

Výroková a predikátová logika - V

Petr Gregor

KTIML MFF UK

ZS 2013/2014

Axiomatický přístup

- základní logické spojky: $\neg$, $\rightarrow$ (ostatní z nich odvozené)
- logické axiomy** (schémata logických axiomů):

$$(i) \quad \varphi \rightarrow (\psi \rightarrow \varphi)$$

$$(ii) \quad (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$$

$$(iii) \quad (\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$$

kde φ, ψ, χ jsou libovolné formule (daného jazyka).

- odvozovací pravidlo:**

$$\frac{\varphi, \varphi \rightarrow \psi}{\psi} \quad (\text{modus ponens})$$

Důkaz (Hilbertova stylu) formule φ v teorii T je **konečná** posloupnost

$\varphi_0, \dots, \varphi_n = \varphi$ formulí taková, že pro každé $i \leq n$

- φ_i je logický axiom nebo $\varphi_i \in T$ (axiom teorie), nebo
- φ_i lze odvodit z předchozích formulí pomocí odvozovacího pravidla.

Poznámka Volba axiomů a odvozovacích pravidel se v může v různých dokazovacích systémech Hilbertova stylu lišit.

Příklad a korektnost

Formule φ je **dokazatelná** v T , má-li důkaz z T , značíme $T \vdash_H \varphi$.

Je-li $T = \emptyset$, značíme $\vdash_H \varphi$. Např. pro $T = \{\neg\varphi\}$ je $T \vdash_H \varphi \rightarrow \psi$ pro každé ψ .

- | | | |
|----|---|-----------------------|
| 1) | $\neg\varphi$ | axiom z T |
| 2) | $\neg\varphi \rightarrow (\neg\psi \rightarrow \neg\varphi)$ | logický axiom (i) |
| 3) | $\neg\psi \rightarrow \neg\varphi$ | modus ponens z 1), 2) |
| 4) | $(\neg\psi \rightarrow \neg\varphi) \rightarrow (\varphi \rightarrow \psi)$ | logický axiom (iii) |
| 5) | $\varphi \rightarrow \psi$ | modus ponens z 3), 4) |

Věta Pro každou teorií T a formuli φ , $T \vdash_H \varphi \Rightarrow T \models \varphi$.

Důkaz

- Je-li $\varphi \in T$ nebo logický axiom, je $T \models \varphi$ (logické axiomy jsou tautologie),
- jestliže $T \models \varphi$ a $T \models \varphi \rightarrow \psi$, pak $T \models \psi$, tj. modus ponens je **korektní**,
- tedy každá formule vyskytující se v důkazu z T platí v T . ■

Poznámka Platí i **úplnost**, tj. $T \models \varphi \Rightarrow T \vdash_H \varphi$ pro každou teorií T a formuli φ .

Rezoluční metoda - úvod

Hlavní rysy **rezoluční metody** (*neformálně*)

- je základem mnoha různých systémů, např. interpret Prologu, SAT řešiče, systémy pro automatické dokazování / verifikování, ...
- předpokládá formule v **CNF** (převod obecně “*drahý*”),
- pracuje s **množinovou reprezentací** formulí,
- má jediné odvozovací pravidlo, tzv. **rezoluční pravidlo**,
- nemá žádné explicitní axiomy (či atomická tabla), ale jisté axiomy jsou skryty “*uvnitř*”,
- obdobně jako u tablo metody, jde o **zamítací** (**vyvracecí**) proceduru, tj. snaží se ukázat, že daná fomule (či teorie) je **nesplnitelná**,
- má různé varianty lišící se např. podmínkami pro použití rezolučního pravidla.

Množinová reprezentace (formulí v CNF)

- **Literál** l je výroková proměnná nebo její negace. $\bar{l}$ značí **opačný** literál k l .
- **Klauzule** C je konečná množina literálů (“*tvořících disjunci*”). **Prázdná klauzule** se značí $\square$, není nikdy splněna (neobsahuje splněný literál).
- **Formule** S je množina (i **nekonečná**) klauzulí (“*tvořících konjunkci*”). **Prázdná formule** $\emptyset$ je vždy splněna (neobsahuje nesplněnou klauzuli). Nekonečné formule reprezentují nekonečné teorie (konjunkcí axiomů).
- (**Částečné**) **ohodnocení** $\mathcal{V}$ je libovolná **konzistentní** množina literálů, tj. neobsahující dvojici opačných literálů. Ohodnocení $\mathcal{V}$ je **totální**, obsahuje-li pozitivní či negativní literál od každé výrokové proměnné.
- $\mathcal{V}$ **splňuje** S , značíme $\mathcal{V} \models S$, pokud $C \cap \mathcal{V} \neq \emptyset$ pro každé $C \in S$.

Např. $((\neg p \vee q) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg r \vee \neg s) \wedge (\neg t \vee s) \wedge s)$ reprezentujeme

$$S = \{\{\neg p, q\}, \{\neg p, \neg q, r\}, \{\neg r, \neg s\}, \{\neg t, s\}, \{s\}\} \quad \text{a}$$

$$\mathcal{V} \models S \quad \text{pro} \quad \mathcal{V} = \{s, \neg r, \neg p\}$$

Rezoluční pravidlo

Nechť C_1, C_2 jsou klauzule a $l \in C_1, \bar{l} \in C_2$ pro nějaký literál l . Pak z C_1 a C_2 odvod' **přes literál** l klauzuli C , zvanou **rezolventa**, kde

$$C = (C_1 \setminus \{l\}) \cup (C_2 \setminus \{\bar{l}\}).$$

Ekvivalentně zapsáno, označíme-li $\sqcup$ disjunktní sjednocení,

$$\frac{C'_1 \sqcup \{l\}, C'_2 \sqcup \{\bar{l}\}}{C'_1 \cup C'_2}$$

Např. z $\{p, q, r\}$ a $\{\neg p, \neg q\}$ lze odvodit $\{q, \neg q, r\}$ nebo $\{p, \neg p, r\}$.

Pozorování Rezoluční pravidlo je **korektní**, tj. pro libovolné ohodnocení $\mathcal{V}$,

$$\mathcal{V} \models C_1 \text{ a } \mathcal{V} \models C_2 \Rightarrow \mathcal{V} \models C.$$

Poznámka Rezoluční pravidlo i modus ponens jsou případy **pravidla řezu**

$$\frac{\varphi \vee \psi, \neg \varphi \vee \chi}{\psi \vee \chi}$$

kde φ, ψ, χ jsou libovolné formule.

Rezoluční důkaz

- **rezoluční důkaz** (*odvození*) klauzule C z formule S je **konečná** posloupnost $C_0, \dots, C_n = C$ taková, že pro každé $i \leq n$ je $C_i \in S$ nebo je C_i rezolventou nějakých dvou předchozích klauzulí (i stejných),
- klauzule C je (rezolucí) **dokazatelná** z S , psáno $S \vdash_R C$, pokud má rezoluční důkaz z S ,
- **zamítnutí** (*vyvrácení*) formule S je rezoluční důkaz $\square$ z S ,
- S je (rezolucí) **zamítnutelná** (*vyvratitelná*), pokud $S \vdash_R \square$.

Věta (korektnost) *Je-li S rezolucí zamítnutelná, je S nespínitelná.*

Důkaz Nechť $S \vdash_R \square$. Kdyby $\mathcal{V} \models S$ pro nějaké ohodnocení $\mathcal{V}$, z korektnosti rezolučního pravidla by platilo i $\mathcal{V} \models \square$, což není možné. ■

Rezoluční strom a uzávěr

Rezoluční strom klauzule C z formule S je **konečný** binární strom s vrcholy označenými klauzulemi takový, že

- (i) kořen je označen C ,
- (ii) listy jsou označeny klauzulemi z S ,
- (iii) každý **vnitřní** vrchol je označen rezolventou z klauzulí v jeho synech.

Pozorování C má rezoluční strom z S právě když $S \vdash_R C$.

Rezoluční uzávěr $\mathcal{R}(S)$ formule S je nejmenší induktivní množina definovaná

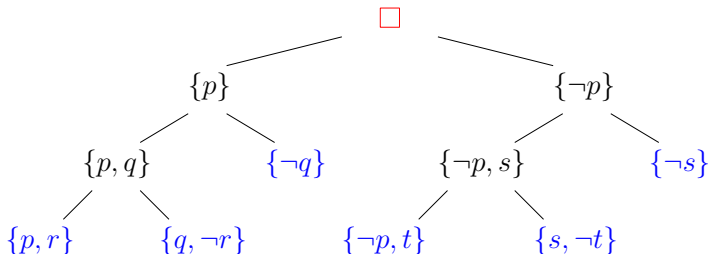
- (i) $C \in \mathcal{R}(S)$ pro každé $C \in S$,
- (ii) jsou-li $C_1, C_2 \in \mathcal{R}(S)$ a C je rezolventa C_1, C_2 , je zároveň $C \in \mathcal{R}(S)$.

Pozorování $C \in \mathcal{R}(S)$ právě když $S \vdash_R C$.

Poznámka Všechny pojmy o rezolučních důkazech lze tedy ekvivalentně zavést pomocí rezolučních stromů či uzávěrů.

Příklad

Formule $((p \vee r) \wedge (q \vee \neg r) \wedge (\neg q) \wedge (\neg p \vee t) \wedge (\neg s) \wedge (s \vee \neg t))$ je nespelnitelná, neboť pro $S = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}\}$ je $S \vdash_R \square$.



Rezoluční uzávěr S je

$$\mathcal{R}(S) = \{\{p, r\}, \{q, \neg r\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{s, \neg t\}, \{p, q\}, \{\neg r\}, \{r, t\}, \{q, t\}, \{\neg t\}, \{\neg p, s\}, \{r, s\}, \{t\}, \{q\}, \{q, s\}, \square, \{\neg p\}, \{p\}, \{r\}, \{s\}\}.$$

Redukce dosazením

Nechť S je formule a l je literál. Označme

$$S^l = \{C \setminus \{\bar{l}\} \mid l \notin C \in S\}.$$

Pozorování

- S^l je ekvivalentní formuli, jež vznikne **dosazením** konstanty $\top$ (true, 1) za literály l a konstanty $\perp$ (false, 0) za literály $\bar{l}$ ve formuli S ,
- S^l neobsahuje v žádné klauzuli literál l ani $\bar{l}$,
- jestliže $\{\bar{l}\} \in S$, pak $\square \in S^l$.

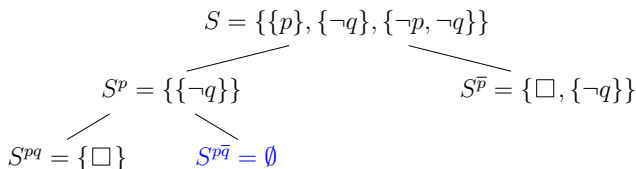
Lemma S je splnitelná, právě když S^l nebo $S^{\bar{l}}$ je splnitelná.

Důkaz ($\Rightarrow$) Nechť $\mathcal{V} \models S$ pro nějaké $\mathcal{V}$ a předpokládejme (búno), že $\bar{l} \notin \mathcal{V}$.

- Pak $\mathcal{V} \models S^l$, neboť pro $l \notin C \in S$ je $\mathcal{V} \setminus \{l, \bar{l}\} \models C$ a tudíž $\mathcal{V} \models C \setminus \{\bar{l}\}$.
- Naopak ($\Leftarrow$) předpokládejme (búno), že $\mathcal{V} \models S^l$ pro nějaké $\mathcal{V}$.
- Jelikož se l ani $\bar{l}$ nevyskytuje v S^l , je i $\mathcal{V}' \models S^l$ pro $\mathcal{V}' = (\mathcal{V} \setminus \{\bar{l}\}) \cup \{l\}$.
- Pak $\mathcal{V}' \models S$, neboť pro $C \in S$ obsahující l máme $l \in \mathcal{V}'$ a pro $C \in S$ neobsahující l je $\mathcal{V}' \models (C \setminus \{\bar{l}\}) \in S^l$. ■

Strom dosazení

Postupnou redukci literálů dosazením lze reprezentovat binárním stromem.



Důsledek S není splnitelná, právě když každá větev obsahuje $\square$.

Poznámka Jelikož S může být nekonečná nad spočetným jazykem, strom může být nekonečný. Je-li ale S nespjitelná, dle **věty o kompaktnosti** existuje konečná část $S' \subseteq S$, která je nespjitelná. Pak po redukci všech literálů vyskytujících se v S' bude $\square$ v každé větvi po konečně mnoha krocích.

Úplnost rezoluce

Věta Je-li *konečná* S nespílitelná, je rezolucí vyvratitelná, tj. $S \vdash_R \square$.

Důkaz Indukcí dle počtu proměnných v S ukážeme, že $S \vdash_R \square$.

- Nemá-li nespílitelná S žádnou proměnnou, je $S = \{\square\}$ a tedy $S \vdash_R \square$,
- Necht' l je literál vyskytující se v S . Dle lemmatu, S^l a $S^{\bar{l}}$ jsou nespílitelné.
- Jelikož S^l a $S^{\bar{l}}$ mají méně proměnných než S , dle indukčního předpokladu existují rezoluční stromy T^l a $T^{\bar{l}}$ pro odvození $\square$ z S^l resp. $S^{\bar{l}}$.
- Je-li každý list T^l z S , je T^l rezolučním stromem $\square$ z S , tj. $S \vdash_R \square$.
- Pokud ne, **doplněním** literálu $\bar{l}$ do každého listu, jenž není z S , (a do všech vrcholů nad ním) získáme rezoluční strom $\{\bar{l}\}$ z S .
- Obdobně získáme rezoluční strom $\{l\}$ z S **doplněním** l ve stromu $T^{\bar{l}}$,
- Rezolucí jejich kořenů $\{\bar{l}\}$ a $\{l\}$ získáme rezoluční strom $\square$ z S . ■

Důsledek Je-li S nespílitelná, je rezolucí vyvratitelná, tj. $S \vdash_R \square$.

Důkaz Plyne z předchozího užitím věty o kompaktnosti.

Lineární rezoluce - úvod

Rezoluční metodu můžeme značně omezit (bez ztráty úplnosti).

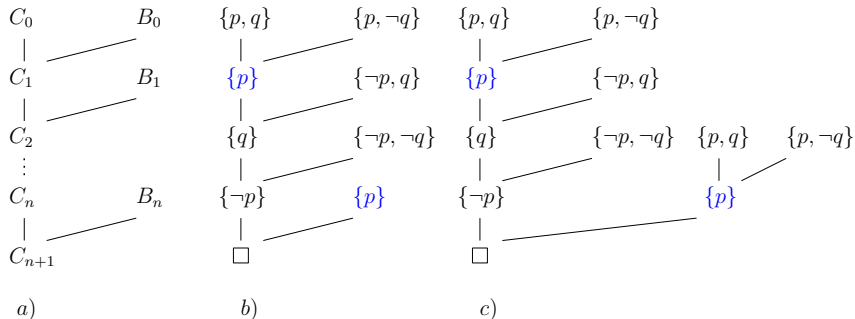
- **Lineární důkaz** (rezolucí) klauzule C z formule S je konečná posloupnost dvojic $(C_0, B_0), \dots, (C_n, B_n)$ taková, že $C_0 \in S$ a pro každé $i \leq n$
 - $B_i \in S$ nebo $B_i = C_j$ pro nějaké $j < i$, a
 - C_{i+1} je rezolventa C_i a B_i , kde $C_{n+1} = C$.
- C_0 zveme **počáteční** klauzule, C_i **centrální** klauzule, B_i **boční** klauzule.
- C je **lineárně dokazatelná** z S , psáno $S \vdash_L C$, má-li lineární důkaz z S .
- **Lineární zamítnutí** (**vyvrácení**) S je lineární důkaz $\square$ z S .
- S je **lineárně zamítnutelná** (**vyvratitelná**), pokud $S \vdash_L \square$.

Pozorování *Je-li S lineárně zamítnutelná, je S nespílitelná.*

Důkaz Každý lineární důkaz lze transformovat na (korektní) rezoluční důkaz.

Poznámka *Platí i úplnost, tj. je-li S nespílitelná, je S lineárně zamítnutelná.*

Příklad lineární rezoluce



a) obecný tvar lineární rezoluce,

b) pro $S = \{\{p, q\}, \{p, \neg q\}, \{\neg p, q\}, \{\neg p, \neg q\}\}$ je $S \vdash_L \square$,

c) transformace lineárního důkazu na rezoluční důkaz.

LI-rezoluce

Pro Hornovy formule můžeme lineární rezoluci dál omezit.

- **Hornova formule** je množina (i nekonečná) Hornových klauzulí.
- **Hornova klauzule** je klauzule obsahující nejvýše jeden pozitivní literál.
- **Fakt** je (Hornova) klauzule $\{p\}$, kde p je pozitivní literál.
- **Pravidlo** je (Hornova) klauzule s právě jedním pozitivním a aspoň jedním negativním literálem. Pravidla a fakta jsou **programové klauzule**.
- **Cíl** je neprázdná (Hornova) klauzule bez pozitivního literálu.

Pozorování Je-li Hornova formule S nesplnitelná a $\square \notin S$, obsahuje fakt i cíl.

Důkaz Neobsahuje-li fakt (cíl), je splnitelná nastavením všech proměnných na 0 (resp. na 1). ■

LI-rezoluce (linear input) z formule S je lineární rezoluce z S , ve které je každá boční klauzule B_i ze (vstupní) formule S .

Je-li klauzule C dokazatelná LI-rezolucí z S , píšeme $S \vdash_{LI} C$.

Úplnost LI-rezoluce pro Hornovy formule

Věta Je-li Hornova T splnitelná a $T \cup \{G\}$ nespílitelná pro cíl G , lze $\square$ odvodit LI-rezolucí z $T \cup \{G\}$ začínající G .

Důkaz Dle věty o kompaktnosti můžeme předpokládat, že T je konečná.

- Postupujeme indukcí dle počtu proměnných v T .
- Dle pozorování, T obsahuje fakt $\{p\}$ pro nějakou proměnnou p .
- Dle lemmatu je $T' = (T \cup \{G\})^p = T^p \cup \{G^p\}$ nespílitelná, přičemž $G^p = G \setminus \{\bar{p}\}$.
- Je-li $G^p = \square$, je $G = \{\bar{p}\}$ a tedy $\square$ je rezolventa G a $\{p\} \in T$.
- Jinak, jelikož T^p je splnitelná (stejným ohodnocením, které splňuje T) a má méně proměnných, dle indukčního předpokladu lze $\square$ odvodit LI-rezolucí z T' začínající G^p .
- **Doplněním** literálu $\bar{p}$ do všech listů, jež nejsou v $T \cup \{G\}$, a všech vrcholů pod ním získáme LI-odvození $\{\bar{p}\}$ z $T \cup \{G\}$ začínající v G .
- Závěrečnou rezolucí pomocí faktu $\{p\} \in T$ získáme $\square$. ■

Příklad LI-rezoluce

$$T = \{\{p, \neg r, \neg s\}, \{r, \neg q\}, \{q, \neg s\}, \{s\}\}, \quad G = \{\neg p, \neg q\}$$

$$T^s = \{\{p, \neg r\}, \{r, \neg q\}, \{q\}\}$$

$$T^{sq} = \{\{p, \neg r\}, \{r\}\}$$

$$T^{sqr} = \{\{p\}\} \quad G^{sq} = \{\neg p\} \quad \{p, \neg r\}$$

$$G^{sqr} = \{\neg p\} \quad \{p\} \quad \{ \neg r \} \quad \{ r \}$$

$$G^{sqrp} = \square$$

$$T^{sqr}, G^{sqr} \vdash_{LI} \square$$

$$G^s = \{\neg p, \neg q\} \quad \{p, \neg r\}$$

$$\{ \neg q, \neg r \} \quad \{ r, \neg q \}$$

$$\{ \neg q \} \quad \{ q \}$$

$$\square$$

$$T^s, G^s \vdash_{LI} \square$$

$$G = \{\neg p, \neg q\} \quad \{p, \neg r, \neg s\}$$

$$\{ \neg q, \neg r, \neg s \} \quad \{ r, \neg q \}$$

$$\{ \neg q, \neg s \} \quad \{ q, \neg s \}$$

$$\{ \neg s \} \quad \{ s \}$$

$$\square$$

$$T, G \vdash_{LI} \square$$

Program v Prologu

(Výrokový) **program** (v Prologu) je Hornova formule obsahující pouze programové klauzule, tj. fakta nebo pravidla.

<i>pravidlo</i>	$p :- q, r.$	$q \wedge r \rightarrow p$	$\{p, \neg q, \neg r\}$	
	$p :- s.$	$s \rightarrow p$	$\{p, \neg s\}$	
	$q :- s.$	$s \rightarrow q$	$\{q, \neg s\}$	
<i>fakt</i>	$r.$	r	$\{r\}$	
	$s.$	s	$\{s\}$	<i>program</i>
<i>dotaz</i>	$?- p, q.$		$\{\neg p, \neg q\}$	<i>cíl</i>

Zajímá nás, zda daný **dotaz** vyplývá z daného programu.

Důsledek Pro každý program P a dotaz $(p_1 \wedge \dots \wedge p_n)$ je ekvivalentní, zda

- (1) $P \models p_1 \wedge \dots \wedge p_n$,
- (2) $P \cup \{\neg p_1, \dots, \neg p_n\}$ je nesplnitelná,
- (3) $\square$ lze odvodit LI-rezolucí z $P \cup \{G\}$ začínající cílem $G = \{\neg p_1, \dots, \neg p_n\}$.

Rezoluce v Prologu

1) S klauzulemi interpret pracuje jako s *uspořádanými seznamy literálů*.

LD-rezoluce (linear definite) je LI-rezoluce, při které v každém kroku rezolventa aktuálního cíle $(\neg p_1, \dots, \neg p_{i-1}, \neg p_i, \neg p_{i+1}, \dots, \neg p_n)$ a boční klauzule $(p_i, \neg q_1, \dots, \neg q_m)$ je $(\neg p_1, \dots, \neg p_{i-1}, \neg q_1, \dots, \neg q_m, \neg p_{i+1}, \dots, \neg p_n)$.

Pozorování Každý LI-důkaz lze transformovat na LD-důkaz stejné klauzule ze stejné formule se stejnou počáteční klauzulí (cílem).

2) Výběr literálu z cílové klauzule, přes který se rezolvuje, je určen daným *selekčním pravidlem* $\mathcal{R}$. Typicky, “vyber první literál z aktuálního cíle”.

SLD-rezoluce (selection) dle $\mathcal{R}$ je LD-rezoluce, při které se v kroku (C_i, B_i) rezolvuje přes literál $\mathcal{R}(C_i)$.

Pozorování Každý LD-důkaz lze transformovat na SLD-důkaz stejné klauzule ze stejné formule se stejnou počáteční klauzulí (cílem).

Důsledek SLD-rezoluce je *úplná* pro dotazy nad programy v Prologu.

Prohledávací SLD-strom

Dosud není určen výběr programové klauzule pro rezoluci s aktuálním cílem.

SLD-strom programu P a cíle G pro selekční pravidlo $\mathcal{R}$ je strom s vrcholy označenými cíly takový, že kořen je označen G a je-li nějaký vrchol označen G' , má tolik synů, kolik je **možností** rezolucí G' s programovými klauzulemi v P dle literálu $\mathcal{R}(G')$. Synové jsou označeni příslušnými rezolventami.

$p :- q, r.$ (1)

$p :- s.$ (2)

$q.$ (3)

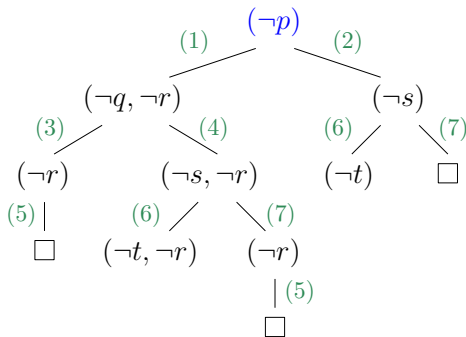
$q :- s.$ (4)

$r.$ (5)

$s :- t.$ (6)

$s.$ (7)

$?- p.$



Závěrečné poznámky

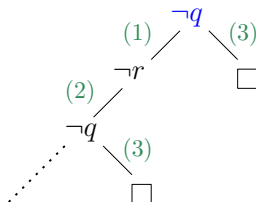
- Interpret Prologu **prochází** SLD-strom, způsob není předepsán.
- Implementace, které používají **DFS**, nezachovávají úplnost.

$q :- r.$ (1)

$r :- q.$ (2)

$q.$ (3)

$?- q.$



- Jistou kontrolu nad prohledáváním poskytuje **!**, tzv. **řez**.
- Při povolení **negace** nastanou potíže se sémantikou programů.
- Síla rezoluční metody bude více patrná v predikátové logice.