y_n)) \dots))$$

$$x_1 = y_1 \rightarrow (x_2 = y_2 \rightarrow (\dots (x_n = y_n \rightarrow p(x_1, \dots, x_n) \rightarrow p(y_1, \dots, y_n)) \dots))$$

Příklady axiomů PL

* axiom kvantifikátoru: $(\forall x(\varphi \rightarrow \psi)) \rightarrow (\varphi \rightarrow (\forall x\psi))$

$$\begin{aligned} & (\forall \text{Brňák} (\text{je hezky} \rightarrow \text{Brňák je na přehradě})) \rightarrow \\ & \quad (\text{je hezky} \rightarrow (\forall \text{Brňák} (\text{Brňák je na přehradě}))) \end{aligned}$$

* axiom substitute: $(\forall x \varphi) \rightarrow \varphi[x/t]$

$$(\forall x x < \infty) \rightarrow (5 + 5 < \infty)$$

* ax. rovnosti: $x_1 = y_1 \rightarrow (\dots (x_n = y_n \rightarrow p(x_1, \dots, x_n) \rightarrow p(y_1, \dots, y_n)) \dots)$

$$(\text{pivo} = \text{chleba} \rightarrow (\text{snídaně}(\text{chleba}, \text{máslo}) \rightarrow \text{snídaně}(\text{pivo}, \text{máslo})))$$

- **Odvozovací pravidla predikátové logiky**

* *Modus ponens*:

Z předpokladů φ a $(\varphi \rightarrow \psi)$ odvodíme závěr ψ .

* *Pravidlo zobecnění (generalizace)*:

Z předpokladu φ odvodíme závěr $(\forall x\varphi)$.

-
- Např. platí-li $x < \infty$, potom platí $(\forall x x < \infty)$

- **Příklad důkazu (z předpokladů)**

(MP)	$\varphi, \varphi \rightarrow \psi \vdash \psi$
(pravidlo zobecnění)	$\varphi \vdash (\forall x\varphi)$
(ax. kvantifikátoru)	$(\forall x(\varphi \rightarrow \psi)) \rightarrow (\varphi \rightarrow (\forall x\psi))$
(ax. substituce)	$(\forall x\varphi) \rightarrow \varphi[x/t]$
A1, A2, A3, ax. rovnosti	...

- | | |
|---|---------------------------------|
| ▪ $p(x, y) \vdash p(y, x)$ | |
| ▪ $p(x, y)$ | předpoklad |
| ▪ $\forall y p(x, y)$ | pravidlo zobecnění |
| ▪ $\forall x \forall y p(x, y)$ | pravidlo zobecnění |
| ▪ $\forall x \forall y p(x, y) \rightarrow \forall y p(z, y)$ | axióm substituce, x za z |
| ▪ $\forall y p(z, y)$ | MP |
| ▪ $\forall y p(z, y) \rightarrow p(z, x)$ | axióm substituce, y za x |
| ▪ $p(z, x)$ | MP |
| ▪ $\forall z p(z, x)$ | pravidlo zobecnění |
| ▪ $\forall z p(z, x) \rightarrow p(y, x)$ | axióm substituce, z za y |
| ▪ $p(y, x)$ | MP |

- **Formální důkaz versus běžný důkaz**

- Formální logický systém je jakýsi „assembler“ logického odvozování
 - Důkaz je absolutně nezpochybnitelný, mechanicky kontrolovatelný
 - Proto je formální odvozování extrémně detailní, mechanicky jednoduché, a systém je minimalistický
 - Formální systém je stavěn tak, aby se dobře přemýšlelo o něm, ne v něm
- Běžný důkaz je poloformální
 - Čtenář by měl snadno pochopit a zároveň nepochybovat
 - Dělaří se větší kroky, spoléhá se na čtenářovu intuici a znalosti
 - Ideálně je expandovatelný na plně formální důkaz
- Hilbertovský důkaz je snadno, mechanicky ověřitelný, ale není snadno vymyslitelný, a to ani pomoci počítače

- **Důležité pojmy pro písemku**

- **Logický systém** – syntax, sémantika a dokazovací systém
- **Odvozovací, důkazový systém**
- **Axióm** – základní pravda
- **Odvozovací pravidlo**
- **Důkaz**
- **Důkaz z předpokladů**

- Rezoluční důkaz ve výrokové logice

- Jen jedno pravidlo
- Pracuje ale v CNF

Rezoluční pravidlo

$$(\ell \vee A), (\neg \ell \vee B) \vdash A \vee B$$

* kde $(\ell \vee A), (\neg \ell \vee B)$ jsou *konfliktní klauzule*

* $A \vee B$ je *resolvent*(a).

○

SNÍH $\Rightarrow$ TRAMVAJE

(děšť $\vee$ sníh), ($\neg$ sníh $\vee$ $\neg$ tramvaje)

(DĚŠŤ $\vee$ $\neg$ TRAM)

○

- **Rezoluční důkaz φ z množiny předpokladů P** je sekvence formulí patřících buď do P nebo odvozených z předchozích rezolučním pravidlem. Píšeme $P \vdash \varphi$

- Rezoluční důkaz 0 z P je **rezoluční vyvrácení P**

- Rezoluční vyvrácení φ je **důkazem nesplnitelnosti φ**

- **Platnost formule φ pomocí rezoluce** dokážeme tak, že:

- Převědeme $\neg\varphi$ do ekvivalentní formule v CNF s množinou klauzulí C , a
- Rezolučně vyvrátíme C

- **Příklad rezolučního důkazu ve výrokové logice**

Dokazujeme rezolucí formuli φ :

$$\varphi: (P \vee R) \rightarrow S \vee (\neg Q \wedge R) \vee (P \wedge \neg T) \vee ((S \rightarrow Q) \wedge (\neg T \rightarrow Q))$$

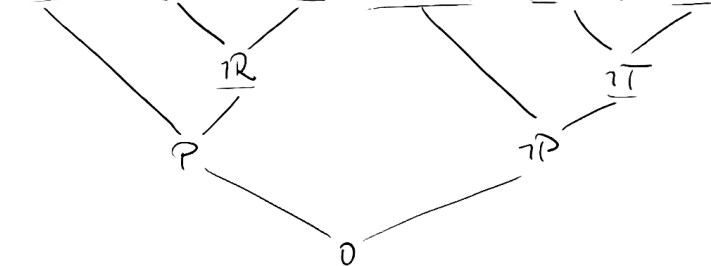
$$\neg[(P \vee R) \rightarrow S \vee (\neg Q \wedge R) \vee (P \wedge \neg T) \vee ((S \rightarrow Q) \wedge (\neg T \rightarrow Q))] \quad (\text{neguj})$$

$$\neg[\neg(P \vee R) \vee S \vee (\neg Q \wedge R) \vee (P \wedge \neg T) \vee ((\neg S \vee Q) \wedge (T \vee Q))] \quad (\text{impl})$$

$$(P \vee R) \wedge \neg S \wedge (Q \vee \neg R) \wedge (\neg P \vee T) \wedge ((S \wedge \neg Q) \vee (\neg T \wedge \neg Q)) \quad (\text{NNF})$$

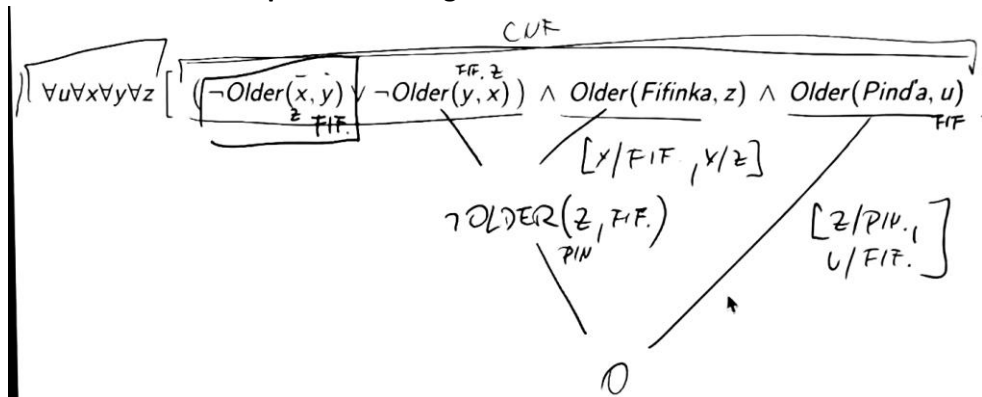
$$(P \vee R) \wedge \neg S \wedge (Q \vee \neg R) \wedge (\neg P \vee T) \wedge \neg Q \wedge (S \vee \neg T) \quad (\text{distrib})$$

$$(P \vee R) \wedge (\overline{Q} \vee \neg R) \wedge \neg Q \wedge (\neg P \vee T) \wedge (S \vee \neg T) \wedge \neg S \quad (\text{CNF})$$



-
- **Původní platí**

○ **Příklad rezolučního důkazu v predikátové logice**



○ **Rezoluční inference v predikátové logice**

$$\forall x \forall y \forall z [(\neg \text{eats}(x, y) \vee \text{fears}(y, x)) \wedge (\text{eats}(x, \text{prey}(x))) \wedge (\neg \text{eats}(x, y) \vee \text{fears}(y, x)) \wedge (\text{eats}(z, \text{prey}(z)))]$$

- Vyvracíme platnost, tj. ukazujeme existenci protipříkladu, **nějakých hodnot proměnných**, pro které formule uvnitř kvant. **Neplatí**
- Můžeme tedy jejich hodnoty omezit tak, aby bylo potom možné použít predikátové rezoluční pravidlo
- Zde konkrétně pomůže omezit se na x stejné jako z a y stejné jako $\text{prey}(z)$

$$\forall z [(\neg \text{eats}(z, \text{prey}(z)) \vee \text{fears}(\text{prey}(z), z)) \wedge (\text{eats}(z, \text{prey}(z)))]$$

$$\forall z [(\text{fears}(\text{prey}(z), z))]$$

- **Substituce**

- Substituce je zobrazení o z proměnných do termů.
- $[X_1/t_1, \dots, X_n/t_n]$ značí substituci σ , kde $\sigma(x_i) = t_i$ pro $1 \leq i \leq n$ a $\sigma(x) = x$ pro ostatní proměnné (množina proměnných bude jasná z kontextu).
- Aplikací substituce σ na výraz (formuli nebo term) E dostaneme výraz $E\sigma$, a to současným nahrazením každého volného výskytu proměnné x termem $\sigma(x)$
Např. pro $E = p(x) \vee p(f(y))$ a $\sigma = [x/y, y/f(a)]$ dostaneme $E\sigma = p(y) \vee p(f(f(a)))$.
Slovem „současně“ v definici aplikace je myšleno, že nenahrazujeme postupně x za y a potom y za $f(a)$, to bychom dostali $p(f(a)) \vee p(f(f(a)))$.

- **Unifikace**

- **Unifikátor** dvou výrazů E a E' je substituce μ taková, že $E\mu = E'\mu$
- Pokud existuje, potom jsou E a E' **unifikovatelné**

Například, necht' a, b, c jsou konstanty, x, y, z proměnné.

Unifikátor $p(x, c)$ a $p(b, c)$ je $[x/b]$.

Unifikátor $p(f(x), y)$ a $p(f(a), z)$ je $[x/a, y/z]$, $[x/a, y/a, z/a]$, $[x/a, y/f(b), z/f(b)] \dots$

$p(x, a)$ a $p(b, c)$ nejsou unifikovatelné, stejně jako $p(f(x), z)$ a $p(a, y)$.

-
- **Maximálně obecný unifikátor (mgu)** výrazů E a E' je unifikátor μ takový, že každý unifikátor E a E' dostaneme jeho složením $\sigma \circ \mu$ s nějakou další substitucí σ

Např. mgu $p(f(x), y)$ a $p(f(a), z)$ je $\mu = [x/a, y/z]$,

ale ne $\mu_1 = [x/a, y/a, z/a]$ ani $\mu_2 = [x/a, y/f(b), z/f(b)]$.

Opravdu, $\mu_1 = [z/a] \circ \mu$ a $\mu_2 = [z/f(b)] \circ \mu$.

▪

▪ Jednoduchý algoritmus pro nalezení mgu

Input: množina M výrazů

Output: mgu μ pro M , pokud existuje, jinak M není unifikovatelná

```

1  $\mu = \emptyset$ 
2 while existují výrazy  $E, E' \in M$  a nejlevější pozice  $i$ , kde se  $E[i]$  liší od  $E'[i]$  do
3   if  $E[i]$  ani  $E'[i]$  není proměnná then
4     return  $M$  není unifikovatelná
5   Necht', w.l.o.g.,  $E[i]$  je proměnná  $x$  a na  $E'[i]$  je term  $t$ .
6   if  $x$  se vyskytuje v  $t$  then
7     return  $M$  není unifikovatelná
8   else
9      $\mu = [x/t] \circ \mu$ 
10    Všechny výrazy  $E'' \in M$  nahraď za  $E''[x/t]$ 
11 return  $\mu$  (nejobecnější unifikátor, mgu, množiny  $M$ )

```

▪ Výpočet mgu

$$M = \{ P(z, w, a), P(u, g(y), y), P(f(x), x, v) \}$$

$$\begin{array}{llll}
 P(z, w, a) & , & P(u, g(y), y) & , & P(f(x), x, v) & [z/u] \\
 P(u, w, a) & , & P(u, g(y), y) & , & P(f(x), x, v) & [w/g(y)] \\
 P(u, g(y), a) & , & P(u, g(y), y) & , & P(f(x), x, v) & [y/a] \\
 P(u, g(a), a) & , & P(u, g(a), a) & , & P(f(x), x, v) & [u/f(x)] \\
 P(f(x), g(a), a) & , & P(f(x), g(a), a) & , & P(f(x), x, v) & [x/g(a)] \\
 P(f(g(a)), g(a), a) & , & P(f(g(a)), g(a), a) & , & P(f(g(a)), g(a), v) & [v/a] \\
 P(f(g(a)), g(a), a) & , & P(f(g(a)), g(a), a) & , & P(f(g(a)), g(a), a) &
 \end{array}$$

Řešení: Nejobecnější unifikátor M (mgu) je substituce

$$\begin{aligned}
 [v/a] \circ [x/g(a)] \circ [u/f(x)] \circ [y/a] \circ [w/g(y)] \circ [z/u] = \\
 [z/f(g(a)), w/g(a), u/f(g(a)), y/a, x/g(a), v/a]
 \end{aligned}$$

•

- Rezoluční odvozovací pravidlo predikátové logiky, základní

Rezoluční pravidlo

$$(\ell \vee \varphi), (\neg \ell' \vee \psi) \vdash \varphi\mu \vee \psi\mu$$

pokud $\text{Free}(\ell \vee \varphi) \cap \text{Free}(\neg \ell' \vee \psi) = \emptyset$ a μ je mgu pro ℓ a ℓ' .

* kde $(\ell \vee \varphi), (\neg \ell' \vee \psi)$ jsou **konfliktní klauzule**,

* $\varphi \vee \psi$ je **resolvent(a)**.

! Pokud není splněna podmínka na prázdný průnik proměnných, je třeba před použitím rezolučního pravidla **přejmenovat proměnné**!

$$(\neg(x = 0)), (z = 0 \vee z > (x - x))$$

○

- Rezoluční odvozovací pravidlo predikátové logiky, obecné

Obecná verze rezolučního pravidla

$$(\ell_1 \vee \dots \vee \ell_n \vee \varphi), (\neg \ell_{n+1} \vee \dots \vee \neg \ell_m \vee \psi) \vdash \varphi\mu \vee \psi\mu$$

pokud $1 \leq n < m$, $\text{Free}(\ell_1 \vee \dots \vee \ell_n \vee \varphi) \cap \text{Free}(\neg \ell_{n+1} \vee \dots \vee \neg \ell_m \vee \psi) = \emptyset$ a μ je mgu pro $\{\ell_1, \dots, \ell_m\}$.

* Sometimes necessary, since the simple rule is not enough.

$$\frac{(p(x) \vee p(y)), (\neg p(u) \vee \neg p(v))}{0} [x/y, u/y, v/y]$$

○

- Rezoluční vyvrácení formule v CNF

Handwritten resolution steps for a formula in CNF:

- Initial clauses: $(\neg p(x,y) \vee p(y,x'))$, $p(x, f(x))$, $(\neg p(x,y) \vee \neg p(y,z) \vee p(x',z))$, $\neg p(x,x')$
- Substitutions: $[x/x', y/f(x)]$, $[x/x, z/f(x)]$, $[x'/x, z/x]$, $[y/x]$
- Intermediate results: $p(f(x), x')$, $\neg p(f(x), z) \vee p(x', z)$, $p(f(x), x)$, $p(x, x)$
- Final result: 0 (contradiction)

○